Ph.D. Thesis

defended to obtain the
Doctor of Philosophy Degree ès Sciences
from Télécom Paris.

Research developed at the Multimedia
Communications Department of the
Institut Eurécom – Sophia Antipolis.

Facial Motion Analysis on Monocular Images for Telecom Applications: Coupling Expression and Pose Understanding

Ana C. Andrés del Valle

Defended September 19th, 2003 in front of the following jury:

Président:	Prof. Michel Barlaud (I3S)

Rapporteurs:	Prof. Françoise Prêteux (INT)
	Prof. Ferran Marqués (UPC)

	Danielle Pelé (France Télécom R&D)
	Prof. Jörn Ostermann (Universität Hannover)

Thesis supervisor:	Prof. Jean-Luc Dugelay (Institut Eurécom)

Abstract

Facial animation has become an active research topic in telecommunications. This field aims at replacing traditional communication systems by more human oriented solutions based on virtual reality technology.

This thesis relates to a complete analysis/synthesis framework for facial rigid and non-rigid motion analysis from monocular video sequences. The obtained motion data are suitable to animate the realistic head clone of the analyzed speaker by generating face animation parameters. The core of the system is the rigid-motion tracking algorithm, which is able to provide the head pose parameters. The Kalman filter being used predicts the translation and rotation parameters, which are applied on the synthetic clone of the user. This process enables us to benefit from the synthetically generated virtual image by providing visual feedback for the analysis.

This work exposes in detail novel techniques to study non-rigid facial motion coupled with head pose tracking. Specific feature analysis methods have been developed to study each one of the features that we believe to be the most relevant while communicating: eye, eyebrows and mouth. We have designed image-processing algorithms based on the physiognomy of the speaker and individual motion models that exploit the correlation existing among the analyzed features. The analysis techniques have been first developed for faces being analyzed from a frontal point of view and then, using the pose parameters derived from the tracking and the 3D data of the clone, they have been adapted to allow the speaker more freedom of movement in front of the camera. This adaptation is possible by redefining the 2D analysis models over the clone (3D head model), in 3D, and reinterpreting the analyzed data in accordance with the 3D location of the head.

This report contains experimental results, main contributions and relevant bibliographic references of the overall research.

Keywords

Facial animation, 3D, monocular image processing, face-feature analysis, Kalman filtering, expression-pose coupling, telecommunications, face animation parameters, FAP streaming.

Résumé

Les techniques d'animation faciale sont devenues un sujet actif de recherche dans la communauté des télécommunications. Ce domaine a pour but de remplacer les systèmes traditionnels de communications par des solutions plus adaptées aux besoins humains, en utilisant, par exemple, la réalité virtuelle.

Cette thèse doctorale se situe dans le cadre du développement d'un système d'analyse/synthèse qui étudie les expressions et la pose des visages sur des séquences vidéo monoculaires. Le mouvement analysé est utilisé pour animer le clone du visage associé à l'utilisateur, tout en générant des paramètres d´animation faciale. Le noyau central du système mentionné est l'algorithme de suivi du visage qui est capable de générer les paramètres qui déterminent la pose du visage. Le filtre de Kalman utilisé pendant le suivi prédit les angles de rotation et les valeurs de translation qui sont ensuite appliqués sur le clone du locuteur. Ces données nous permettent de profiter de l'image virtuelle de l'animation du clone obtenue pour rétro-alimenter l'analyse.

Ce rapport expose minutieusement une nouvelle approche pour étudier les expressions faciales couplées avec le suivi du visage. Nous avons développé des méthodes d'analyse spécifiques pour chaque trait caractéristique du visage que nous avons considéré comme les éléments les plus importants pendant la communication : les yeux, les sourcils et la bouche. Nous avons conçu des algorithmes basés sur la physionomie du locuteur et qui utilisent des modèles de mouvement individuels pour chacun des traits. Les algorithmes font une double vérification de la cohérence des résultats en utilisant la corrélation existant entre les traits analysés. D'abord, ces algorithmes ont été développés et testés pour fonctionner sur des visages analysés depuis un point de vue frontal. Ensuite, ils ont été adaptés pour travailler avec n'importe quelle pose en utilisant des paramètres de la pose et des données 3D du clone. Cette solution permet une plus grande liberté de mouvement du locuteur face à la camera. L'adaptation est possible en redéfinissant les modèles d'analyse des traits sur le clone (le modèle 3D), et en réinterprétant l'information analysée en relation avec les paramètres 3D qui indiquent la pose du visage.

Ce travail contient les résultats expérimentaux, les contributions principales et les références bibliographiques pertinentes sur l'ensemble des travaux de recherche.

Mots clés

Animation faciale, 3D, traitement des images monoculaires, analyse basée sur traits faciaux, filtrage Kalman, couplage expression-pose, télécommunications, paramètres d´animation faciale, FAP streaming.

Resumen

Las técnicas de animación facial se han convertido en un tema candente de investigación en la comunidad científica de las telecomunicaciones. En este campo se ha propuesto sustituir los sistemas tradicionales de comunicación por soluciones más adaptadas a la necesidades humanas, utilizando la realidad virtual.

Esta tesis doctoral se enmarca en el desarrollo de un sistema de análisis/síntesis que estudia las expresiones y la pose de las caras que aparecen en secuencias de video monoculares. El movimiento analizado es utilizado para animar un clon de la cara del usuario, a media que se generan parámetros de animación facial. El nodo central del sistema mentado es el algoritmo de seguimiento de la cara que es capaz de generar los parámetros que determinan la pose de la cabeza. El filtro de Kalman que es utilizado durante el seguimiento predice los ángulos de rotación y translación que se aplican seguidamente al clon del locutor. Estos datos nos permiten aprovechar la imagen virtual de la animación del clon obtenida gracias a retroalimentación del análisis.

Este informe expone minuciosamente una nueva técnica de estudio de expresiones acopladas al seguimiento de la cara. Hemos desarrollado métodos de análisis específicos para cada rasgo de la cara que hemos considerado más importante para la comunicación humana: los ojos, las cejas y la boca. Hemos concebido algoritmos basados en la fisonomía del locutor y que utilizan modelos de movimiento individuales para cada uno de los rasgos faciales. Los algoritmos verifican la coherencia de los resultados utilizando la correlación existente entre los rasgos analizados. Primero, estos algoritmos han sido desarrollados y testados para que funcionen sobre caras analizadas desde un punto de vista frontal. Después, han sido adaptados para trabajar con cualquier tipo de pose, utilizando los parámetros de la localización y los datos 3D del clon. Esta solución permite más libertad de movimiento al locutor que se encuentra delante de la cámara. La adaptación es posible gracias a que los modelos de análisis son redefinidos sobre el clon (en 3D), y a que se interpreta la información analizada en relación con los parámetros 3D que indican la pose de la cara.

Este trabajo contiene los resultados experimentales, las contribuciones principales y las referencias bibliográficas relevantes a la totalidad de la investigación llevada a cabo.

Palabras clave

Animación facial, 3D, procesado de imagen monocular, análisis basados en rasgis faciales, filtrado de Kalman, acoplamiento expresión-pose, telecomunicaciones, parámetros de animación facial, FAP streaming.

Acknowledgements

I would like to show my appreciation to each one of the jury members for granting part of their time to read and evaluate the research work we have developed for this thesis.

I want to thank my supervisor Professor Jean-Luc Dugelay. He has provided the means for the development of high quality work. I thank Professor Francisco Perales for his cooperation during my brief stay at the UIB. I also thank Institut Eurécom for giving me the opportunity and resources to do my Ph.D and I would like to express my gratitude towards France Telecom R&D for partially supporting my grant.

I say thanks to colleagues and friends from the institute who have shared these four years with me (Philippe de Cuetos, Caroline Mallauran, Carine Simon, Christian Rey, Gwennaël Doerr, Navid Nikaein, and so many others.). I specially mention Adriano Brunetti, Vahid Khamsi, Julien Mouille and Fabrice Souvannoung, the students who have helped me in developing some of the programs for the test platforms.

No quiero dejar de agradecer a mis padres y más queridos amigos su continuo apoyo.
Ellos son la sal de mi vida.

Ευχαριστώ, Σωκράτη. Η εμπειρία σου και η υπομονή σου ήταν οι καλύτερες σύμβουλες
κατά τη διάρκεια αυτών των χρονών του δοκτοράτ.

Muchísimas gracias Isaac: por existir, por estar a mi lado y por comprender.
Tú has compartido mis momentos de tensión,
el final de esta tesis es en parte fruto tuyo. TQM

Table of Contents

Table of Figures

Table of Tables

Notation Conventions

Lower case letters	coordinates and vector components
Capital letters	represent matrices and vectors
Italic	Words or expression with special stress or in a foreign language
	Variables in mathematical expressions
$_n$	neutral 3D-coordinates
$_P$	projected coordinates
t_φ	tan(φ)
s_φ	sin(φ)
c_φ	cos(φ)
2D	Two-dimensional
3D	Three-dimensional
ANN	Artificial Neural Network
AU	Action Unit
BIFS	Binary Format for Scenes
deg	degree
extract	Specific data extracted from references of documents
F	focal distance
f/s	frame per second
FA	Facial Animation
FAP	Facial Action Parameter (MPEG-4 compliant)
fap	General facial animation parameter
FDP	Facial Definition Point
H	Hue
HCI	Human Computer Interface
HMM	Hidden Markov model
HS	Hue and Saturation pixel components
I	Intensity
ICA	Independent Component Analysis
MPEG-4	Motion Picture Expert Group – 4 standard

OGL	OpenGL
PCA	Principal Component Analysis
PDF	Probability Density Function
pel	pixel
pel/s	Pixel per second
ROI	Region of Interest
VRML	Virtual Reality Modeling Language

Introduction

1 Motivation

Face cloning has become a need for many multimedia applications for which human interaction with virtual and augmented environments enhances the interface. Its promising future in different areas such as mobile telephony and the Internet has turned it into an important subject of research. Proof of this interest is the increasing appearance of companies offering their customers the creation of customized synthetic faces and the government support through public grants like the European Project INTERFACE (1999). We can classify synthetic faces in two major groups: avatars and clones. Avatars generally are a rough or symbolic representation of the person. Their appearance is not very accurate. They are speaker-independent because their animation follows general rules independently of the person that they are assumed to represent. Most of the current commercial synthetic faces fall in this category. In some applications, avatars do not completely please people because they may create a feeling of mistrust. In (Ostermann & Millen, 2000), the authors expose how a simple avatar pleases more than an avatar customized by texturing it with a real human face. Indeed, people prefer a good avatar animation to a bad clone synthesis. Clones must be very realistic and their animation must take into account the nature of the person: they need to be speaker-dependent.

Motivated by the multiple advantages and improvements that using realistic virtual characters could supply to telecommunications, we want to investigate the feasibility of using them in traditional videoconference systems, with just one single camera. This dissertation covers the research developed on the creation of new facial motion and expression analysis algorithms in order to replicate human motion on realistic head models that will be used in telecommunication applications.

The complete development of our analysis framework is based on the hypothesis that a realistic 3D-head model of the speaker in front of the camera is available. We believe that realistic motion can only be reproduced on realistic head models and, in such a case, the model is already available to the system. The most accurate information obtained from monocular video sequences taken from standard environments (with

unknown lighting; no markers; …), can only be retrieved if some data about the user's geometry is known 'a priori', for example, by using his realistic clone, as we do.

2 Contributions

We propose new image analysis algorithms for specific features of the face (eye, eyebrows and mouth) that try to profit as much as possible of the physiognomy and the anatomy of the speaker's head. First, these techniques have been defined and tested for a frontal position:

Eye State Tracking: We have developed lighting-independent eye motion estimation algorithms that use natural anatomical intra-feature constraints to obtain gaze and eyelid behavior from the analysis of the energy distribution on the eye area. We have also tested the possibility of using color information during the analysis. We interpret the analysis results in terms of some specific action units that we associate to the temporal states. Following a Temporal State Diagram that uses inter-feature constraints to set the coherence between both eyes, we relate our analysis results to the final parameters that describe the eye movement.

Eyebrow Movement Analysis: To study eyebrow behavior from video sequences, we utilize a new image analysis technique based on an anatomical-mathematical motion model. This technique conceives the eyebrow as a single curved object (arch) that is subject to the deformation due to muscular interactions. The action model defines the simplified 2D (vertical and horizontal) displacements of the arch. Our video analysis algorithm recovers the needed data from the arch representation to deduce the parameters that deformed the proposed model.

The complete ocular expression analysis is obtained after applying some inter-feature constraints among eyes and eyebrows. This allows us to enrich the amount of motion information obtained from each feature, by complementing it with the information coming from another one.

Mouth: It is the most difficult feature to analyze; therefore we believe that a hybrid strategy to derive its motion should be utilized: voice and image conjointly. Our analysis is based on the following facts: mouth motion may exist even if no words are spoken and voiceless mouth actions are important to express emotion in communication. This thesis presents some early results obtained from the analysis technique designed to study the visual aspects of mouth behavior. We deduce which are the mouth characteristics available from the face, that may be the most useful when lighting conditions are not known, and how these characteristics may be analyzed

together to extract the information that will control the muscular-based motion-model template proposed for its analysis.

The main contribution of our work comes from the study of the coupling of these algorithms with the pose information extracted from the rigid head motion tracking system. The presented technique allows the user more freedom of movement because we are able to use these algorithms as independently of the speaker's location as possible.

Facial Expression Analysis Robust to 3D-Head Pose Motion:

Kalman filters are often used in head tracking systems for two different purposes: the first one is to temporally smooth out the estimated head global parameters, the second one is to convert the positions of the 2D facial features observations on the video image into 3D estimates and predictions of the head position and orientation. In our application, the Kalman filter is the central node of our face-tracking system: it recovers the head global position and orientation, it predicts the 2D positions of the features points for the matching algorithm, and –this is the point exploited for telecom applications– it makes the synthesized model have the same scale, position, and orientation as the speaker's face in the real view, despite the acquisition by a non-calibrated camera.

Having already developed and positively tested face feature analysis algorithms for heads studied from a frontal perspective, we need to adapt these algorithms to any pose. In fact, all developed analysis algorithms count on the beforehand definition of the Region of Interest to be analyzed and the automatic location of the interesting features (eyes, eyebrows and mouth). The solution we propose defines the feature regions to be analyzed and the parameters of the motion templates of each feature on 3D, over the head model in its neutral position to automatically obtain them on the image plane thank to the pose data extracted from the tracking. The complete procedure goes as follows:

(i) We define and shape the area to be analyzed on the video frame. To do so, we project the 3D-ROI defined over the head model on the video image by using the predicted pose parameters of the synthesized clone, thus getting the 2D-ROI.

(ii) We apply the feature image analysis algorithm on this area extracting the data required.

(iii) We interpret these data from a three-dimensional perspective by inverting the projection and the transformations due to the pose (data pass from 2D to 3D). At this point, we can compare the results with the feature analysis parameters already predefined on the neutral clone and decide which action has been made.

The technique we use differs from other previous approaches on that we explicitly use the clone 3D data: the location of the model vertices, to define the analysis algorithm on 3D. The main advantages of our solution are the complete control of the location and shape of the region of interest (ROI), and the reutilization of robust image analysis algorithms already tested over faces frontally looking towards the camera.

Other contributions: The thesis contains analyses and discussions about the role of facial animation in telecommunications. We have also given a formal description of facial animation using synthetic models in terms of the generation and the understanding of motion parameters. This theoretical explanation enables the classification of complete facial animation systems by comparing their performance regarding the degree of realism they permit. It also describes a framework to understand the level of interoperability among different facial animation systems.

3 Outline of the Thesis Report

This thesis is organized as follows:

In Chapter I, we review extensively some state-of-the-art analysis techniques for expression analysis on monocular images and their related processing algorithms.

Some of the ideas that help to understand realistic facial animation in the context of communications are explained in Chapter II. We have developed the notions of realism and motion semantics to situate the concept of face cloning in our research.

In Chapter III, we describe the proposed facial animation framework for telecom applications. The requirements of the facial motion analysis techniques needed for the studied framework are also detailed.

Chapter IV includes the development and performance tests of the proposed novel analysis techniques to study facial feature motion on monocular images. It contains the image processing algorithms and related analysis motion templates for eyes, eyebrows and mouth. Experiments show the convenience and robustness of utilizing anatomical knowledge to set intra-feature and inter-feature constraints during the analysis.

In Chapter V, we detail the procedure to couple the use of the developed feature analysis techniques with the knowledge of the head pose. The chapter also includes the theoretical study of the influence of the pose prediction on the analyzed results.

Chapter VI contains the performance analysis of the pose-expression coupling in our face animation framework. To experimentally evaluate the proposed coupling approach, the techniques developed in Chapter IV are adapted following the procedure detailed in Chapter V in order to use the pose tracking technique based on Kalman filtering proposed by Valente and Dugelay (2001).

The thesis concludes with a summary and some comments on future perspectives.

I Facial Image Analysis Techniques & Related Processing Fundamentals

Researchers from the Computer Vision, Computer Graphics and Image Processing communities have been studying the problems associated with the analysis and synthesis of faces in motion for more than 20 years. The analysis and synthesis techniques developed can be useful for the definition of low bit-rate image compression algorithms (model-based coding), new cinema technologies as well as for the deployment of virtual reality applications, videoconferencing, etc. As computers evolve towards becoming more human oriented machines, human-computer interfaces, behavior-learning robots, and disable adapted computer environments will use face expression analysis to be able to react to human action. The *analysis of motion and expression from monocular (single) images* is widely investigated first, because image analysis is the least invasive method to study natural human behavior and, second, because non-stereoscopic static images and videos are the most affordable and extensively used visual media.

I.1 Introduction

Many video encoders do motion analysis over video sequences to search for motion information that will help compression. The concept of *motion vectors*, first conceived at the time of the development of the first video coding techniques, is intimately related to motion analysis. These first analysis techniques help to regenerate video sequences as the exact or approximate reproduction of the original frames, by using motion compensation from neighboring pictures. They are able to compensate but not to *understand* the actions of the objects moving on the video and therefore they cannot restore the object's movements from a different point of view, or immersed in a three-dimensional scenario. Current trends in research focus on the development of new ways of communicating through the use of visual tools that would permit more human interaction while communicating. For instance, this interaction is sought when using 3D in creating virtual teleconference rooms. As said before, traditional motion analysis techniques are not sufficient to provide the information needed for these applications.

Faces play an essential role in human communication. Consequently, they have been the first objects whose motion has been studied in order to recreate animation on synthesized models or to interpret motion for an *a posteriori* use. Figure I-1 illustrates the basic flowchart for systems dedicated to facial expression and motion analysis on monocular images. Video or still images are first analyzed to detect, control and deduce the face location on the image and the environmental conditions under which the analysis will be made (head pose, lighting conditions, face occlusions, etc.). Then, some image motion and expression analysis algorithms extract specific data that is finally interpreted to generate face motion synthesis.

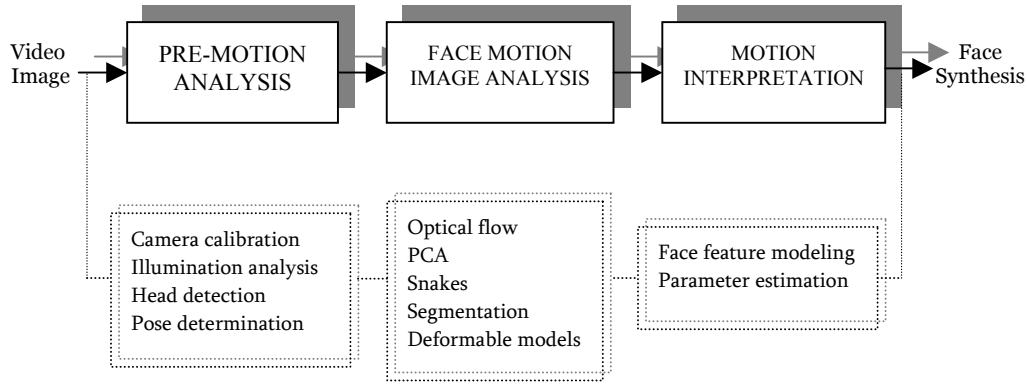


Figure I-1. Image input is analyzed in the search for the face general characteristics: global motion, lighting, etc. At that point some image processing is performed to obtain useful data that can be afterwards interpreted to obtain face animation synthesis

Each of the modules may be more or less complex depending on the purpose of the analysis (i.e., from the understanding of general behavior to exact 3D-motion extraction). If the analysis is intended for later face expression animation, the type of Facial Animation synthesis often determines the methodology used during expression analysis. Some systems may not go through either the first or the last stages or some others may blend these stages in the main *motion & expression image analysis*. Systems lacking the *pre-motion analysis* step are most likely to be limited by environmental constraints like special lighting conditions or pre-determined head pose. Those systems that do not perform *motion interpretation* do not focus on delivering any specific information to perform face animation synthesis afterwards. A system that is supposed to analyze video to generate face animation data in a robust and efficient consists of the three modules. The approaches currently under research and that will be exposed in this section clearly perform the *facial motion and expression image analysis* and to some extend the *motion interpretation* to be able to animate. Nevertheless, many of them fail to have a strong *pre-motion analysis* step to ensure some robustness during the subsequent analysis.

This chapter reviews current techniques for the analysis of single images to derive animation. These methods can be classified based upon different criteria:

1. the nature of the analysis: global versus feature-based, real-time oriented, …;
2. the complexity of the information retrieved: general expression generation versus specific face motion;
3. the tools utilized during the analysis, for instance, the cooperation of a 3D head model;
4. the degree of realism obtained from the Face Animation (FA) synthesis; and
5. the environmental conditions during the analysis: controlled or uniform lighting, head-pose independence.

Table I-1, in page 33, depicts a rough evaluation of the techniques that we review by comparing these criteria, considering the data provided by the referenced articles, books and other bibliographical material and the appreciation of the author. In this section, the systems will be presented in three major categories, grouped following the existing relationship between the image analysis and the expected FA synthesis, namely:

Methods that retrieve emotion information: these are the systems whose *motion & expression analysis* aims at understanding face motion in a general manner. These techniques evaluate the actions in terms of expressions: sadness, happiness, fear, joy, etc. These expressions are sometimes quantified and then interpretable by FA systems but the analysis techniques are not concerned about the FA itself.

Methods that obtain parameters related to the FA synthesis used: this includes the methods that apply image analysis techniques over the images in the search for specific measurements directly related to the animation synthesis.

Methods that use explicit face synthesis during the image analysis: some techniques use the explicit synthesis of the generated animated 3D-head model to compute mesh displacements, generally via a feedback loop.

Regardless of the category they belong to, many of the methods that perform facial analysis on monocular images to generate animation share some image processing and mathematical tools.

I.2 Processing Fundamentals

I.2.1 Pre-processing techniques

The conditions under which the user may be recorded are susceptible to change from one system to another, and from one determined moment to the next one. Some changes may come from the hardware, for instance, the camera, the lighting environment, etc. Furthermore, even though only one camera is used, we cannot presuppose that the speaker's head will remain motionless and looking straight onto that camera at all instants. Therefore, pre-processing techniques must help to homogenize the analysis conditions before studying non-rigid face motion, therefore in this group we also include head detection and pose determination techniques.

Camera calibration

Accurate motion retrieval is highly dependent on the precision of the image data we analyze. Images recorded by a camera undergo different visual deformations due to the nature of the acquisition material. Camera calibration can be seen as the starting point of a precise analysis.

If we want to express motion in real space we must relate the motion measured in terms of pixel coordinates to the real/virtual world coordinates, that is, we need to relate the image reference frame to the world reference frame. Simply knowing the pixel separation in an image does not allow us to determine the distance of those points in the real world. We must derive some equations to link the world reference frame to the image reference frame in order to find the relationship between the coordinates of points in 3D-space and the coordinates of the points in the image. In Appendix I-A we describe the basics of camera calibration. The developed methods can be classified into two groups: photogrammetic calibration and self-calibration. We refer the reader to (Zhang, 2000) and (Luong & Faugeras, 1997) for some examples and more details about these approaches.

Although camera calibration is basically used in Shape From Motion systems, above all, when accurate 3D-data is used to generate 3D-mesh models from video sequences of static objects, it is a desired step for face analysis techniques that aim at providing motion accuracy.

Illumination analysis and compensation

Other unknown parameters during face analysis are the lighting characteristics of the environment in which the user is being filmed. The number, origin, nature and intensity of the light sources of the scene can easily transform the appearance of a face. Face reflectance is not uniform all over the face and thus, very difficult to model. Appendix I-B contains information about the characteristics of the nature of light and one of the most commonly used models for surfaces.

Due to the difficulty of deducing the large number of parameters and variables that the light models compute, some assumptions need to be taken. One common hypothesis is to consider faces as *lambertian* surfaces (only reflecting diffuse light), so as to reduce the complexity of the illumination model. Using this hypothesis, Luong, Fua and Lecrerc (2002) study the light conditions of faces to be able to obtain texture images for realistic head synthesis from video sequences. Other reflectance models are also used (Debevec et al., 2000) although they focus more on reproducing natural lighting on synthetic surfaces than on understanding the consequences of the lighting on the surface itself.

In most cases, the analysis of motion and expressions on faces is more concerned with the effect of illumination on the facial surface studied than with the overall understanding of the lighting characteristics. A fairly extended approach to appreciate the result of lighting on faces is to analyze it by trying to synthetically reproduce it on a realistic 3D-model of the user's head. Whether it is used to compensate the 3D model texture (Eisert & Girod, 2002) or to lighten the 3D model used to help the analysis (Valente & Dugelay, 2001), it proves to be reasonable to control how the lighting modifies the aspect of the face on the image.

Head detection and pose determination

If we intend to perform robust expression and face motion analysis, it is important to control the location of the face on the image plane and it is also crucial to determine the orientation of the face with regard to the camera. The find-a-face problem is generally reduced to the detection of its skin on the image. The most generalized methods for skin detection use a probabilistic approach where the colorimetric characteristics of human skin are taken into account. First, a probabilistic density function - $P(rgb|skin)$ - is usually generated for a given color space (RGB, YUV, HSV, or others.). $P(rgb|skin)$ indicates what is the probability of a color belonging to the skin surface. It is difficult to create this function as well as to decide which will be threshold to use to determine if the current pixel belongs to the skin or not. Some approaches (Jones & Rehg, 1999) study in detail the color models used and also give a probability

function for the pixels that do not belong to the skin - $P(rgb|\neg skin)$. Others, like the one presented by Sahbi, Geman and Boujemaa (2002), perform their detection in different stages, giving more refinement at each step of the detection. More complex algorithms (Garcia & Tziritas, 1999) allow regions with non-homogeneous skin color characteristics to be found.

Determining the exact orientation of the head becomes a more complicated task. In general, we find two different ways to derive the head pose: using static methods and using dynamic approaches. Static methods search for specific features of the face (eyes, lip corners, nostrils, etc.) on a frame-by-frame basis, and determine the user's head orientation by finding the correspondences between the projected coordinates of these features and the *real world* coordinates. They may use template-matching techniques to find the specific features, as Nikolaidis and Pitas (2000) do. This method works fine although it requires very accurate spotting of the relevant features; unfortunately, this action has to be redone at each frame and it is somewhat tedious and imprecise. Another possibility is to use 3D-data, for instance from a generic 3D-head model, to accurately determine the pose of the head on the image. This is the solution given by Shimizu, Zhang, Akamatsu and Deguchi (1998).

To introduce time considerations and to take advantage of previous results, dynamic methods have been developed. These methods perform face tracking by analyzing video sequences as a more or less smooth sequence of frames and they use the pose information retrieved from one frame to analyze and derive the pose information of the next one. One of the most extended techniques involves the use of *Kalman* filters to predict some analytical data as well as the pose parameters themselves. We refer the reader to (Ström, Jebara, Basu & Pentland, 1999; Valente & Dugelay, 2001; Cordea, E. M. Petriu, Georganas, D. C. Petriu & Whalen, 2001) to find related algorithmic details.

Other approaches, like the one presented by Huang and Chen (2000), are able to find and track more than just one face on a video sequence but they do not provide any head pose information. Other techniques (Zhenyun, Wei, Luhong, Guangyou & Hongjian , 2001; Spors & Rabestein, 2001), simply look for the features they are interested in. They find the features' rough location but they do not deduce any pose from this information because their procedure is not accurate enough.

I.2.2 Image processing algorithms

The complexity of expression analysis is usually simplified by trying to understand either the shape of some parts of the face, the location of very specific points or the change in magnitude of some characteristics of the area analyzed, as for example, its color. In order to do this, several image-processing techniques are used and tuned to work on human faces. In this section, we try to summarize the basics of the most commonly used algorithms.

Optical flow

There are two major methods to perform motion estimation: either we match objects with no ambiguity from image to image, or we calculate the image gradients between frames. In the first case, the main goal consists in determining in one of the studied images the group of points that can be related to their homologues in the second image, thus giving out the displacement vectors. The most difficult part of this approach is the selection of the points, or regions to be matched. For practical purposes many application use an *artificial* division of the image into blocks. Block matching algorithms have long been used in video coding. In general, the most important disadvantage of this kind of methods is that it determines motion in a discrete manner, and motion information is only precise for some of the pixels on the image.

In the second case, the field of displacement vectors of the objects that compose a scene cannot be computed directly: we can just find the apparent local motion, also called *optical flow* (OF), between two images. Its computation is also restricted by the *aperture problem* – explained in detail later – consequently, the only component of the motion perpendicular to the contours of an image can be estimated from local differential data.

The most used technique to computes OF, the *gradient-descent* method, generates a dense optical flow map, providing information at the pixel level. It is based on the supposition that the intensity of a pixel $I(x, y, t)$ is constant from image to image, and that its displacement is relatively small. In these circumstances we verify

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0, \qquad \textbf{(I-1)}$$

where $u = \frac{\partial x}{\partial t}$ and $v = \frac{\partial y}{\partial t}$ are the pixel displacements between two images. Each point on the image has one equation with two unknowns, u and v, which implies that motion

cannot be directly computed (it can also be seen as a consequence of the aperture problem [1]). There exist different methods that try to solve (I-1) iteratively.[2]

A complete list of different optical flow analysis methods can be found in (Wiskott, 2001).

Principal component analysis – Eigen-decomposition

Optical flow methods are extensively used in shape recognition but they do not perform well in presence of noise. If we want to identify a more general class of objects, it is convenient to take into account the probabilistic nature of the object appearance, and thus, to work with the class distribution in a parametric and compact way.

The Karhunen-Loève Transform meets the requirements needed to do so. Its base functions are the eigen vectors of the covariance matrix of the class being modeled:

(I-2) $$\Lambda = \Phi^T \Sigma \Phi ,$$

being Σ the covariance matrix, Λ the diagonal matrix of eigen values and Φ the matrix of eigen vectors. The vector base obtained is optimal because it is compact (we can easily isolate vectors of low energy) and parametric (each eigen vector is orthogonal to the others creating a *parametric eigenspace*).

Elements of one class, that is a vector whose dimension is *M*, can be represented by the linear combination of the *M* eigenvectors obtained for this class. The Principal Component Analysis (PCA) technique states that the same object can be reconstructed by only combining the $N<M$ eigen vectors of greatest energy, also called principal components. It also says that we will approximate by minimizing the error difference if the linear coefficients for the combination are obtained from projecting the class vector onto the sub-space of principal components.

---

[1] Equation (I-1) is called the *optical flow constraint equation* since it expresses a constraint on the components u and v of the optical flow.

$$(I_x, I_y) \cdot (u, v) = -I_t.$$

Thus, the component of the image velocity in the direction of the image intensity gradient at the image of a scene point is

$$(u, v) = \frac{-I_t}{\sqrt{I_x^2 + I_y^2}}.$$

We cannot, however, determine the component of the optical flow at right angles to this direction. This ambiguity is known as the *aperture problem*

[2] Some of the most known algorithms are: Lucas and Kanade, Uras, Fleet and Jepson, and Singh algorithms.

This theory is only applicable to objects that can be represented by vectors, such as images. Therefore this theory is easily extensible to image processing and generally used to model the variability of 2D objects on images, like for example, faces.

Very often PCA is utilized to analyze and identify features of the face. However, it introduces some restrictions. One of them is the need for one stage previous to the analysis during which the base of principal component vectors, in this case images, must be generated. It also forces all images being analyzed to be the same size. Using PCA in face analysis has lead to the appearance of concepts like *Eigenfaces* (Turk & Pentland, 1991), utilized for face recognition, or *Eigenfeatures* (Pentland, Mohaddam & Starner, 1994) used to study more concrete areas of faces robustly.

The book *Face image analysis by unsupervised learning* (Bartlett, 2001) gives a complete study of the strengths and weaknesses of the methods based on Independent Component Analysis (ICA[3]) in contrast with PCA. It also includes a full explanation of concepts, like *Eigenactions,* and describes the most recent approaches in face image analysis.

Active contour models - snakes

Active contour models, generally called *snakes*, are the geometric curves that approximate the contours of an image by minimizing an energy function. Snakes have been widely used to track moving contours within video sequences because they have the property of deforming themselves to stick onto a contour that evolves with time.

In general, the energy function can be decomposed in two terms, an internal energy and an external energy:

(I-3) $$E_{total} = E_{\text{int}} + E_{ext} .$$

The role of the external energy is to attract the point of the snake towards the image contours. The internal energy tries to ensure certain regularity on the snake while E_{ext} acts, from a spatial as well as from a temporal perspective.

Once the energy function is defined, we use an iterative process to find its minimum. We can understand the minimum energy point as the equilibrium position of a dynamic system submitted to the forces derived from the energy functions. We find the minimum energy by solving a dynamic equation of second order whose form is similar to:

[3] Using ICA means to apply factoring probability distributions, and blind source separation to image analysis. This technique is related to other fields - entropy and information maximization, maximum likelihood density estimation (MLE), EM (expectation maximization, which is MLE with hidden variables) and projection pursuit. It is basically a way of finding special linear (non-orthogonal) co-ordinate systems in multivariate data, using higher-order statistics in various ways see (ICA, 2003).

(I-4) $$M\ddot{U}+C\dot{U}+KU=F(t)\,.$$

This is why snakes are often represented by a group of weights (the sampling points of the contour) connected by springs (applying the internal forces among the points). U is vector representing the coordinates of the contour points, and M, C and K are the mass, the elasticity and the stiffness of the dynamic system. $F(t)$ is the force function derived form the energy constraints. At equilibrium, the system remains immobilized and follows the shape of the contour.

The most difficult about deploying snakes is their initialization: we need to place the contours close to the border that has to be tracked; otherwise, we may place it close to another contour that also minimizes the energy function (local minimum).

Mathematical morphology - edge detection & segmentation

When analyzing images of faces under unconstrained conditions, classical image filtering techniques may not be robust enough to extract all the information.

Mathematical morphology appeared as an alternative math tool to process image from a visual perspective instead of from a numerical perspective. The techniques for mathematical morphology are based on set-theoretic concepts and non-linear superposition of signals and images. Morphological operations have been applied successfully to a wide range of problems including image processing, analysis tasks, noise suppression, feature extraction and pattern recognition. In (Serra, 1982, 1988), the authors explain in depth how to take advantage of these techniques for the processing of images. The set of tools gives the means to develop algorithms that efficiently detect edges and specific areas of the face.

One of the most used morphological algorithms, the *watershed* transformation, is described in Appendix I-C.

Deformable models (templates)

A deformable model is a group of parametric curves with which we try to approximate the contours of an image and the behavior of its objects. The advantages of a deformable template are its computational simplicity and the few number of parameters needed to describe different shapes. Unfortunately, since a template is generally made specifically for a given shape, we need to redefine the rules of parameter variation so that the model follows and behaves like the right contours. It is also reproachable the fact that it has a difficult adaptation to unexpected shapes, which may become a disadvantage when dealing with noisy images. The diversification of solutions

is well seen in the literature, where we can find as many different models as articles treating the subject. Some of the most common models are:

- Elliptical: circles and ellipsoids can model the eyes (Holbert & Dugelay, 1995).
- Quadratic: parabolic curves are often used to model the lips (Leroy & Herlin, 1995).
- Splines: splines are an option to develop more complex models. They have already been used to characterize mouth expressions (Moses, Reunard & Blake, 1995).

I.2.3 Post-processing techniques and their related mathematical tools

To recreate motion on synthesized 3D-models, it is necessary to relate the analyzed information to the facial Action Units (AUs)[4] of Facial Animation Parameters (FAPs)[5]. If motion is not derived heuristically from the image processing results themselves –perhaps sometimes helped by the iterative feedback synthesis of the motion actions on the model, as seen explained by Eisert and Girod (1998)– we must find some way to tie analysis to synthesis. There are two major approaches to do so:

- by modeling the motion with a direct relationship between the analyzed results and the physical deformation the parameters exert on the model/technique utilized for the analysis, or
- by relating the analysis results to the motion parameters 'blindly', not knowing how the parameters influence the analysis but building the relationship on previously seen results, most of the times through the use of mathematical estimators. This approach needs a training preprocessing to tune the estimators.

Motion modeling of facial features

To extract motion information from specific features of the face (eyes, eyebrows, lips, etc.), we must know the animation semantics of the FA system that will synthesize the motion. Deformable models, snakes, etc. deliver information about the feature in the form of magnitudes of the parameters that control the analysis. It is also necessary to relate these parameters to the actions that we must apply to the 3D-model to recreate motion and expressions. If there exist many different image-processing techniques to analyze face features, there are at least, as many corresponding feature motion models. These motion models translate the results into face animation parameters.

Malciu and Prêteux (2001) track face features using deformable prototypes compatible with the Facial Definition Parameters (FDPs) defined in the MPEG-4 standard. This allows them to deduce the FAPs related to eyes and mouth; they code them into an MPEG-4 stream; and they finally animate a face clone with them. Chou, Chang and Chen (2001) present an analysis technique that searches for the points belonging to the projection of a simple 3D-model of the lips, also containing the FDPs.

[4] AUs are the minimal measurements of actions conceived within the Facial Action Coding System – concept explained in next subsection – to describe facial motion.
[5] FAPs are the minimal actions conceived within the MPEG-4 standard – explained in detail in Chapter VI – to describe facial motion.

From their projected location they derive the FAPs that operate on the FDPs to generate the studied motion. Since one FAP may act on more than one point belonging to the lip model, they use a least-square solution to find the magnitudes of the FAPs involved.

Goto, Kshirsagar and Magnenat-Thalmann (1999) use a simpler approach where image processing is reduced to the search of edges, and where the mapping of the obtained data is done in terms of motion interpretation: open mouth, close mouth, half opened, etc. The magnitude of the motion is related to the location of the edges. They extend this technique to eyes, developing their own eye motion model. Similarly, eyebrows are tracked on the image, and are associated to model actions.

Estimators

Once facial expressions are visually modeled by some image processing technique, we obtain a set of parameters. The mapping of these parameters onto the corresponding face animating parameters can be seen as finding the estimator that relates face motion parameters to analysis parameters. Establishing the mapping relationship requires a training process. Among others, there exist the following estimators: linear, neural networks (NNs), Radial Basis Functions networks, etc. We describe the first two in Appendix I-D. Valente, Andrés del Valle and Dugelay (2001) compare the use of a linear estimator against an RBF network estimator.

ANNs perfectly complement image-processing techniques that need to 'understand' images and in analysis scenarios where some previous training is permitted. This is why they have been used in face recognition for many years and in recent times its use has been extended to the analysis of face motion and expression. In (Tian, Kanade & Cohn, 2001), we find one fine example of the help neural networks can provide. In their article, the authors explain how they have developed the Automatic Face Analysis to analyze facial expressions. Their system takes as input the detailed parametric description of the face features that they analyze. They use neural networks to convert these data into AUs following the motion semantics of the FACS[6]. A similar approach, aiming at analyzing spontaneous facial behavior, is described by Bartlett et al. (2001). Their system also uses neural networks to describe face expressions in terms of

[6] The Facial Action Coding System(© Ekman and Friesen, 1978) FACS objectively describes and measures facial expressions and movements. Based on an anatomical analysis of facial action, it offers a comprehensive method for describing all facial movements, those related to emotion and those that are not in terms of Action Units. EMFACS focuses only on movements known to be related to emotion. A new version of the Facial Action Coding System by Paul Ekman, Wallace V. Friesen, and Joseph C. Hager is complete. If you are training new FACS coders or actively using FACS in research, the new version of FACS is an essential acquisition. The changes to FACS are significant for the future use of FACS and enable much more efficient training of coders. See the Web site below for details of these changes

AUs. These two approaches differ in the image processing techniques and parameters they use to describe the image characteristics introduced as input to the neural network.

Fuzzy systems♦

Fuzzy systems are an alternative to traditional notions of set membership and logic. The notion central to fuzzy systems is that truth values (in fuzzy logic) or membership values (in fuzzy sets) are indicated by a value on the range [0.0, 1.0], with 0.0 representing the absolute Falseness and 1.0 representing the absolute Truth. This is a new approach to the binary set 0 (False) – 1 (Truth) used by classical logic. Fuzzy systems try to gather mathematical tools to represent natural language, where the concepts of Truth or False are too extreme, and intermediate or more vague interpretations are needed. Appendix I-E shows the mathematical basis of fuzzy logic.

The first applications that have benefited from the use of fuzzy systems theory have been information retrieval systems, navigation systems for automatic cars, feature-definition controllers for robot vision, … In many of them, it appears as a complement to the image processing involved and they help in the decision-making process needed to evaluate results from analyzed images. Huntsberger, Rose and Ramaka (1998) have developed a face processing system called Fuzzy-Face that combines wavelet pre-processing of input with a fuzzy self-organizing feature map algorithm. The wavelet-derived face space is partitioned into fuzzy sets, which are characterized by face exemplars and memberships values to those exemplars. The most interesting properties for face motion analysis that this system presents are that it improves the training stage because it uses relatively few training epochs and that it generalizes to face images that are acquired under different lighting conditions. Fellenz, et al. (2000) propose a framework for the processing of face image sequences and speech, using different dynamic techniques to extract appropriate features for emotion recognition. The features will be used by a hybrid classification procedure, employing neural network techniques and fuzzy logic, to accumulate the evidence for the presence of an emotional facial expression and the speaker's voice.

Hidden Markov models

Hidden Markov models (HMM) are a powerful modern statistical technique that has been applied to many subject areas. A Markov process not only involves probability but also depends on the *memory* of the system being modeled.

♦ Information partially taken from "Fuzzy Systems – A Tutorial" by Brulé (1985).

An HMM consists of several states. In the formulation of HMMs, each state is referred to individually, and thus practical and feasible examples of these models have a small number of states. In an HMM, a system has a number of states S_1 …S_n. The probability that the system goes from state *i* to state *j* is called $P(i, j)$. The states of the system are not known, but the system does have one observable parameter on output, which has *m* possible values from 1 to *m*. For the system in state *i*, the probability that output value *v* will be produced is called $O(i, v)$. We must point out that the transition probabilities are required to depend on the state, not the output. Appendix I-F presents techniques for modeling HMMs.

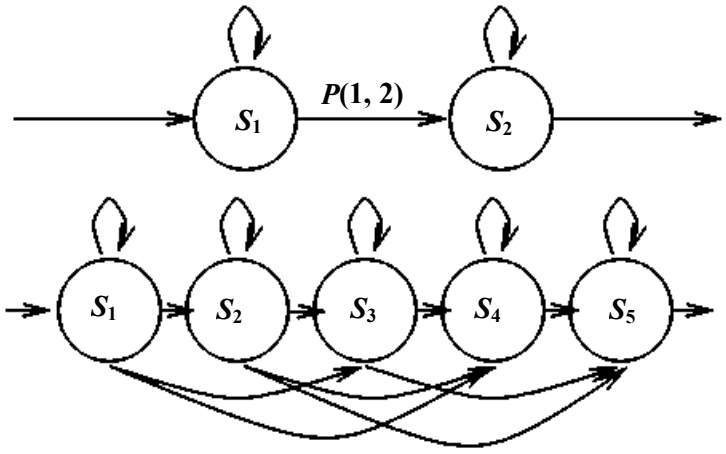


Figure I-2. Top: A typical illustration of a two state HMM. Circles represent states with associated observation probabilities, and arrows represent non-zero transition arcs, with associated probability. Bottom: This is an illustration of a five state HMM. The arcs under the state circles model the possibility that some states may be skipped.

We refer the reader to the tutorial on HMMs by Rabiner (1989), where theoretical bases are further discussed and examples of the most common applications can be found.

A model for motion

We can model behavior patterns as statistical densities over configuration space by collecting data from real human motion. Different configurations have different observation probabilities.

One very simple behavior model is the mixture model, in which distribution is modeled as a collection of Gaussians. In this case the composite density is described by:

(I-5)
$$\sum_{k=1}^{N} P_k \cdot \Pr(O|\lambda = k)$$

where P_k is the observed prior probability of sub-model k.

The mixture model represents a clustering of data into regions within the observation space. Since human motion evolves over time in a complex way, it is advantageous to explicitly model temporal dependence and internal states. An HMM is one way to do this, and has been shown to perform quite well at recognizing human motion. In (Metaxas, 1999), the author presents a framework to estimate human motion (including facial movements) where the traditional use of HMMs is modified to ensure reliable recognition of gesture. More specifically, Pardàs and Bonafonte (2002) use an HMM to deduce the expression of faces on video sequences. They introduce the concept of high-level/low-level analysis. In their approach, the high-level analysis structure takes as input the FAP produced by the low-level analysis tool and, by means of an HMM classifier, detects the facial expression on the frame.

I.3 Face Motion and Expression Analysis Techniques: State of the Art

Systems analyzing faces from monocular images are designed to give motion information with the most suitable level of detail depending on their final application. In this sense, some of the most significant differences among the techniques found in the literature come from the animation semantics they utilize to describe face actions. Some systems aim at providing very high level face motion and expression data in the form of emotion semantics, for instance, detecting joy, fear or happiness on faces. Some others provide generic motion data determining what the action of the facial features is, for example, detecting open/close eyes. And others can even estimate more or less accurately the 3D-motion of the overall face giving out very low-level face animation parameters.

In an analysis-synthesis scheme for generating face animation, both analysis and synthesis parts must share the same level of semantics. The higher the level of detail of the motion information given by the analysis the fewer standard motion interpretations the FA system will have to make. To replicate the exact motion of the person being analyzed it is necessary to generate very detailed action information. Otherwise, if we only generate rough data about the face actions, we will only be able to get customized face motion if the person's expression behavior has previously been studied and the FA already has the specific details of the individual.

It is quite difficult to classify face motion and expression analysis methods due to the common processing characteristics that many of them share. Despite this fact, we have tried to group them based on the precision of the motion information generated and the importance of the role that the synthesis plays during the analysis.

I.3.1 Methods that retrieve emotion information

Humans detect and interpret faces and facial expressions in a scene with little or no effort. The systems we present in this section accomplish this task automatically. The main concern of these techniques is to classify the observed facial expressions in terms of generic facial actions or in terms of emotion categories, and not to attempt to understand the face animation that could be involved to synthetically reproduce them.

Y. Yacoob has explored the use of local parameterized models of image motion for recognizing the non-rigid and articulated motion of human faces. These models provide a description of the motion in terms of a small number of parameters that are intuitively related to the motion of some facial features under the influence of

expressions. The expression description is obtained after analyzing the spatial distribution of the motion direction field obtained from the optical flow analysis, which are computed at points of high gradient values of the facial image. This technique gives fairly good results although the use of optical flow forces very stable lighting conditions and very smooth movement of head motion during the analysis. Computationally, it is also quite demanding. From the early research (Yacoob & Davis, 1994) to the last published results about the performance of the system (Black & Yacoob, 1997), improvements in the tuning of the processing have been added to make it more robust to head rotations.

C.-L. Huang and Y.-M. Huang (1997) introduce a system developed in two parts: facial feature extraction (for the training-learning of expressions) and facial expression recognition. The system applies a point distribution model and a gray-level model to find the facial features. Then, the position variations are described by 10 action parameters (APs). During the training phase, given 90 different expressions, the system classifies the principal components of the APs into 6 different clusters. In the recognition phase, given a facial image sequence, it identifies the facial expressions by extracting the 10 APs, analyzes the principal components, and finally calculates the AP profile correlation for a higher recognition rate. To perform the image analysis, deformable models of the face features are fitted onto the images. The system is only trained for faces on a frontal view, apparently it seems more robust to illumination conditions than the previous approach but no details about the image processing techniques are given, which makes this point difficult to evaluate.

Pantic and Rothkrantz (2000) describe another approach, which is the core of the Integrated System for Facial Expression Recognition (ISFER). The system finds the contour of the features with several methods suited to each feature: snakes, binarization with thresholds, deformable models, etc., making it more efficient under uncontrolled conditions: irregular lighting, glasses, facial hair, etc. It is worth mentioning the NN architecture of the fuzzy classifier, which is designed to analyze the complex mouth movements. In this article, the authors do not present a robust solution to the non-frontal view positions.

To some extend, all systems discussed have based their description of face actions on the Facial Action Coding System proposed by Ekman and Friesen (1978). The importance granted to FACS is such that two research teams, one at University of California San Diego (UCSD) and the Salk Institute, and another at University of Pittsburgh and Carnegie Mellon University (CMU), were challenged to develop prototype systems for automatic recognition of spontaneous facial expressions.

The system developed by the UCSD team, described in (Bartlett et al., 2001), analyzes face features after having determined the pose of the individual opposite the camera; although tests of their expression analysis system are only performed on frontal view faces. Features are studied using Gabor filters and subsequently classified using a previously trained HMM. HMMs are applied in two ways:

- taking Gabor representations as inputs, and
- taking support vector machine (SVM) outputs as inputs.

SVMs are used as classifiers. They are a way to achieve good generalization rates when compared to other classifiers, because they focus on maximally informative exemplars, i.e., the support vectors. To match face features, they first convolve them with as set of kernels (out of Gabor analysis) to make a jet. Then, that jet is compared with a collection of jets taken from training images, and the similarity value for the closest one is taken. In their study Bartlett et al. claim an AU detection accuracy from 80% for eyebrow motion to around 98% for eye blinks. They do not give any results on mouth analysis.

CMU has opted for another approach, where face features are modeled in multistate facial components of analysis. They use neural networks to derive the AUs associated with the motion observed. The have developed the facial models for lips, eyes, brows, cheeks and furrows. In their article, Tian, Kanade and Cohn (2001) describe this technique, giving details about the models and the double use of NN, one for the upper part of the face and a different one for the lower part (see Figure I-3). They do not discuss the image processing involved in the derivation of the feature model from the images. Tests are performed over a database of faces recorded under controlled light conditions. Their system allows the analysis of faces that are not completely in a frontal position; however, most tests are only performed on frontal view faces. The average recognition rates achieved are around 95.4% for upper face AUs and 95.6% for lower face AUs.

Piat and Tsapatsoulis (2000) take the challenge of deducing face expression out of images from another perspective, no longer based on FACS. Their technique first finds the action parameters (MPEG-4 FAPs) related to the expression being analyzed and then they formulate this expression with high-level semantics. To do so, they have related the intensity of the most used expressions with their associated FAPs. Other approaches (Chen & Huang, 2000) complement the image analysis with the study of human voice to extract more emotional information. These studies are oriented to develop the means to create a human-computer interface (HCI) in a completely bimodal way.

The reader can find in (Pantic & Rothkrantz, 2000) overviews and comparative studies of many techniques, including the ones we have discussed. These techniques are

analyzed from the HCI perspective, which contrasts with our considerations about face animation.

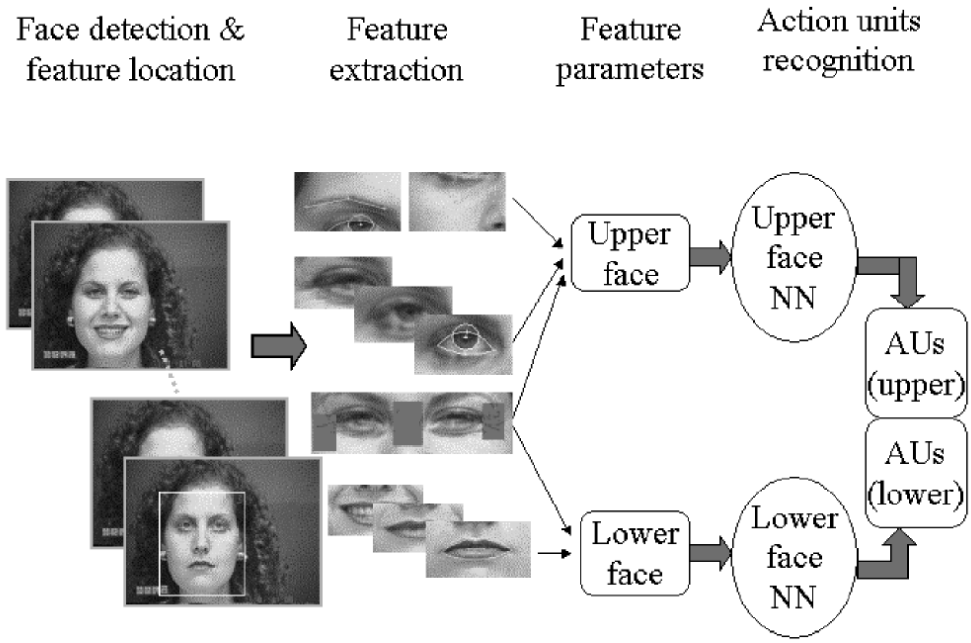


Figure I-3. Face features (eyes, mouth, brows, …) are extracted from the input image; then, after analyzing them, the parameters of their deformable models are introduced into the NNs which finally generate the AUs corresponding to the face expression. Image courtesy of The Robotics Institute at the Carnegie Mellon University.

I.3.2 Methods that obtain parameters related to the Facial Animation synthesis used

Some face animation systems need, as input, action parameters that specify how to open the mouth, the position of the eyelids, the orientation of the eyes, etc. in terms of parameter magnitudes associated to physical displacements. The analysis methods studied in this section try to measure displacements and feature magnitudes over the images to derive the actions to be performed over the head models. These methods do not evaluate the expression on the person's face but extract from the image those measurements that will permit its synthesis on a model.

Terzopoulos and Waters (1993) developed one of the first solutions of this nature. Their method tracks linear facial features to estimate corresponding parameters of a three-dimensional wireframe face model, allowing them to reproduce facial expressions. A significant limitation of this system is that it requires facial features to be highlighted with make-up for successful tracking. Although active contour models are used, the

system is still passive; the tracked contour features passively shape the facial structure without any active control based on observations.

Based on a similar animation system to that of Waters', that is, developed on anatomical based muscle actions that animate a 3D face wireframe, Essa and Pentland define a suitable set of control parameters using vision-based observations. They call their solution FACS+ because it is an extension of the traditional FAC system. They use optical flow analysis along the time of sequences of frontal view faces to get the velocity vectors on 2D and then they are mapped to the parameters. They point out in (Essa, Basun, Darrel & Pentland, 1996) that driving the physical system with the inputs from noisy motion estimates can result in divergence or a chaotic physical response. This is why they use a continuous time Kalman filter (CTKF) to better estimate uncorrupted state vectors. In their work they develop the concept of *motion templates*, which are the 'corrected' or 'noise-free' 2D motion field that is associated with each facial expression. These templates are used to improve the optical flow analysis.

Morishima has been developing a system that succeeds in animating a generic parametric muscle model after having been customized to take the shape and texture of the person that the model represents. By means of optical flow image analysis complemented with speech processing, motion data is generated. These data are translated into motion parameters after passing through a previously trained neural network. In (Morishima, 2001) he explains the basis of this system as well as how to generate very realistic animation from electrical captors on the face. Data obtained from this hardware based study permits a perfect training for coupling the audio processing.

To control the optical flow data generated from the analysis of continuous frames, Tang and Huang (1994) project the head model wireframe vertices onto the images and search for the 2D motion vectors only around these vertices. The model they animate is very simple and the 2D motion vectors are directly translated into 2D vertex motion (no 3D actions are generated).

Almost the same procedure is used by Sarris and Strintzis (2001) in their system for video phoning for the hearing impaired. The rigid head motion (pose) is obtained by fitting the projection of a 3D wireframe onto the image being analyzed. Then, non-rigid face movements (expressions) are estimated thanks to a feature-based approach adapted from the Kanade, Lucas, and Tomasi algorithm. The KLT algorithm is based on minimizing the sum of squared intensity differences between a past and a current feature window, which is performed using a Newton-Raphson minimization method. The features to track are some of the projected points of the wireframe, the MPEG-4 FDPs. To derive MPEG-4 FAPs from this system, they add to the KLT algorithm the

information about the degrees of freedom of motion (one or several directions) that the combination of the possible FAPs allows on the studied feature FDPs.

Ahlberg (2002) also exposes in his work a wireframe fitting technique to obtain the head rigid motion. He uses the new parameterized variant of the face model CANDIDE, named CANDIDE-3, which is MPEG-4 compliant. The image analysis techniques include PCA on *eigentextures* that allow the analysis of more specific features that control the model deformation parameters. They derive 6 different FAPs for their wireframe model.

More detailed feature point tracking is developed in the work of Chou et al. (2001). The authors track the projected points belonging to the mouth, eyes and nostrils provided. These models are also based on the physical vertex distribution of MPEG-4's FDPs. They are able to obtain the combination of FAPs that regenerate the expression and motion of the analyzed face. Their complete system also deals with audio input, analyzing it and complementing the animation data for the lips. The main goal of their approach is to achieve real time analysis to employ these techniques in teleconferencing applications. They do not directly obtain the pose parameters to also synthetically reproduce the pose of the head; but they experiment on how to extend their analysis to head poses other than a frontal view face, by roughly estimating the head pose from the image analysis and rectifying the original input image.

The MIRALab research team, at the University of Geneva, has developed a complete system to animate avatars in a realistic way, so that they can be used for telecommunications. In (Goto et al., 2001), the authors review the entire process to generate customized realistic animation. The goal of their system is to clone face behavior. The first step of the overall process is to physically adapt a generic head mesh model (already susceptible of being animated) to the shape of the person to be represented. In essence, they follow the same procedure as Morishima presents in his work, but T. Goto et al. do it by using just a frontal and side view picture of the individual, whereas Morishima also includes other views to recover texture for the self occlusions. Models are animated using MPEG-4 FAPs, to allow for compatibility with other telecom systems. Animation parameters are extracted from video input of the frontal view face of the speaker and then synthesized, either on the cloned head model or on a different one. Speech processing is also utilized to generate more accurate mouth shapes. An interesting post-processing step is added; analysis results are double-checked before being synthesized and if they are not coherent, they are refused and the system searches in a probability database for the most probable motion solution to the incoherence. In (Goto, Escher, Zanardi & Magnenat-Thalmann, 1999), the authors give a more detailed explanation about the image processing involved. Feature motion models for eyes, eyebrows, and mouth allow them to extract image parameters in the

form of 2D point displacements. These displacements represent the change of the feature from the neutral position to the instant of the analysis and are easily converted into FAPs. Although the system presents possibilities to achieve face cloning, the current level of animation analysis only permits instant motion replication with little precision. In general, we may consider that face cloning is not guaranteed but realistic animation is.

I.3.3 Methods that use explicit face synthesis during the image analysis

Some face motion analysis techniques use the synthesized image of the head model to control or to refine the analysis procedure. In general, the systems that use synthesized feedback in their analysis need a very realistic head model of the speaker, a high control of the synthesis and the knowledge of the conditions of the face being recorded.

Li, Roivainen and Forchheimer (1993) presented one of the first works to use resynthesized feedback. Using a 3D model –Candide–, their approach is characterized by a feedback loop connecting computer vision and computer graphics. They prove that embedding synthesis techniques into the analysis phase greatly improves the performance of motion estimation. A slightly different solution is given by Ezzat and Poggio (1996a, 1996b). In their articles, they describe image-based modeling techniques that make possible the creation of photo-realistic computer models of real human faces. The model they use is built using example views of the face, bypassing the need of any 3D computer graphics. To generate the motion for this model, they use an analysis-by-synthesis algorithm, which is capable of extracting a set of high-level parameters from an image sequence involving facial movement using embedded image-based models. The parameters of the models are perturbed in a local and independent manner for each image until a correspondence-based error metric is minimized. Their system is restricted to understand a limited number of expressions.

More recent research works are able to develop much more realistic results with three-dimensional models. Eisert and Girod (1998), for instance, present a system that estimates 3D motion from image sequences showing head and shoulder scenes typical for video telephone and teleconferencing applications. They use a very realistic 3D head model of the person in the video. The model constrains the motion and deformation in the face to a set of FAPs defined by the MPEG-4 standard. Using the model, they obtain a description of both global (head pose) and local 3D head motion as a function of unknown facial parameters. Combining the 3D information with the optical flow constraint leads to a linear algorithm that estimates the facial animation parameters. Each synthesized image reproducing face motion from frame *t* is utilized to analyze the image

of frame *t*+1. Since natural and synthetic frames are compared at the image level, it is necessary for the lighting conditions of the video scene to be under controlled. This implies, for example, regular well distributed light.

Pighin, Szeliski and Salesin (1999) exploit this approach to the maximum by customizing animation and analysis in a person-by-person basis. They use new techniques to automatically recover the face position and the facial expression from each frame in a video sequence. For the construction of the model, several views of the person are used. For the animation, studying how to linearly combine 3D face models, each corresponding to a particular facial expression of the individual, ensures realism. Their mesh morphing approach is detailed in (Pighin, Hecker, Lischinski, Szeliski & Salesin, 1998). Their face motion and expression analysis system fits the 3D model on each frame using a continuous optimization technique. During the fitting process, the parameters are tuned to achieve the most accurate model shape. Video image and synthesis are compared to find the degree of similarity of the animated model. They have developed an optimization method whose goal is to compute the model parameters yielding a rendering of the model that best resembles the target image. Although being a very slow procedure, the animated results are impressive because they are highly realistic and very close to what we would expect from face cloning (see Figure I-4).

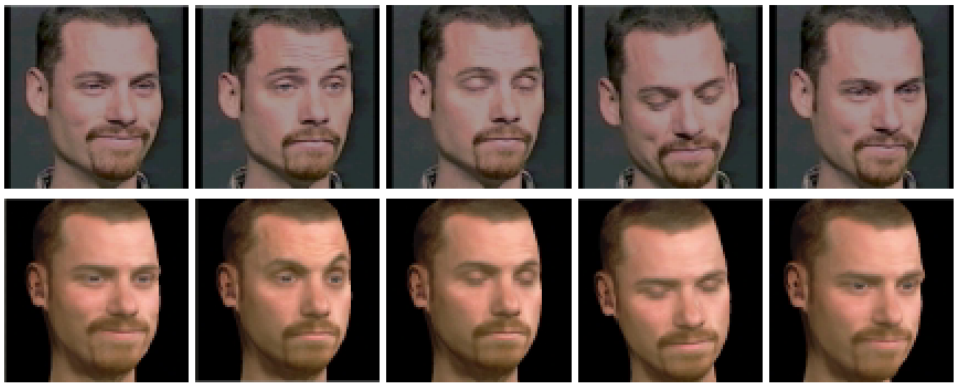


Figure I-4. Tracking example of Pighin's system. The bottom row shows the result of fitting their model to the target images on the top row. Images courtesy of the Computer Science Department at the University of Washington.

Table I-1

COMPARATIVE STUDY OF SOME ANALYSIS TECHNIQUES REVIEWED

			Training?	Controlled lighting?	Does it allow rotations? (pose understanding)	Markers?	Potential real-time?	Dees it use a 3D face model?	Possible synthesis in other head poses?	Realistic reproduction?	Time-line (video) analysis?
Methods that obtain emotion information											
Optical flow / parametric model of image motion	[BY97]	J. Black & Y. Yacoob	N	Y	Y	N	N	N	N.A.	N.A.	Y
Deformable models / PCA	[HH97]	C. H. Huang & Y. M. Huang	Y	Y♣	N	N	N	N	N.A.	N.A.	Y
Feature modeling / neural networks	[TKC01]	Y. Tian et. al.	Y	Y	N	N	N	N	N.A.	N.A.	N
NN / Fuzzy logic / deformable models	[PR00]	M. Pantic & L.J.M. Rothkrantz	Y	N♣	N	N	N	N	N.A.	N.A.	N
HMM / optical flow / Gabor filters / PCA / ICA	[BBL+01]	M. S. Bartlett et. al.	Y	Y	Y	Y♦	N	N	Y	Y	Y
Methods that obtain parameters related to the Face Animation synthesis used (I)											
Snakes	[TW93]	D. Terzopoulos & K. Waters	N	N	N	Y	N	N	Y	Y	Y
Optical flow / Motion templates	[EBD+96]	I Essa et. al.	Y	Y	N	N	N	N	Y	Y	Y
Optical flow / neural networks	[Mor01]	S. Morishima	Y	Y	N	N/Y	N	N	Y	Y	Y
Model fitting / feature point tracking	[SS01]	N. Sarris & M.G. Strintzis	N	N	~	N	Y	Y	Y	N/Y	Y
Model fitting / PCA / active model / *eifentextures*	[Ahl02]	J. Ahlberg	Y	N	~	N	Y	Y	Y	N/Y	Y
Optical flow	[TH94]	Li-an Tang	N	Y	N	N	N	Y	N	N	Y

			Training?	Controlled lighting?	Does it allow rotations? (pose understanding)	Markers?	Potential real-time?	Dees it use a 3D face model?	Possible synthesis in other head poses?	Realistic reproduction?	Time-line (video) analysis?
		& T. S. Huang									
Methods that obtain parameters related to the Face Animation synthesis used (II)											
Feature models	[CCC01]	J. C. Chou, Y.-J. Chang & Y.-C. Chen	N	N	~	N	Y	N	Y	N/Y	Y
Feature motion models	[GKMT01] [GEZ+99]	Goto et. al.	N	N	N	N	Y	N	Y	Y	Y
Methods that use explicit synthesis during the analysis											
Image-based techniques	[EP96] [EP96²]	T. Ezzat & T. Poggio	Y	N	N	N	Y	N	N	Y	Y
Optical flow / spline-based 3D face model	[EG98]	P. Eisert & T. Poggio	N	Y	Y	N	N	Y	Y	Y	Y
3D model fitting / image difference minimization	[PSS99]	F. Pinghin et. al.	Y	Y	Y	N	N	Y	Y	Y	Y

* Author's comment

♦ For the face tracking, which is based in point tracking

~ Slight rotations are permitted although there is no direct use of the pose data during image processing

II Realistic Facial Animation & Face Cloning

Evaluating facial animation systems is an ambiguous task because predefined generalized quality criteria do not exist. Most of the times, the degree of realism and naturalness of synthetic facial reproduction is determined from subjective judgment.

This chapter contains the definition of some theoretical concepts related to facial animation. We have tried to formalize the notion of realism within the context of our research. We aim at providing a conceptual basis where the notions of avatar and clone are clearly and unambiguously stated. This formal framework allows us to describe the interaction existing between facial motion generation and its synthesis from a global perspective.

We conclude the chapter with some considerations about face cloning viewed from an ethical perspective.

II.1 Understanding the Concept of Realism in Facial Animation

Realism is the latest and greatest challenge human animation confronts. Computer graphic techniques have already being used *to bring to life* synthetic characters that behave in a more or less human way. In the past few years, we have been able to see the first results on three-dimensional realistic human animation in the entertainment industry. We find one of the most impressive results in the movie *Final Fantasy* (Lee, 2001). All the characters and scenarios of this movie have been rendered synthetically and, unlike other examples of computer-graphic aided films, its creators have aimed at reproducing humans in a highly realistic manner. In general, the degree of realism sought depends on the application which the synthetic human character will be involved in.

Face actions and expressions are very important in human communication and the lack of them is one of the weaknesses of traditional telecom systems (e-mail and telephone). Video-based communications seem to be the solution to this problem. Nowadays, telecom networks are not ready to carry all the video data to enable the extensive use of high quality teleconference, and users do not like the constrained environment that one-to-one communications with very little motion flexibility impose. Synthesized or virtual characters, above all animated faces, have appeared as a possible way-out to create better communication environments at a lower network cost. Talking-Heads (2002) that represent speakers in conversations are already a reality. Unfortunately, we cannot create easily face models and animate them in such a way that they are able to substitute not only the physical presence but also the trust we have on the real person we are speaking with. In order to achieve so, we need to *build* 3D-head models that can be *realistically* animated; furthermore, we will have to make a *virtual clone* of the speaker, implying the customization the 3D head motion to the actions of the speaker, if we want the person to be completely represented by its synthesized model.

It is difficult to determine at which point we can consider a realistic 3D-head model to be animated realistically. In fact, realism depends on two issues: on the one hand, the physical appearance of the head model and, on the other hand, the motion and actions the model can generate. Ideally, a 3D-head model is realistic when the surface or surfaces that compose the synthesize object exactly represent a human head. To obtain so, not only the geometry of the model must be a detailed human reproduction but also the texture, color and light reflection characteristics of the surface must match those of the parts of a human head. We talk about realistic head animation when we refer to the number R of undetermined but limited motion actions that the human head model can

generate. One motion action is each of the 3D-movements $B\vec{v}$ ♦, where $B\vec{v} = (\Delta x, \Delta y, \Delta z)$, which are exerted on each of the points belonging to the surface of the model to create animation. The magnitude of these displacements, *B*, is limited to the maximum value permitted by real human motion. Due to the wide range of actions and the complexity of human nature it is very difficult to completely render highly realistic virtual heads. It took 5 years to finish *Final Fantasy*, each of the actions of the human virtual characters involved thousands of computations, and nevertheless, the result, although very astonishing, is still far from being humanly believable.

A *head clone* is a specific case of realistic head animation. We consider that a clone is a realistic 3D-head representation of a living human. From the overall actions *R* that a realistic head model can do, only $S \in R$, will be exerted. *S* are those actions that involve the specific and exact head motion and expressions of the person whom the clone represents. Obviously, this definition is the ideal that all FA systems should target when cloning people. It seems *a priori* very difficult and almost impossible to generate the right and exact motion belonging to someone's action, and even as difficult to generate the 3D-model representation of the person. That is why, in a more general way, we call *head clone* the 3D-head model and the group $C \in S$ of actions that does not allow us to differentiate the rendered moving representation of the clone from a recorded video from the person. Figure II-1 represents graphically the relationship amongst the different action groups.

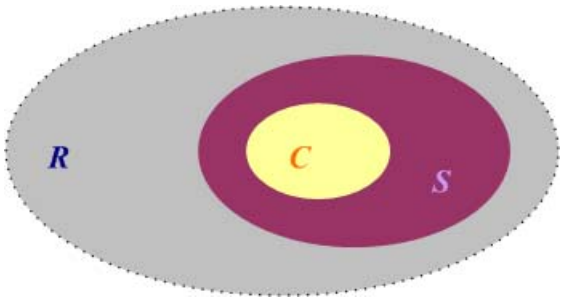


Figure II-1. It is simple to understand how we can generate cloned actions in a realistic way if we ensure that the group of actions $C \subset S \subset R$. *R* represent actions feasible on a realistic head model animation, C those actions that would customize the realistic animation to become a clone, S are the final action that we are able to replicate from the study of the media input, in our case, monocular images.

We must point out that the concept of *difference* between a human being recorded and a rendered clone is subjective. Realistic human head actions, actions belonging to *R*, will always ensure a human-like clone behavior, but they may be just too general to

♦ We define motion actions as physical vector displacements of the surface points, but we do not imply the use of vertex displacements as the only way to generate synthesized animation, just that the final rendered result must imitate the natural head surface motion (deformations seen as displacements of the infinite points that belong to the surface). $B\vec{v}$

represent the person being cloned. Similarly, just using a subgroup S of all the possible actions being susceptible of occurring when that person communicates may not be sufficient to represent the individual. Judging the realism will depend on the degree of acquaintance we have with the person himself. For instance, for those people that intimately know the person being cloned, it will be much easier to find out they are dealing with his clone.

Head avatars are another class of animated head models. An avatar is a rough or symbolic representation of an entity, in this case, a person. When an avatar takes the form of a head/face there is no pre-established rule that forces it to act in a realistic human-like way. The number of actions permitted V is fairly more extensive than those allowed to realistic facial animation, but since they are just a rough 3D representation they do not permit complete realistic synthesized behavior. Only a limited group $V_R = R \cap V$ of actions will fall in the definition of realistic human head animation. Avatars willing to create a human-like feeling will only use actions belonging to V_R.

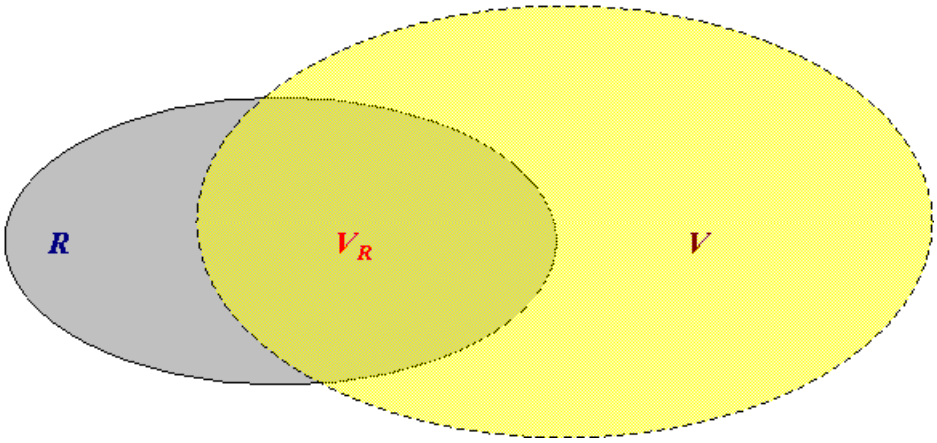


Figure II-2. The freedom of action permitted on avatars V is larger than the one allowed on realistic head models R. Avatars are meant to just be a rough representation and therefore they are limited by the nature of their models. $V_R = R \cap V$ is the group of actions performed on an avatar that will make it behave like a human being

II.2 The *Semantics* of Facial Animation

As stated in the previous section, 3D-head animation synthesis can be seen as the combination of motion actions to be applied on each of the points belonging to the head-model surface. Let us call A^i the group of all possible actions that a head model can undergo in the specific FA system *i*. Each time a new FA system is created we need to set its A^i so to specify how to generate the minimal actions on the head model, $a_n^i = B\vec{v}\big|_{i,n}$. Most of the times, determining each of the single movements, a_n^i, that the model must undergo to generate a specific action is too complicated to do in a one-per-one basis; therefore, this operation is done once. Afterwards, actions are grouped following some sort of *semantic motion* criteria associated to the movement they involve. Each of these subgroups, A_j^i, represents a more or less complex combination of actions related amongst themselves by its motion.

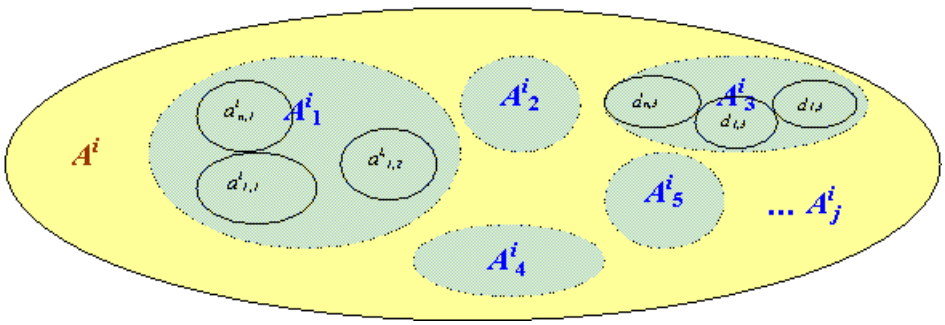


Figure II-3. A specific FA system has its own animation designed once: $A^i = \{a_n^i\}$. Afterwards, minimal actions, a_n^i, are gathered in more or less large groups associated by the semantics of their movement

We define the concept of *semantic level* as the amount of motion information given by each of the subgroups A_j^i, that is, it refers to the degree of detail that a FA general action A_j^i includes. For instance, the same overall action to be performed over a 3D-head model –let us choose as an example *how to close the right eye*– may be expressed in many different ways by using more or less detailed semantic levels. Taking very *high-level* semantics, an FA system could directly understand the action «close the right eye»: A_j^1. Another FA system, using more detailed semantics, may define the action like «move right eye upper lid down and move right eye lower lid up»: A_j^2. And if the FA system only understands very *low-level* actions then we could get «move point $p_1 \in right\_eye$ y

units along the y-axis, move $p_2 \in right\_eye$ z units along the z-axis, etc.»: A_j^3. Let j denote an overall action of the same nature.

Naturally, for a determined FA technique, generation and animation are always in *semantic resonance*, that is, all subgroups A_j^i are completely understood and synthetically rendered by the model. This must occur regardless of the origin chosen for the generation of the movement actions. Most of current FA systems animate motion derived from manually set actions. Currently, new ways of obtaining actions are been searched for and there exist some encouraging results to automatically generate head/face actions from different media: text, audio and video.

Since the *semantic* meaning of motion may be very different from one FA technique to another, most of the times, it is difficult for actions defined and designed for a concrete FA system to be understood by a different one. High-level actions generally are a compound of several low-level actions. Although the final motion animation done on the 3D-head model is always performed by applying minimal (the lowest semantic level) actions, a_n^i, it does not mean that all general actions are universally understood. For a specific FA system to understand motion actions designed for a different FA technique, it must semantically transform through $T\{\}$, the incoming actions and translate them into understandable motion actions before applying them on the head model. The $T\{\}$ transform will only be possible if both FA systems share some minimal action a_n^i for a given general movement *j*.

In general, $T\{\}$ is difficult to find. The minimal actions grouped by a FA system for a specific movement, A_j^1, could be included in several groups in the other FA system: $\{A_j^2, A_i^2, A_k^2, \ldots\}$; or vice versa, several action groups from the FA origin, $\{A_j^1, A_i^1, A_k^1, \ldots\}$, may be contained in just one group at the FA synthesis A_j^1. Figure II-4b and Figure II-4c illustrate these situations.

When generating motion information to clone human behavior, we must ensure that the semantic motion information obtained during the analysis of the person's actions is completely understandable by the FA system that will render the head model.

Figure II-5 presents the three major situations that appear when connecting a system generating motion data with a FA system:

- **Resonance:** In this situation motion data completely suits the FA semantics and it is directly understood.
- **Misunderstanding:** The motion data generated is not directly understandable by the FA system. The semantics of the motion data are incompatible with those of the synthesis.

- **Understanding:** After adapting the generated motion data, initially incomprehensive for the FA system, we are able to animate the model. The most optimal situation happens when both motion generation and synthesis are 100% compatible. In this case, there will be no loss of information. Otherwise, the FA system will not render all the motion data generated.

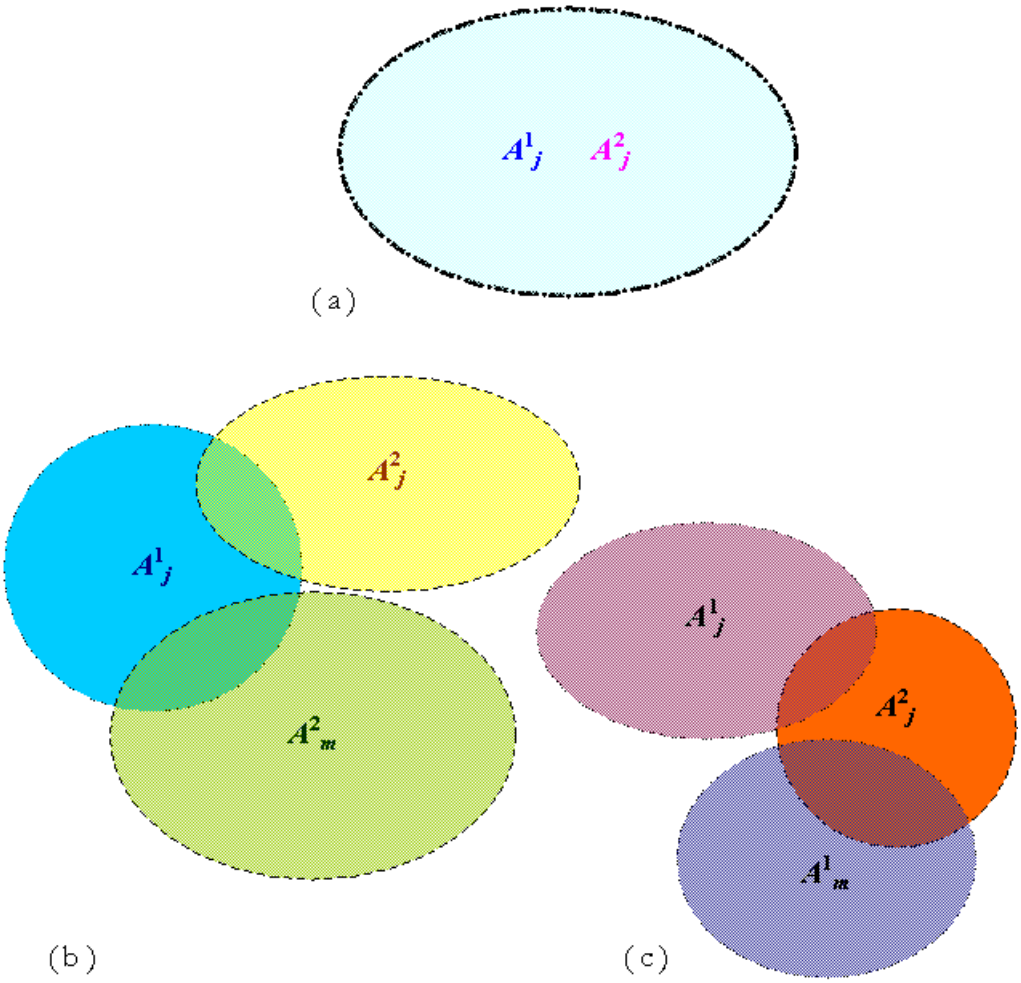
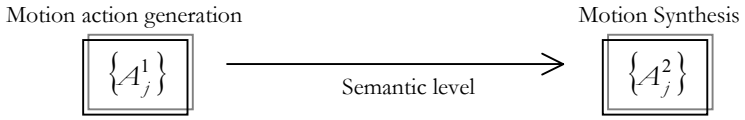


Figure II-4. Actions designed to animate face models in a specific FA system (A_j^i) are grouped following the semantics of the motion. In (a) we illustrate how the semantics of the generated animation parameters , A_j^1, and those of the animation system, A_j^2, are the same. In (b) the general action generated is expressed by means of several actions of the FA system; and in (c) we need to generate diverse general actions to animate just one

Resonance (same semantic level):

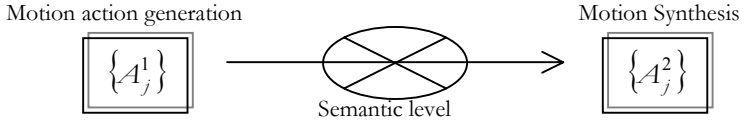


$$A^1_j = A^2_j, \quad A^1_j \cap A^2_j = A^2_j \quad \forall j$$

Example:

Action:
Turn eye pupil +10 degrees over the y-axis.
Synthesis understanding:
Turn eye pupil +10 degrees over the y-axis.

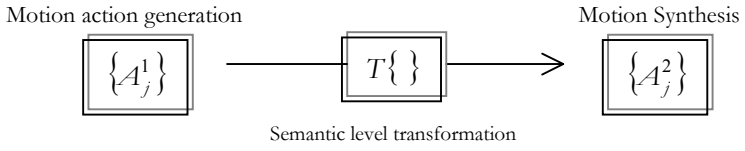
Misunderstanding (different semantic level):



$$A^1_j \cap A^2_j = \varnothing \quad \text{for some } j$$

Action:
Turn eye pupil +10 degrees over the y-axis.
Synthesis understanding:
Nothing.

Understanding (adapting semantic levels):



$$T\{A^1_j\} \cap A^2_j = A^{1,2}_j \quad \text{for some } j$$

$$T\{\ \}\exists \Rightarrow A^1_j = \{a^1_{1,j}, a^1_{2,j}, a^1_{3,j}, \ldots, a^1_{N,j}\}, \quad A^2_j = \{a^2_{1,j}, a^2_{2,j}, a^2_{3,j}, \ldots, a^2_{M,j}\},$$

$$\& \; a^1_{k,j} = a^2_{k,j} \; \textit{for at least one } i \textit{ of one general movement } j$$

In resonance, $T\{\ \} = I$; I being the identity function.

Action:
Turn eye pupil +10 degrees over the y-axis.
Synthesis Transform:
Understand physical eye motion: +10 deg = right.
Synthesis understanding:
Look right.

Figure II-5. When generation and synthesis of animation are in *resonance*, all generated movements are completely understood and reproduced. If the system of animation generation does not follow the semantics of the face motion synthesis, there is *misunderstanding* and we need to adapt motion parameters to have some *understanding*. This is possible if both sides, generation and synthetic animation, share at least some minimal motion actions

In a complete FA system, analysis and synthesis must share the same *semantics* and the same *syntax* in their motion description. While by semantics we have referred to the concept of generating the same set of possible actions or analyzed movements as the synthesis is able to render, in the concept of syntax we imply the way this motion is described. Given certain parameters and magnitudes involved in a specific movement, we should be able to express the same action while they are generated as well as when they are applied. Although accomplishing these requirements apparently seems a trivial task, in current research there is still little implication in how the motions semantics and syntax determine the way action parameters should be generated. Therefore, proposed solutions to achieve the same goal, face motion analysis and synthesis, are sparse and have led to the development of algorithms to be used in very specific environments.

In the history of Computer Graphics, there have been several attempts to define groups of actions and the associated semantics for face animation. Researchers have generally focused on the generation of face expressions. In his Ph.D. dissertation *A parametric model for human faces*, Parke (1974) defined a facial model that is rendered as a mesh of Gouraud-shaded polygons from a set of real-valued parameters. Parke's model allowed both the expression and the *conformation*, or shape, of the face to be defined by parameters. Ekman and Friesen (1978) defined a system (the Facial Action Coding System, or FACS) for encoding facial expressions in terms of *action units* (AUs). Their system was designed to be used by noting down observed facial expressions, and defines six categories for complete expressions (happiness, sadness, anger, fear, disgust and surprise). Waters (1987) based his facial animation system on FACS, mapping the action units to muscles in the face. Waters also rendered the face using finer meshes of polygons, displaced smoothly so as to model skin elasticity. Kalra, Mangili, Magnenat-Thalmann and Thalmann (1992) developed another technique for more realistically rendering facial images; they used rational free form deformations, a method of distorting curved surfaces by manipulating grid points surrounding them. The facial model used is similar to FACS, in that it was made of perceivable actions; it consisted of 21 Minimum Perceptible Actions (MPAs), each represented by a real value.

Later works on face animation, above all those focused on very realistic animation (Eisert & Girod, 1998; Pighin, Szeliski & Salesin, 1999), define more concrete and precise actions for the model animation. Their semantics are just suited for their specific system and cannot be easily generalized, thus making the use of their actions by other systems apparently more difficult.

II.3 Animating Realism

As stated in Section II.1, actions exerted on a head-model must belong to the group *R* of realistic human motion actions to create realistic facial animation, that is, all $a_n^i \in R$ for a given FA system *i*.

Realistic synthetic animation of 3D heads needs learning from the speaker behavior. Motion capture (Sturman, 1994) is often utilized to obtain specific animation parameters that are studied to customize human animation for synthetic representation. Some motion capturing techniques use hardware-based methods like electric or electromagnetic captors (Motion capture websites, 2002), some others utilize an image-based approach and they generally analyze video image input from the person's face to study its movements and expressions. Hardware-based methods can give very precise results but they are expensive and complicated to operate. Most of the image analysis algorithms that accurately retrieve motion information are developed to work on images recorded under known and stable conditions. Furthermore, some of them need some face markers to track the person's movements. These algorithms are generally too computationally demanding to perform in real time. All these approaches are commonly used by the entertainment industry for the creation of virtual characters (Thalmann, 1996) but they suffer from too many constraints to be used in telecommunication applications.

If we are animating a clone, the analyzed parameters must respond to the customized movement of a specific individual. Fast motion generation for the clone could give out global action in the form of more or less general parameters (higher level semantics) for a given FA; if that is the case, it means that the utilized FA system already contains the customized animation information about the person being cloned. If we are capable of generating very low-level actions then the FA system will need to keep less information.

Clone synthesis and animation from video analysis

In the literature, we find three main 3D-head animation techniques matching suitable face motion analysis:

(i) **animation rules and feature-based techniques:** these methods are based on *parametric face* models which are animated by a few parameters directly controlling the properties of facial features, like the mouth aperture and curvature, or the rotation of the eye-balls. Face analysis provides some measurement data, for instance, the size of the mouth area,

and some animation rules translate these measurements in terms of animation parameters.

(ii) **wire frame adaptation and motion-based techniques:** motion information, computed on the user's face, is interpreted in terms of displacements of the face model wire frame, via a feedback loop. These techniques have proved to be very precise, especially when a realistic face model is used. However, they are generally slow because they use iterative methods.

(iii) **key-frame interpolation and view-based techniques:** face animation is done by interpolating the wire frame between several predefined configurations (*key-frames*) that represent some extreme facial expressions. The difficulty of this approach is to relate the performer's facial expressions to the key-frames, and find the right interpolation coefficients. This is generally done by view-based techniques, which use appearance models of the distribution of pixel intensities around the facial features to characterize the facial expressions.

Regardless of the technique used to derive the animation actions, realism is never lost as long as minimal action units are shared amongst systems and, if being transformed, they remain in the group *R* of realistic actions. Face cloning animation may completely lose the cloning specifications if, when adapting the semantics of one FA system to another, the minimal actions concretely attached to the individual's behavior are lost during the adaptation. We can conclude that the degree of detail of the described motion plays a very important role in face cloning.

II.4 Privacy and Security Issues about Face Cloning: Watermarking Possibilities

Synthetic objects can be used to create videos and images. In this new scenario, watermarking techniques become useful for several purposes. In particular, it could help viewers to check the creation origin (synthetic or natural) of an image/video object, to determine if the use of a given object is legal or not, and to access additional information concerning that object (e.g. copyright, date of creation, and so on.).

Privacy and security concerns will increase as soon as the use of tools able to manipulate hybrid media (synthetic and natural) will reach the general public. Highly realistic clones will add uncertainty to all visual recordings. Very realistic synthetic avatars are already a reality. For instance, people from Vir2elle (2003) are able to synthesize talking heads that cannot be distinguished from recorded video. The algorithmic techniques behind Vir2elle technology can be found in (Cosatto & Graph, 2000). In a not so far future, it could become possible for an anonymous person to usurp someone else's identity by making *his clone* act, say and behave in a certain way on a video or image.

Classical 3D watermarking algorithms (Benedes, 1999) focus on the protection of the computer representation of the 3D object (via its geometry data); they cannot protect its usage (i.e., the set of all possible synthetic images generated from the model).

It seems more interesting to develop algorithms that aim at protecting the use of a 3D model by watermarking its associated texture. Dugelay, Garcia, and Mallauran (2002) propose to embed a watermark into the texture of the model, and then to recover it from the synthesized model image views. The recovery of the mark from synthesized views would be a more or less valid proof (depending on the current performance of the algorithms) that the scene is synthetic. This algorithm is resilient to any modification of the internal representation of the 3D model since it is based only on the views of the object and on some arbitrary original representation of the object used to embed the watermark (which needs to be known for the mark recovery).

III Investigated FA Framework for Telecommunications

In this chapter we develop the practical aspects of facial animation and more specifically face cloning viewed from the perspective of deploying telecommunication applications. We describe how facial expression analysis techniques developed and studied in this thesis are framed inside an analysis/synthesis cooperation scheme whose main objective is to achieve face cloning for teleconferencing purposes.

We start exploring the practical scenario under which we have developed our algorithms, also reviewing current trends in how to deploy facial animation in telecommunications. Then, we discuss some technical issues related to the use of facial animation for telecommunications and we present how our framework allows some interesting networking performance evaluations.

Finally, we introduce the facial motion and expression analysis procedure described inside the proposed framework.

III.1 Introduction

The range of possible and expected applications for systems aiming at deploying highly realistic facial animation is wide. Facial animation can be useful to video communicate via newer and more flexible communication links such as, the Internet or mobile telephony, which do not have high bit rate capacities and cannot ensure nice quality of service. Next generation mobile communications already contemplate the possibility of face-to-face conversations. E-commerce using virtual sellers enhances the contact with customers by using face-to-face human computer interfaces. The game industry can also benefit from using clones of players instead of simple avatars. Finally, some advanced communication systems involving several persons (video teleconferencing) could be designed to reduce the feeling of distance between participants by introducing some elements that exist in real meetings when creating artificial but realistic work discussion environments. In this sense, our thesis research has been done following the spirit of developing more advanced teleconference sites. It is important to notice that until now, all the applications mentioned before have preferred using avatars rather than animating insufficiently realistic artificial faces. This justifies the great effort and resources put into face cloning research and the relevance of the thesis work herein exposed in current telecommunications.

To generate face cloning we need a highly realistic 3D model of the speaker; and we should be able to realistically animate it. In addition to that, the animation of the model is enslaved by, or reproduces, a certain reality because the goal of cloning is to reproduce the behavior of a real speaker. Contrary to avatars or talking heads (even if realistic), face cloning implies for the complete system of animation generation to be speaker-dependent. This domain falls in the larger category of virtualized reality, in opposition to virtual reality since the realism of the restitution is not reached from scratch by advanced computer vision techniques but inspired and constrained by resting on real audio-visual data of the speaker. Face cloning is a relevant illustration of the recent phenomenon of convergence between different research domains: image analysis (i.e. signal processing), image synthesis (computer graphics), and telecommunications.

Modeled synthetic faces are animated following the actions derived from the interpretation of some animation parameters. Generating face animation parameters becomes a difficult task if done manually; therefore automatic or semi-automatic parameter generation systems have been developed. These systems extract face motion information from either the person's speech, image or both. Visual Text-To-Speech (Visual TTS) synthesizers, which refer to those TTS that also provide face synthesis, generate their animation parameters from the input text given to the TTS. The Visual

TTS analyzes the text and supplies the corresponding phonemes. These phonemes have their matching synthetic mouth motion representation, also called visemes, which can be synthesized. Visual TTS present several advantages: they are the most simple analysis systems to generate face animation parameters, they do not need human interaction and they can generate quite accurate mouth movement. For these reasons, some of the current face animation products available to the public use this technique. We can also utilize the duality phoneme-viseme to derive animation from speech. In this case, speech is analyzed to deduce the phonemes. Whether we extract the phonemes from text or from speech, the major drawback of these methods is that they only generate automatic movement for the mouth therefore some other source of action generation is needed to complete face animation. They give acceptable results when animating non-realistic characters (cartoons, animals, etc.) but since their generated parameter information is not speaker dependent, they hardly give a natural human feeling.

We can obtain improvement in realism and naturalness by customizing the synthesis based on someone's face motion and expression. Motion capture is often utilized to obtain specific animation parameters that are studied to customize human animation for synthetic representation. Some motion capturing techniques use hardware-based methods like electric or electromagnetic captors, some others utilize an image-based approach and they generally analyze video image input from the person's face to study its movements and expressions. Hardware-based methods can give very precise results but they are expensive and complicated to operate. A software-based alternative for customizing animation is the use of image-processing algorithms to accurately retrieve motion information from videos of the speaker. These algorithms are generally developed to work on images recorded under known and stable conditions. Furthermore, some of them need some face markers to track the person's movements. Some solutions can generate information to synthesize complex animation but they are computationally heavy and they cannot perform in real time.

In addition to make more realistic and natural generic facial animation, we also need motion analysis techniques to instantly study the actions of the speakers at a given time. When using FA in communications, the application environment requires real-time non-invasive analysis methods to generate facial animation parameters; therefore most of the approaches taken to customize facial animation systems are no longer useful to be applied to communications.

III.2 Framework Overview

Figure III-1 illustrates the system that we propose for facial motion and expression cloning. During the analysis (green print) analyzed information from the speaker, mainly visual although it could also be of different origin, is obtained and used to replicate the facial behavior (denoted by λ), on a highly realistic 3D-head model. The generated parameters could be encoded and sent to be directly interpreted; instead, it is preferable to simulate the decoding and synthesis during the image analysis. Adding this synthesis feedback, we can control the error committed and we can adjust the analyzed parameters to adapt them to a more accurate motion (α). The final data (μ) must be understandable by the facial animation engine of the decoder in the remote site (orange print), following the specific semantics or maybe after having been adapted to a standard. The use of a highly realistic head model of the speaker enables us not only the use of a convenient and exploitable visual feedback but also the knowledge of anthropometric data that can also be utilized during the analysis.

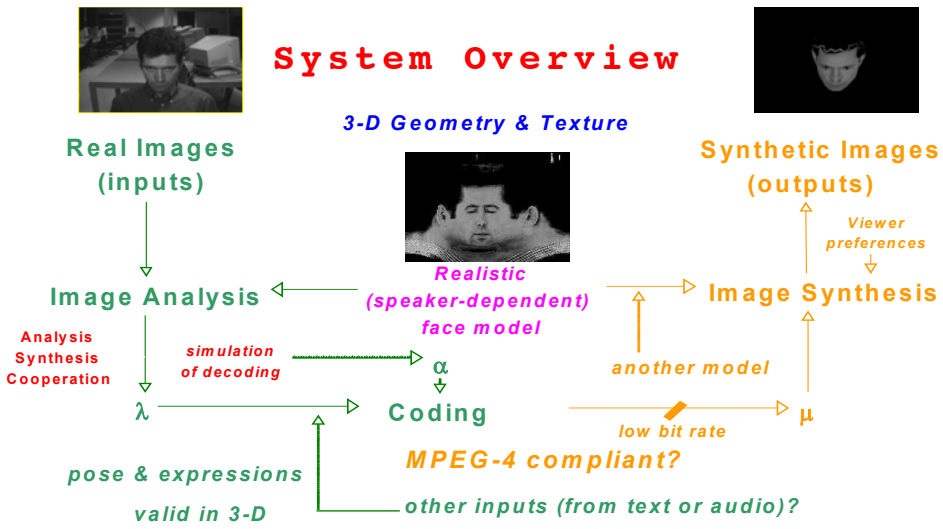


Figure III-1. When using clone animation for communications, there exist two main active parts. The facial animation parameter generator (green print), which is included in the encoding/transmission-part and does heavy image processing; and the facial animation engine (orange print), which is located in the receiver and whose task is to regenerate the facial motion on the speaker's clone by interpreting faps. The framework herein presented uses synthetic clone image feedback to improve the image analysis and generate more accurate motion information

Basically, the complete development of this system contains 4 major blocks:

(i) acquisition or creation of artificial, speaker-dependent and realistic 3D synthetic head model of the speaker;

(ii) analysis of video of a speaker recorded in a natural environment to extract parameters for animation;

(iii) compression and transmission of parameters between encoder and decoder;

(iv) synthesis of the 3D model and its animation from the received parameters.

The main core of the research work presented in this thesis: the pose-expression coupling strategy for facial non-rigid motion analysis has been developed to suit the requirements of block (ii).

III.3 Our FA Framework from a Telecom Perspective

Until recently, visual interpersonal telecommunications have been using only video and audio coding and transmission techniques. Video quality depends on the available communications bandwidth. The performance of video techniques noticeably decreases when only very low rates are attainable, as it happens, for instance, in the Internet and the Mobile networks (Dubuc & Budreau, 2001). Using face animation methods becomes an alternative to video transmission because face animation parameters can replace images, thus reducing the amount of information to be sent. Even when high bit rates are available, current videoconference systems cannot provide a natural immersing feeling. Video communication among more than two people becomes uncomfortable. As already discussed by Dugelay, Fintzel and Valente (1999), current trends in research focus on developing teleconference systems where speakers will enjoy a more natural communication environment. If these Telecom systems succeed in developing face cloning, they could provide two main advantages compared to classical audio-video communication systems:

- they would simulate the same visual result, requiring lower transmission needs (bandwidth);
- the synthetic nature of the visual reproduction would permit offering extended services through original functionalities for the viewers, such as modification of point of view, language translation of speech with adjustments of lip motion, etc.

Rough evaluation of the use of FA on a telecom network

To illustrate the advantages of using face animation, let us evaluate what would be the load of the FA data in a telecom system when wanting to have the visual moving representation of the speaker. This example is suitable for applications like mobile telephony and tries to compare two very extreme situations: no compression at all or plane data compression against facial animation coding. Currently there exist coding solutions like the H. 264 standard for video coding that give outstanding results in video coding but, unlike facial animation coding, do not introduce yet 3D information coding possibilities (see MPEG-4 for details).

- Downloading the model (raw complete size: ~1 MB):

** Texture:

RAW:	593519 bytes
(1) JPEG compressed (100% quality):	204206 bytes
(2) JPEG compressed (50% quality):	27137 bytes

Note: The visual difference between (1) and (2) may not be significant for some applications, nevertheless the compression rate difference is not negligible.

Estimated time of transmission at 10 kb/s:

RAW:	474.81 s ~7 min and 55 s
(1):	163.36 s ~2 min and 44 s
(2):	21.7 s

At first glance, only the last case seems to be reasonable. Nevertheless, the size of the texture could be customized to the application's requirements.

** 3D mesh (in the form of ASCII values):

High quality – 5000 vertices:

Uncompressed:	450000 bytes
ZIP compressed:	165367 bytes

Estimated time of transmission at 10 kb/s:

Uncompressed:	360 s ~6 min
ZIP compressed:	130 s ~2 min and 10 s

Reducing the number of vertices by a factor of 6, the ZIPped mesh would take 21.6 s to be transmitted.

Mesh and texture are the largest data of the complete FA system. A light solution could be to keep the models already stored in the receiver. In any case, the advantage of

the system is that once the model and the mesh have been sent, they remain the same for future communications and they do not have to be retransmitted.

- Animating the model:

We will just evaluate the animation of the model as a whole, that is, applying on it only rigid motion parameters. It is easy to extend the results to more complex animations by adding extra parameters to be sent. For global pose tracking we can use six parameters. Three of them determine translations along the x-, y- and z-axis, and the other three represent the rotation degrees of the head with regard to these same axis.

Assuming that each parameter is stored in 2 bytes (there is no compression). If we want to achieve 10 f/s (understanding frame as the difference in movement determined by the parameter change), we would need: 2x6x10x8 = 960 bit/s ~ 1kbit/s.

A non-compressed b&w video of size 384x288 pixels (coded with 8 bit/pel) would need 8,847,360 bit/s ~ 9 Mbit/s. In the high-quality video, we could also appreciate other face movements besides the pose change. Nevertheless, we face a ratio difference of almost 10000 between systems, which could be understood in the following ways:

- To achieve the same bit rate, at high quality, the image to be sent should be 10000 times smaller. In practical scenarios, video frames would be too small to appreciate anything (10000 times smaller is just impossible). We must recall that the quality of the synthesized faces does not depend on the transmission rate but on the FA system used, the model and texture quality, the coding of the components, and rendering used. Therefore can almost be considered size independent.
- To achieve the same bit rate, maintaining the size of the image, the video sent should be compressed by a factor of 10000. Assuming that we can obtain such a compression, there are many chances that the quality received would be so low that all the details of the face would have been lost, and moreover, there would be an overall displeasing feeling due to the general loss. In fact, this is the main argument for the development of model-based coding techniques of compression: video compression techniques trying to use generic 3D models to be able to compress videos even more. They can be considered to be half way from classical video to Face Animation synthesis.
- The FA system could use around 5000 parameters more to improve its animation. Current avatar systems utilize much less than 1000 parameters to animate faces and they give out quite pleasant face motion synthesis.

> Transmitting so many parameters, although loading networks as much as video would, still has the advantages of having synthetic motion reproductions, such as, changing the speaker's point of view, integrating the speaker in different backgrounds, etc.

Considering this example from a mobile-telephony perspective, as handsets are getting more sophisticated and they have more means of storing data in their memory, FA, in the form of avatars and maybe later, more realistic 3D head animations, are a viable application to be deployed on 3-G mobile telephones.

This example alludes to the fact that face animation communications involve the coding and transmission of the modeled representation of the face (a highly realistic 3D mesh for clone communications and its material/texture characteristics) and the set of face motion parameters that animate the model. We consider face animation parameters (faps) as the practical specification of the motion actions of a concrete FA system.

When a given system is made to achieve the most realistic animation that is possible, the generated analyzed motion data must be completely understandable, that is, it must suit the parametric semantic requirements of the face synthesis in use. Proper coding and transmission techniques must prevent communications from altering the generated data flow so the sent parameters remain coherent for the synthesis.

III.3.1 Coding face models and facial animation parameters: an MPEG-4 perspective

Each different analysis-synthesis scheme has its own way of handling head model and animation data. To tackle coding and transmission issues, most systems elaborate their own solutions. For example, Varakliotis, Ostermann and Hardman (2001) propose a method for 3D mesh animation coding. Proprietary approaches may fulfill single application needs but they cannot ensure intercommunication amongst different face animation encoding-decoding systems.

The new multimedia standard, MPEG-4, tries to standardize the way natural and synthetic animation data are coded so they can suit global communication needs. We refer the reader to (MPEG-4, 2000) for a complete overview on MPEG-4. The standard covers a wide range of multimedia possibilities and includes face and body animation as separate items to code. When using face clone communications, what is the advantage of using specific standardized coding for face animation? It is basically to achieve high compression and to obtain complete interoperability with other different communication systems. What are the main constraints of using MPEG-4's specific coding? This

standard specifies common syntax to describe face behavior. This syntax has been created in such way that MPEG-4 considers face animation as specific case of parametric[1] 3D-face motion synthesis. Therefore, we find that it is intrinsically tied to its own *semantic level* of animation. To ease the standardization among different FA systems, 3D-head models are required to contain certain specific vertices or Facial Definition Points (FDPs) and motion is coded in the form of 68 Facial Animation Parameters (FAPs). One FAP generally represents the magnitude of the vertex motion related to a given action, although it can also describe a higher-level action, being then an expression or viseme[2] parameter. For an MPEG-4 compliant terminal to interpret FAP values, it may use animation tables (FAT), one per each FAP, to associate magnitude values to action motion for the head model in use. A more extended explanation on how Facial Animation is tackled inside the MPEG-4 standard can be found in Pandzic and Forchheimer (Eds.) (2002). We refer the reader to Section 2.2 in Chapter VI of this thesis dissertation, where we detail how the standard has been applied to make the models used in our experiments MPEG-4 compliant.

It is not clear yet which is the degree of realism that such a coding standard permits. MPEG-4 FAPs are not the most minimal actions FA can universally understand (if such actions exist), therefore some different face animation systems, based on more complex parameters or different animation techniques, may find the coding requirements too restrictive. Systems whose face animation parameters do not directly match MPEG-4's face coding syntactic and semantic structure must reinterpret their face animation rules in terms of FAPs and FATs. Figure III-2, which displays the communication process, illustrates how before encoding, animation data must be *transformed*, $T\{\ \}$, to become complying FAP magnitude values. Likewise, MPEG-4 decoded FAP streams will have to undo this transformation, $T^{-1}\{\ \}$, to obtain comprehensible animation data. If these transformations are possible, no loss of motion information can only be guaranteed if the degrees of action permitted by the analysis-synthesis system are fewer than the action restrictions imposed by the coding.

If coding following MPEG-4's Face Object requirements becomes too constraining, the standard also manages the coding of 2D, 3D meshes and textures, whose use may be more convenient in some communication scenarios.

[1] Understanding parametric face synthesis as the face animation synthesis where actions are described in terms of magnitudes that represent the physical displacements of the vertices of the 3D-mesh face model.
[2] Viseme: head model deformation associated to the reproduction of a specific phoneme.

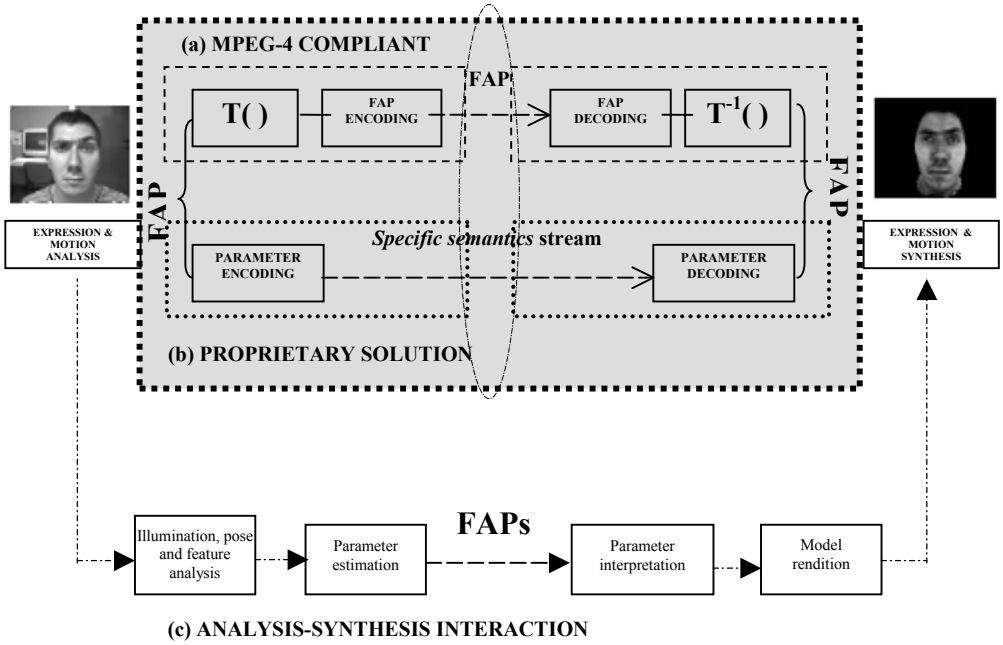


Figure III-2. Face animation parameters (FAPs) are the generated numerical values associated to the facial motion we want to synthesize. We are interested in building encoding and decoding techniques to efficiently compress and transport animation data over different telecom networks. Proprietary solutions (b) ensure perfect communication between motion generation and synthesis. Using standardized solutions, for instance, MPEG-4 coding (a), enables interoperability amongst different systems, at the expense of readapting animation to the encoding requirements and maybe losing animation precision. Teleconferencing systems (c) are an example of applications that would profit from the introduction of facial animation analysis and synthesis.

III.3.2 Facial animation parameters transmission

The main requirements for the transmission process in face communication applications are: accurate audio and parameter flow synchronization, minimum delay to avoid latency disruptions and minimum loss of information because the action data transmitted has been already much reduced. These conditions are hard to achieve in very-low bit rate networks and effort is being put in developing efficient streaming methods.

If face animation intends to clone the behavior of a living person, the overall communication should be completely transparent to the FA system. This implies that we must use lossless coding techniques and that the transmission should not alter the sent data. This last point is most of the times out of control of the most commonly used telecom systems (mobile & the Intenet) and thus, almost impossible to achieve. Nevertheless, it is possible to minimize its effects on the animation (as it has already been done for other media).

Approaches for very-low bit rate networks

The generalized use of packet networks makes the Internet Protocol (IP) employment extensive to other domains, e.g. 3-G Mobile Networks (Platon, 2000). Streaming data over IP will eventually ensure applications to be independent of the physical network.

Novel teleconference systems use multicast for efficient, real-time, point-to-multipoint delivery. From the two transport layer protocols available in the IP suite, TCP and UDP, UDP is the most convenient for multicast applications. Since the latter does not guarantee timely and reliable datagram delivery, RTP (Schulzrinne, Casner, Fredderick & Jacobson, 1996) is used to provide such capabilities end-to-end to the application layer, by use of timestamps, sequence numbers and payload identification, among other header fields.

Humans are very sensitive to the synchronization of voice and lip movements. Accurate audio synchronization must be achieved when generating face animation parameters. Transporting FAP data over packet networks may result in delay variations and packet loss or reordering due to increased congestion levels, therefore protocols involved in the transport must ensure resynchronization of FAPs at the receiver. Relevant work has already been developed for lip synchronization between audio and video streams over RTP (Kouvelas, Hardman & Watson, 1996; RAT & VIC, 2002). RTP/UDP streaming seems the natural way to transport face animation data. Tackling synchronization, delay, buffering, error concealment, etc. for facial data becomes an extension of the work already being developed for video and audio streaming.

RTP has also been the protocol of choice for the delivery of MPEG-4 media over IP (Avaro et al., 2001). For MPEG-4 each media has an associated Elementary Stream. Media transport has been addressed in a joint effort with the Internet Engineering Task Force (IETF). Although Face Animation information is considered an Elementary Stream, how to encapsulate face animation data is still an open issue.

One-to-one conference communications: a practical example

In this practical example, we wanted to test the possible capabilities of a basic transmission system and evaluate the weaknesses and strengths of transmission approaches already used for other media, in particular, audio.

Our implementation is based on RTSP (Real Time Streaming Protocol), which is currently used for streaming applications, and defines a specific format for sending FAPs

(following MPEG-4's specifications) through an IP network. It enables to stream FAPs over the network to any MPEG-4 compliant face animation terminal.

To generate the FAPs we use automatic video input analysis of the speaker's face (explicitly its head pose and its eye actions). To render the animation we have a 3D-model viewer, where an avatar capable of understanding the incoming FAPs is animated.

The video input analysis and face motion synthesis have already been positively tested working together on the same machine (a PC running *Microsoft* Windows™ NT). They interact instantly, the FAPs are animated by the model on the viewer with no delay, thanks to the use of *Windows internal messages*[3] among windows. More detail about the analysis/synthesis implementation and the related algorithmic background are given in Chapters IV, V and VI. FAPs are not compressed and it is not the purpose of this test to evaluate different coding methods. This study will only consider the transmission procedure designed from classical audio streaming procedures applied on FAP streaming.

Some previous approaches

In their work, Haverlant and Dax (1999) describe a networking strategy for a point-to-point communication in avatar worlds. The Face Animation Parameters are proprietary, which excludes any interaction with other avatar systems. In Chen's work (Chen & Kambhamettu, 1997), MPEG-4 compliant FAPs are transmitted using RTP on UDP, with a self-defined payload type, which excludes any interaction with other MPEG-4 terminals. Chen's main concern is to explore multicast and graphics compensation algorithms, and not inter-operability. In our work the whole teleconferencing system, from the extraction of the FAPs to the clone animation, including the sending of movement features, was designed to be as much MPEG-4 compliant as possible

RTP streaming for real-time FAPs

RTP is the chosen protocol for the delivery of MPEG-4 streams over IP. In order to use it, we need to define the payload associated to the specific stream we want to transmit. Payloads are still in the process of standardization. Having "one payload for one media", that is, encapsulating each MPEG-4 Elementary Stream by means of individual payloads, is the most flexible way to handle streaming but encourages the development of very different payloads.

We decided to use the Phoneme and Facial Animation (PFAP) RTP payload format presented by Ostermann, Rurainsky and Cinvanlar (2001) as a draft to IETF. In

[3] We refer the reader the *Microsoft* Windows Developer's website: msdn.Microsoft.com.

their proposal the authors describe a payload for transporting phoneme and facial animation parameter (PFAP) streams in RTP packets. In their article, Ostermann, Rurainsky and Civanlar (2002) discuss its performance when streaming the output of a visual TTS[4]. We have chosen this payload for several reasons: it already contains a recovery strategy for loss-tolerant transmission of these streams, it includes not only plain FAPs but also phonemes (leaving an open door for lip motion coding in the form of phonemes) and, since it was intended to be used by visual TTS applications, it would make our systems directly compatible with them. We must point out that the proposal expired in April 2002, and no further action was pursued, therefore has not been officially accepted.

The PFAP payload consists of three types of information: phoneme descriptor, FAP descriptor, and recovery information (Figure III-3). Each payload starts with a packet descriptor field followed by optional recovery information. Phoneme descriptors and FAP descriptors may follow the packet descriptor or the recovery information if available.

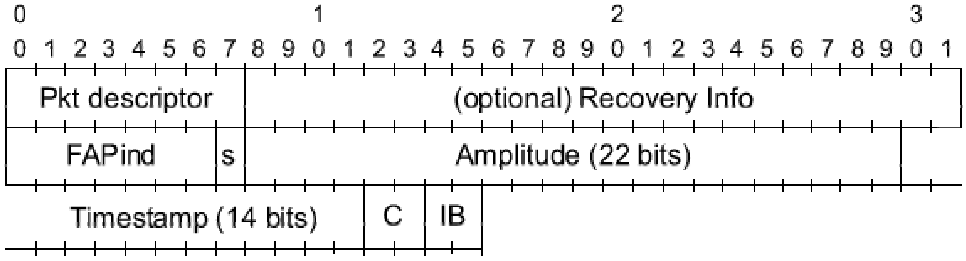


Figure III-3. Packet Descriptor of the PFAP Payload. Image courtesy of Ostermann, Rurainsky and Civanlar (2002)

FAPs are associated with phonemes to determine their timing in a sentence. The start time of a FAP is the same as the start time of the first phoneme following the FAP. Therefore, a packet must end with a phoneme if it contains any information other than recovery information. In the PFAP payload, it is still possible to send FAPs without phoneme descriptors, by including the timing information in the transition field. And in this way, we are able to profit of this payload in our transmission system.

Settings

The original analysis-synthesis application was designed to work on a standalone desktop. The analysis module analyzes a real-time video, or a saved video sequence,

[4] TTS: Text-to-Speech

corrects the illumination, predicts the user's head pose, analyzes the eye features and converts the data to MPEG-4 compliant FAPs[5]. The model viewer module instantly receives FAPs and renders the animated face model. This application is meant for future use in teleconferencing. The test-oriented setting of having analysis and synthesis in a single location had to be changed to turn it into a more teleconference-like platform. To do so, we decided to provide a network frame that would allow one-to-one videoconferencing over the Internet network. For our development we supposed that the receiver already has the head model, its texture and the associated FATs.

To implement the transmission, the functionality of the existing analysis-synthesis application was extended so that streaming could be supported. This was achieved by separating the original application in a client/server structure. Naturally the analysis module, which was sending the media elements, was chosen as the Server, and a FAP streaming server was added to it. The Synthesis module acts as a remote control, asking for PLAY/PAUSE of FAP streams and is the Client, an FAP streaming client was added to it, see Figure III-4.

Since RTSP provides some of the functionalities required to develop a streaming version of the analysis-synthesis application, we decided that this protocol would be used in addition to RTP.

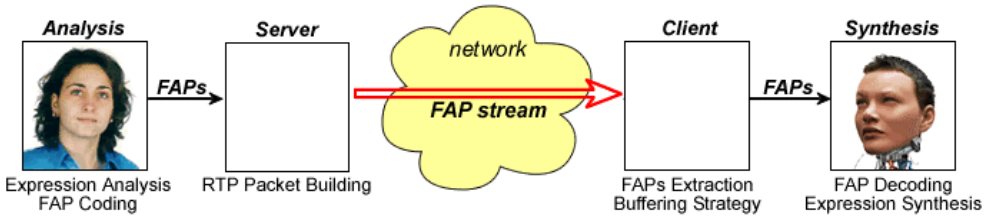


Figure III-4. High Level Networking Architecture

Design aspects

Our classical RTSP-implementation initially streamed audio files, with a wave format in our case. The aim of the implementation was to extend its capabilities to stream FAPs, and integrate them into the analysis-synthesis application.

During the merging of capabilities, our main concern was not to interrupt the application layer functions (analysis of video input for the server and synthesis of the model for the client) with the networking functions (connection, transportation, synchronization, …). Threads always provide good ways to isolate and separate tasks and execute them simultaneously. The following processes need to run simultaneously:

[5] The developed algorithms for the facial analysis involved can be found in Chapter IV, V and VI.

- (Server) Adding generated FAPs into the Input Buffer
- (Server) Reading the Input Buffer and Sending the elements
- (Client) Receiving FAPs and Adding to the Output Buffer
- (Client) Reading the Output Buffer and playing the FAPs

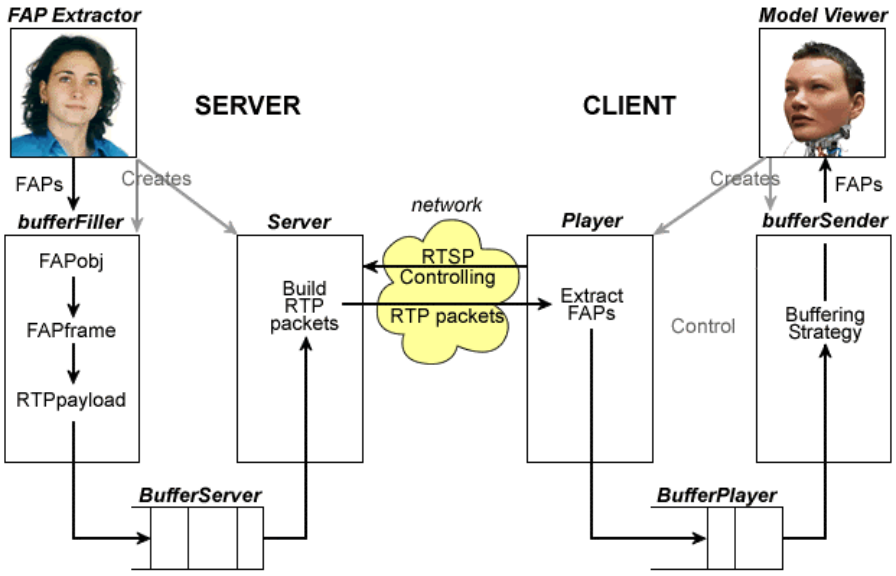


Figure III-5. Detailed description of the complete networking capabilities of the Server (analysis) and the Client (synthesis)

Server

The Server works as an asynchronous RTSP server, it is waiting to receive data and is able to stream audio/WAV and audio/AU packets, and it has also been extended to deal with FAPs by defining its own MIME type: MIME_FAP.

MPEG-4 Client

The Client implements the minimal standard RTSP control functions SETUP/PLAY/PAUSE/TEARDOWN. The reference RTSP client recognizes the audio/PCM and audio/PCMU MIME types. The additional MIME type was created for FAPs. The received stream of FAPs is added to the Output Buffer, and played out using the time-stamping information.

Buffering Strategies

We implemented the Fixed Playout Strategy in the Output buffer for sending the FAP frames (group of FAPs obtained from the analysis of one video frame) to the model viewer. According to this strategy, the player attempts to playout each frame exactly *d* ms (we fixed it at 250 ms by default) after it is generated. So if a frame which is

contained in one RTP packet is time stamped at time *t*, the player plays out the FAP frame at time *t* + *d*, assuming the frame has arrived by that time. Packets that arrive after their scheduled playout times are discarded and considered lost.

The differences between the Instant Playout Strategy and the Fixed Playout Delay Strategy are illustrated in Figure III-6. This figure shows the time at which packets are generated and played. As shown by the left staircase, the sender generates packets asynchronously. The first packet is received after a time that corresponds to the current network delay. For the first playout schedule the fourth packet does not arrive at the time specified by its timestamp because of the network jitter. In the other hand, the Fixed Playout Strategy enables to play out this same packet respecting its timestamp relative to the previous packet.

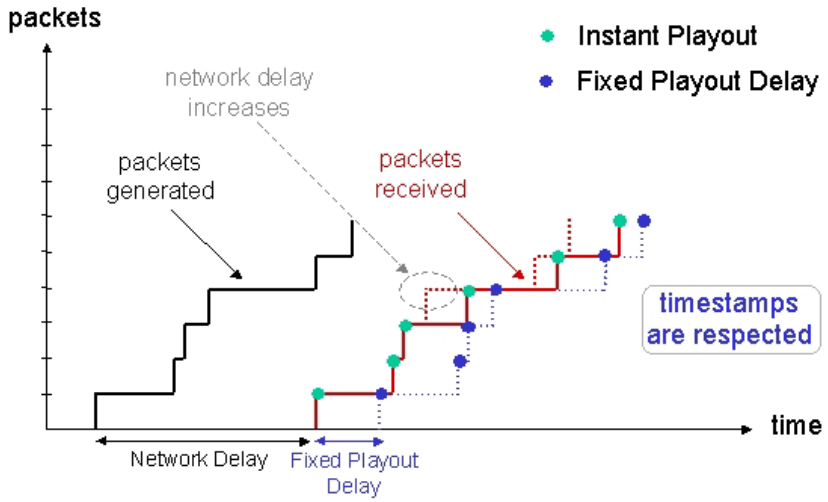


Figure III-6. Comparing buffering strategies

What is a good choice for the play-out delay? By making the initial delay large, most packets will make their deadlines and there will therefore be negligible loss; however long delays can become bothersome if not intolerable, especially for audio. Internet phones can support delays up to about 400 ms, and then many packets may miss their scheduled playback times. Roughly speaking, if large variations in end-to-end delay are typical, it is preferable to use a large delay. Nevertheless if delay is small and variations in delay are also small, it is preferable to use a small fixed playout delay, perhaps less than 150 ms.

The timestamp reference of each of the FAP frames is obtained by extracting the time reference right after the analysis of face motion and expression has finished. For real-time applications the difference between two consecutive FAP-frame timestamps should be smaller than 1/*fps*, where *fps* is the frame rate of the video input. In practice,

the analysis constrains the speed of FAP generation. Nevertheless, transmission must ensure that the model viewer client has received enough frames to be able to play smoothly.

Conclusions

After implementing, we were able to validate and test our design choices. Clearly the choice of RTSP is not optimal, because we do not have enough controls on the server side, or better expressed, the controls are suitable for classical video and audio applications but not suitable for virtual videoconference applications. RTSP is a standard and its use is advisable unless new shortcomings demonstrate to perform better. Nevertheless it likes suitable controls for virtual videoconferencing related to the 3D nature of the application: location of the speakers, 3D environment control, etc. The choice of RTP proved to be more optimal, as it is efficient and connectionless. Choosing PFAP payload as our application payload was not optimal because at the moment of the implementation only 40% of the payload information was used. We could remove all duplicate and unnecessary headers but, in turn, our implementation would not be able of streaming FA parameters over the network to any MPEG-4 compliant face animation terminal that would be using the PFAP payload.

We built a demo application on a local network but unfortunately, not in real life conditions (no delays, no losses, etc.). A sober evaluation made by simulating packet loss and delays (by dropping FAPs and slowing down the transmission) showed some of the problems we may encounter in real life application. As it happens in video transmission, if many frames are dropped, that is many FAPs are lost the quality lowers exponentially. FA systems are even more sensitive because the animation information is condensed in the form of parameters, and once they are lost no action takes place. Losses can make the animation synthesis abrupt. From the image quality point of view, the rendering would still be visually "nice"; only motion would be affected, whereas in video transmission, visual quality is usually highly perturbed under these same networking conditions.

The main interest of our study was to show the potential of the analysis-synthesis scheme from the communications, and more concretely, the transmission perspective. There exist many open issues that are, by themselves, subject of broader and deeper research:

- The buffering effects on communications, analyzed when the network conditions change during transmission;
- the design of an efficient teleconference-oriented payload;

- the development of coding mechanisms for the FAP information adapted to the application and the communications requirements (study of the advantages and disadvantages of using MPEG-4 coding algorithms);
- the development of data recovery strategies more suitable for this kind of application; and
- the search for an alternative to the regular RTSP commands so they are more convenient for teleconferencing-like applications.

Many of these points are already being investigated, in most cases, in the context of avatar animation. The novelty of the platform used here lies in the FAP generation from video input, which permits to perform almost realistic animation thanks to the analysis of real human behavior. We believe it is the best context for simulating and studying networking issues related to the use of FA for teleconferencing.

III.4 Facial Motion Analysis: Coupling Expression and Pose

To obtain motion data from the speaker's face over video sequences in our real-time teleconference environment, we must develop video image analysis techniques robust to any environment and lighting conditions and not to impose any physical constraints. Most of the existing face expression analysis techniques are not really suitable for practical purposes. The most performing analysis systems where rigid and non-rigid motion is simultaneously analyzed utilize the synthesis of head models. For example, Eisert and Girod (1998) have improved the approach proposed by DeCarlo and Metaxas (1996) in their teleconferencing system. Their analysis technique is based on optical flow constraints to analyze and interpret face motion. Their research results are very encouraging but their analysis algorithms work under many environmental restrictions due to the use of optical flow techniques.

The process developed for the analysis of facial features is conceived for very specific uses. The main goal of the research described in this thesis is to develop robust and fast image analysis algorithms to be used in the proposed FA framework for telecommunications. In this practical setting, motion data extracted from video input is intended for real-time reproduction on a 3D synthetic head of the speaker (its clone). The algorithmic solution sought tries to fit the following requirements:

- We use currently available media for teleconference systems, that is, monocular images. Video data extracted from one camera in front of the user (for instance, web cams) without any calibration.
- We do not allow any interference over the natural environment:
  - no makeup or markers on the speaker;
  - no precise lighting conditions: simply, illumination that would allow humans to understand the motion;
  - as much freedom of movement as possible for the user, avoiding the 'near-to-front' point of view restriction, which iscommon in this kind of analysis.
- We try to avoid any training previous to the analysis, or visual knowledge of the person's characteristics that cannot be obtained from its synthetic model.

- We want to obtain motion data that are as precise as the analysis conditions can allow us, by generating an analysis strategy with improvement potential in its precision.
- The obtained data must be complete enough to be used for coherent natural facial synthesis.
- The image analysis processing will be designed to be as universal as possible.
- The algorithms and chains of processes involved in the analysis must be oriented to potentially work in real-time, thus permitting their deployment in telecommunication applications.

It is difficult to create a complete analysis process able to fit all these requirements. Algorithms that are universally useable generally lack precision. Indeed, if no previous assumptions are taken, then, making suitable the analysis to all cases implies lots of computations and therefore the loss of real-time possibilities. To compensate this restriction, we may generate less precise analysis algorithms but keeping in mind the possibility of improving the complexity of the system; as the computational requirements become less and less restrictive, a flexible analysis scheme will allow us to increase the complexity and to extract more detailed motion data.

Facial expressions can be considered independent of the head rigid motion (Bascle & Blake, 1998). Although their projected appearance on the image is not completely independent of the pose, the solution developed in this article tries to exploit the real expression-pose independence in 3D space to study ocular expressions in 2D.

We have designed a two-step process to develop image analysis algorithms to analyze non-rigid and rigid facial motion simultaneously. First, we design image-processing techniques to study the features extracted from a frontal point of view of the face. Faces show most of their expression information under this pose and this allows us to verify the correct performance of the image processing involved and the correctness of the hypotheses made for the analysis. Second, we extend our algorithms to analyze features taken at any given pose. This adaptation is possible because the motion models (motion templates) utilized during the analysis can be redefined in 3D space and the accuracy of the retrieved pose parameters is such that it enables us to reinterpret the data we obtain from the image analysis in 3D.

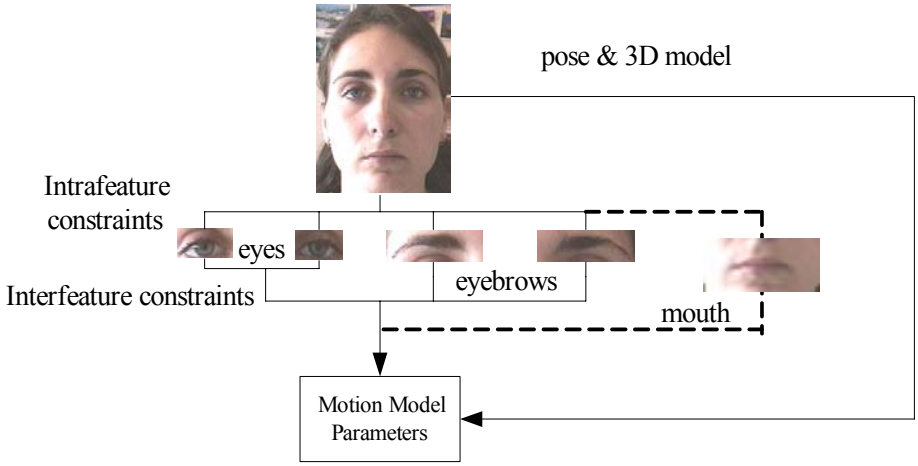


Figure III-7. General diagram of the proposed analysis framework.

Developing a video analysis scheme where head pose tracking and face feature analysis are treated separately permits to design specialized image analysis algorithms adjusted to specific needs, feature characteristics, etc. For our work on virtual teleconferencing environments, a pose-tracking algorithm based on Kalman filtering (Valente & Dugelay, 2001) was first developed. The pose tracking system permits a robust prediction of the pose of the speaker frame by frame with the help of the synthesis of speaker's clone. The analysis strategy proposed to study non-rigid facial motion profits from the pose tracking algorithms robustness and the use of a highly realistic 3D head model of the speaker.

The study of monocular images is constrained by the fact that motion information is only retrieved from one view. When analyzing faces from a single perspective, we can estimate the motion of the interesting features (eye, eyebrows and mouth) by studying the displacements of their projection on the video frame. Regarding the image-processing techniques developed to estimate the displacements and the generated motion, we have opted for applying specific and different image-processing techniques per feature. Many analysis schemes apply the same technique independently of the feature or expression they analyze. In our preliminary approach (Valente & Dugelay, 2000), we used a solution based on image correlation. We utilized PCA to build the image databases. Storing the images taken from all possible lighting conditions, global pose situations and FAP combinations became difficult for features like the mouth and the eyes, whose expressions can be quite complex. Nevertheless, the approach was fairly suitable for eyebrow movement (Valente, Andrés del Valle & Dugelay, 2001).

The image-processing techniques involved are intended to extract useful information for the motion templates that have been designed for each feature. We aim

at obtaining data under very general circumstances. To help the algorithms to behave correctly, we use intra- and inter-feature constraints derived from standard human motion. All standard actions that generalized the motion of features of the same nature are considered intra-feature constraints (for example, both eyes act the same way). Motion information from one feature derived from the analysis of another feature is considered an inter-feature constraint.

The geometrical nature of the data that controls the motion templates enables us to extend the algorithms previously designed to work on faces showing other poses by simply doing an adaptation that is completely detailed in Chapter V. The motion templates and the image-processing algorithms studied are described in Chapter IV.

Classifying the proposed methodology into one of the categories described in Chapter I, we could consider it a hybrid between methods *that obtain parameters related to the Facial Animation synthesis used* and those *that use explicit synthesis during the image analysis*. In fact, our analysis framework tries to take advantage of the strengths of both techniques. In the one hand, obtaining parameters that directly control the synthesis by using processing techniques specific to each facial feature makes the analysis more efficient; in the other hand, utilizing synthesis feedback by using the speaker's clone during the pose tracking guarantees a high degree of robustness.

Looking at the approach proposed from a coding perspective, we would like to point out that the use of realistic 3D models is only mandatory during the analysis/encoding of motion. Facial animation parameters extracted from a system that utilizes the techniques herein studied can be used on any other head model, as long as the animation system for that model shares the same motion syntax and semantics.

IV Facial Non-rigid Motion Analysis from a Frontal Perspective

We have designed image-processing techniques that study facial feature motion from a frontal point of view. This chapter contains the formal description of the motion models that have been designed (motion templates) and the natural constraints utilized (intra-feature & inter-feature). It also presents the experimental evaluation of the proposed algorithms by studying the robustness of the processing involved.

IV.1 Introduction

The developed feature motion templates do not only utilize anatomical constraints (intra-feature) to derive feature actions, they also use human natural standard motion constraints to generate natural realistic facial animation exploiting the existing correlation between eyes and among eyes and eyebrows (inter-feature). The image processing techniques proposed try to globally minimize the influence of the unknown sources of error and improve the overall behavior understanding by imposing some standard human motion restrictions on the data obtained from the analysis of each feature. They conform an analysis strategy that aims at providing coherent motion understanding that can generate reliable animation data to synthetically reproduce facial feature expressions from analyzed video input.

The developed algorithms assume that location and delimitation of the feature's (eyes, eyebrows or mouth) region of interest (ROI) are known. This assumption is realistic in the present context because, as it is explained in Chapter V, the procedure that extends the use of these algorithms to any other pose also takes care of the tracking and definition of feature ROIs.

For all the presented experimental results the location of the facial features and the delimitation of the ROIs has been made manually. In Chapter V, we describe how this delimitation is done automatically, once the analysis algorithms are coupled with the pose tracking system.

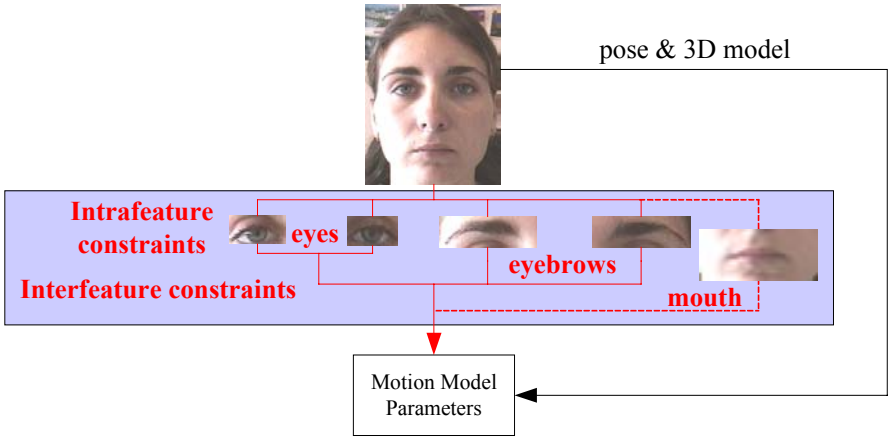


Figure IV-1. General diagram of the proposed analysis framework – The parts related to the facial expression analysis have been highlighted

When analyzing facial features from video input recorded in unknown environments, very few assumptions can be made because we cannot guarantee any determined image quality or specific lighting over the face. To create robust algorithms, the development of our sight expression analysis is based on the following premises: a) the behavior of the features to be analyzed is known and it can be modeled by understanding specific image data; b) the physical structure of eyes and eyebrows is similar for all human beings and image processing algorithms need to profit from this fact; and c) the features will be assumed to be completely visible, occlusions will be considered an uncontrolled source of misleading results, as it happens with extreme lighting conditions.

IV.2 Eye State Analysis Algorithm

The importance of eye gaze on human communication is significant. "Gaze is a richly informative behavior in face-to-face interaction. It serves at least five distinct functions (…), regulating conversation flow, providing feedback, communicating emotional information, communicating the nature of interpersonal relationships and avoiding distraction by restricting visual input" (Garau, Slater, Bee & Sasse, 2001). When developing new telecom systems for videoconferencing the correct understanding and reproduction of eye motion becomes necessary. An example is Microsoft's Research project "Gaze Master", a tool aiming at providing gaze-corrected videoconferencing (Gemmell, Zitnick, Kang & Toyama, 2000). Recently, the 10th Int. Conf. in Human-Computer Interaction granted a complete session to eye analysis (Eye movements in HCI, 2003).

Due to the vast number of applications where eye-motion understanding through image analysis is useful (eye-closing detection in vehicle driving, model-based coding in telecommunications, human actions awareness in HCI, etc.), there exist many techniques to study eye activity on monocular images. It is not the purpose of this chapter to go over all the possible methods that can be found in different fields of research, but we will overview some approaches that relate to our work in video communications.

Two major techniques have been used to analyze eye movement on images: PCA and deformable template matching (motion modeling), we refer the reader to Chapter I for theoretical details about these techniques. PCA has been widely investigated to study facial motion, above all coupled with the use of optical flow as a source of motion data (Valente, 1999). Most recent works prefer to do this analysis through ICA (independent component analysis) rather than using PCA (Fidaleo & Neumann, 2002). In both cases, their main drawback is the performance dependency on the environmental conditions of the analysis, basically on the lighting. The use of motion templates seems to be the chosen solution to retrieve eye actions robustly (Goto, Escher, Zanardi and Magnenat-Thalmann, 1999; Tian, Kanade and Cohn, 2001). Generally, these motion templates are composed by ellipses and circles, representing the eye shape, that are extracted from the images and tracked along video sequences.

If lighting independence is sought, optical flow cannot be used and other image-processing tools, analysis using mathematical morphology, non-linear filtering, etc. … are utilized. Aiming at working under flexible conditions leads researchers to look for solutions where erroneous results in the analysis – many may occur – should be compensated or minimized, for instance, by studying the temporal behavior of eye actions. Ravyse, Sahli, Reinders and Cornelis (2000) perform eye gesture analysis by

using a mathematical morphology scale-space approach, forming spatio-temporal curves out of scale measurements statistics. The resulting curves provide a direct measure of the eye gesture, which can then be used as an eye animation parameter. Although in their article they only consider the opening and closing of eyes, they already show the potential of using the temporal evolution of eye motion for its action analysis.

Our approach follows the same analysis philosophy as the one presented by Ravyse et al. It differs in the image processing involved: we propose motion deduction through the study of the pupil location because it provides both eye gaze and opening/closing information. Instead of a statistical analysis, we introduce a temporal state diagram based on human motion standard behavior that constrains motion using some intra-feature restrictions. In communications it is very important to generate non-disturbing facial expressions. As it is already discussed by Al-Qayedi and Clark (2000), the knowledge of standard human behavior can be helpful to track and animate eyes.

IV.2.1 Analysis description

Our analysis strategy decomposes the eye tracking actions in two categories: the open-close movement and the eyeball movement. Eye behavior can be described through two major actions: the open-close eyelid movement and the eyeball rotations. Non exaggerated ocular actions are characterized by the existence of a tight relationship between the pupil vertical location and the eyelid opening; therefore, we can expect obtaining most of eye motion information from the analysis of the pupil activity.

The proposed analysis scheme exploits this fact and reduces the study of eye motion to the determination of the pupil position in the eye area. Then, we assign an action state to the eye based on this position and finally we apply a Temporal State Diagram that deduces the best eye action by comparing the state of both eyes in the current frame and the states obtained in previous analyses.

Pupil search algorithm:

First, we estimate the size of the pupil inside the complete eye feature to determine the shape of the evaluation zone. Figure IV-2 illustrates the shape pf an average eye upon which we have studied the pupil size related to the overall eye ROI. Then, we perform an exhaustive scan by performing the following energy computation:

$$(X,Y) = (x,y)\ s.t.\ \arg\min\left[\frac{3}{4(2\cdot\alpha\cdot Weye)^2}\cdot\sum_{l=-\alpha\cdot Weye/2}^{\alpha\cdot Weye/2}\sum_{m=-\alpha\cdot Weye/2}^{\alpha\cdot Weye/2} I^2(x+l,y+m)\right]+ \left[\frac{1}{4(2\cdot\alpha\cdot Weye)^2}\cdot\sum_{l=-\alpha\cdot Weye}^{\alpha\cdot Weye}\sum_{m=-\alpha\cdot Weye}^{\alpha\cdot Weye} I^2(x+l,y+m)\right] \quad \forall x,y\ \ \exists\ \ x+l>0,\ y+m>0 \tag{IV-1}$$

inside this specific zone and at the same time sweeping vertically and horizontally thorough the eye analysis feature. (X,Y) corresponds to the point where this evaluation is minimized. $I^2(x,y)$ is the energy of a pixel computed as the square of its intensity component. The evaluation formula tries to look for the intensity distribution that is closest to the pupil-iris shape on the analysis area. Since the position of the head with respect to the camera and the video input characteristics may be different on sequences of diverse origin, we define α as the ratio $Wpupil/Weye$. Since α is an anthropometric measurement uniquely related to an individual, it remains constant for all analysis scenarios thus completely determining the evaluation.

This algorithm relies on the intensity information of the image, therefore it is 'a priori' dependent on the lighting conditions. This dependence is low because the pupil mainly stays as the lowest energy point of the eye thanks to the anatomical eye characteristics. Unexpected analysis results are controlled by the next steps of the process to minimize the influence of erroneous pupil detection during the interpretation of eye behavior. The specular nature of the eyeball surface may introduce very high points of energy (white reflections on the pupil/iris) that should not mislead the results. Pixels whose intensity value is considered too high are ignored during the energy evaluation.

Parameterization and interpretation of the analysis

To interpret and synthesize the results from the previous analysis technique, it is necessary to parameterize the resulting data. The parameterization process maps the X, Y values of the pupil location onto their corresponding state values.

A state value *S* is obtained as a function of the location of the pupil with reference to the width (W_{ROI}) and the height (H_{ROI}) of the eye Region of Interest that is being analyzed. To define the states we divide the region of analysis in different zones and we assign to each of them a concrete eye action (look so much up, look so much down and left, etc.). This action can be rendered if each of the states can be synthetically reproduced. It is assigned following these hypotheses: (a) eyelids partially cover the irises and vertically follow the pupil motion; (c) both eyes behave alike; therefore pupil motion is correlated; (d) pupils remain always the darkest part of the eye, when it is open,

regardless of the lighting conditions; (b) the absence of pupil indicates that the eye is closed, in such a case, the darkest point falls on the eyelashes.

The assignment of states is at the same time a 2D quantization process where the exact X, Y location of the pupil is mapped onto the closest pupil location for a given state. The number of possible states is limited to the size of analysis area and to the accuracy of the pupil search. When analyzing very small eye features or under very extreme lighting conditions, it is advisable to assign few states to compensate the instability of the image processing output. In Figure IV-2, a simplified diagram of eye ROI subdivision in presented. The ROI is divided into nine state zones on which we have assigned an eye action: look right & down, look center & up, etc., and where pupil and eyelid motion are correlated. We also define an extra state, S_{close}, which is assigned when Y value of both eyes is at its lowest point.

Synthetic motion generated from the analysis of eye behavior must recreate natural human action. Unexpected analysis results must not lead to unnatural eye motion that would interfere in proper communications. To avoid generating unpleasant eye synthesis, we filter the individual state with a Temporal State Diagram (Figure IV-3), which assigns the most suitable common state assuming the same behavior in both eyes and using previous results to compensate for misleading analysis. We first input the pupil location as a State (one for the left eye and one for the right eye), we compare both states, if they are alike, we determine that we have analyzed right, otherwise the filter starts comparing both States together and then both states with the state of the eyes on the previous frame until the filter matches a correct analysis.

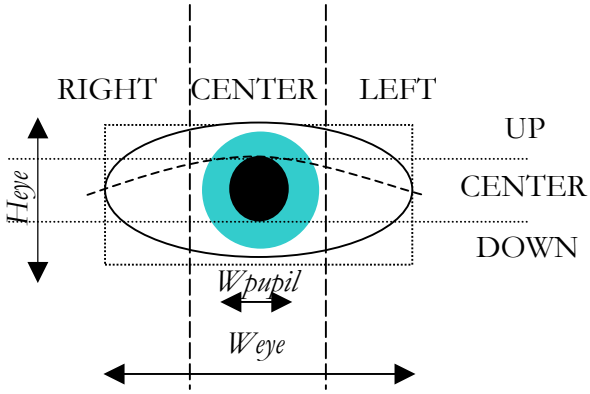


Figure IV-2 This diagram shows the simplest subdivision possible for the eye ROI so to extract meaningful motion information regarding the eye actions

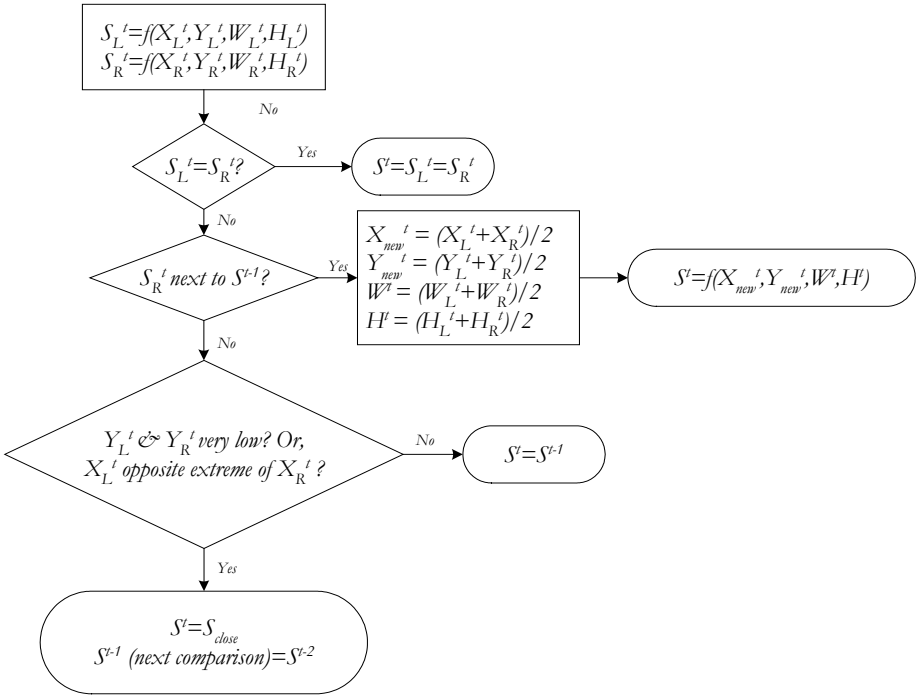


Figure IV-3. The analyzed results of each eye feature are filtered through this Temporal Diagram. Current eye states S_L^t and S_R^t are contrasted to obtain a common state for both eyes: S^t. Since the state S_{close} does not have any physical information about the pupil location, it is ignored for future analysis in the temporal chain. The starting state is fixed with the X, Y of the eyes in their neutral position

IV.2.2 Experimental evaluation and conclusions

To evaluate the complete procedure, some video sequences of an individual who was rigidly standing facing the camera have been used. The person was asked to move his/her eyes in a natural manner (up, down, left, right & closing eyes). Different lighting was used for each recording. One set of sequences was shot under natural standard conditions: neon lighting. Another set of sequences was recorded under extreme lighting conditions: direct light coming either from the front, from the right or the left side. The average length of each sequence was 500 frames, and the average eye ROI was 32x24 pixels.

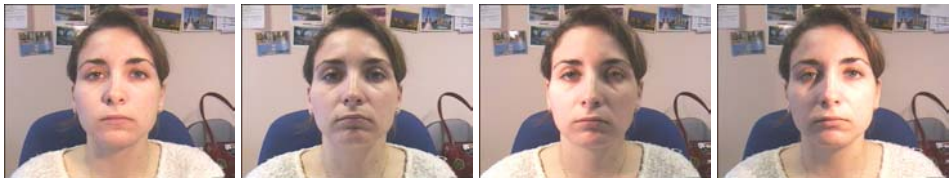


Figure IV-4. Four frames extracted from each of the analyzed sequences: "FRONTAL", "NEON", "RIGHT SIDE" and "LEFT SIDE", respectively

Estimation of the improvement obtained by using the Temporal State Diagram

We run the pupil-search algorithm over the video sequences (shots of these sequences can be seen on Figure IV-4). To consider that the algorithm deduced successfully eye actions on one frame (Table I-Ok), we checked these criteria:

– quantitative: the X and the Y components of both eyes were the same, computed with the following intervals of accuracy:

(IV-2a) $$\left|X_L - X_R\right| \leq 3\%, 5\% \text{ or } 10\% \text{ of } W_{ROI} \text{ and}$$

(IV-2b) $$\left|Y_L - Y_R\right| \leq 3\%, 5\% \text{ or } 10\% \text{ of } H_{ROI};$$

– qualitative: the result obtained defined the expected eye action. This was visually inspected.

The performance evaluation can be seen on Table IV-1 on the rows labeled as "WITHOUT". Appendix IV-G contains the data from the analysis results of the sequence "NEON".

To understand the level of improvement achieved when adding the Temporal State Diagram, we compared the previous results with those obtained after utilizing the diagram (Table IV-1 [A]). The improvement is notorious; above all, the state diagram could determine the S_{close} state that the algorithm utilized independently could not. Lighting conditions influence the image analysis results but the state diagram ensures a success rate of 70%-90% by providing animation data that is as smooth as possible and that generates natural eye motion.

We have also studied the percentage of errors coming from detecting a false S_{close} state (Table IV-1 [B]) and in (Table IV-1 [C]) we count the percentage of correct analyzed close motions (that would rendered to a closed eye) but that were not detected as S_{close}. The most disturbing artifacts come from the introduction of S_{close} where there has not been any such action. We conclude from the analysis that a trade-off between the accuracy of motion and the robustness of the Temporal State Diagram is needed. The greater the interval of accuracy permitted, the less smooth the eye motion will be and the smaller the margin of action that the Temporal State Diagram will have to correct errors (see Figure IV-5 and Figure IV-6, where the graphs plot the smoothness of the results).

Appendix IV-H includes the data obtained after applying the Temporal State Diagram on the analyzed results from sequence "NEON".

Table IV-1

TEST RESULTS ON EYE-STATE ANALYSIS

			SEQUENCES			
			NEON	FRONTAL	LEFT SIDE	RIGHT SIDE
PERCENTAGE	WITHOUT	3%	16.84	17.10	27.77	9.53
		5%	34.95	43.33	38.15	23.26
		10%	73.47	68.15	90.29	86.28
	WITH STATE DIAGRAM	A - 3%	88.78	72.60	79.91	85.35
		A - 5%	95.15	84.78	88.94	78.60
		A - 10%	93.88	86.65	90.97	86.29
		B - 3%	27.27	28.21	62.92	73.02
		B - 5%	0.00	18.46	61.22	33.70
		B - 10%	0.00	21.05	2.50	37.63
		C - 3%	4.68	0.32	0.00	0.00
		C - 5%	9.26	13.54	0.00	0.00
		C - 10%	3.59	17.30	5.65	0.00

A – Correct interpreted results B – Errors derived from false S_{closed} assigned states. C – A correct 'close-eye' motion interpretation is obtained without need of deducting S_{closed} in the Temporal State Diagram
Analysis for accuracy of 3%, 5% and 10% of the ROI's width and height.
Rows WITHOUT, A, C contain percentage over the complete video sequence.
Row B reflects percentages over the amount of total erroneous results

Study of the Diagram filtering effect and the standard eye-motion constraints

After examining the evolution of the results obtained from the pupil search analysis (examples from the "NEON" sequence are on Figure IV-5a and Figure IV-6a) we realized that the variability of the data is such that deriving animation parameters directly from them would lead to incoherent eye motion. This would disturb interpersonal communications.

Quantizing eye-motion (X and Y component) with the same value ensures a common behavior that allows us recreating natural eye motion based on the observed movements. Looking at the graphs from Figure IV-5b and Figure IV-6b, we notice how the accuracy on the similarity chosen (3%, 5% or 10%) plays and important role in the determination of the smoothness of the final data. For instance, looking at frames ranged 26-46 on Figure IV-5b, we see that the quantization at 10% introduces some annoying peaks inherited from the analyzed data on the left eye. These peaks do not appear at 3% and at 5%, after the Temporal State Diagram has been applied, because the Temporal State Diagram is able to choose the best values for these accuracies, where no coherent

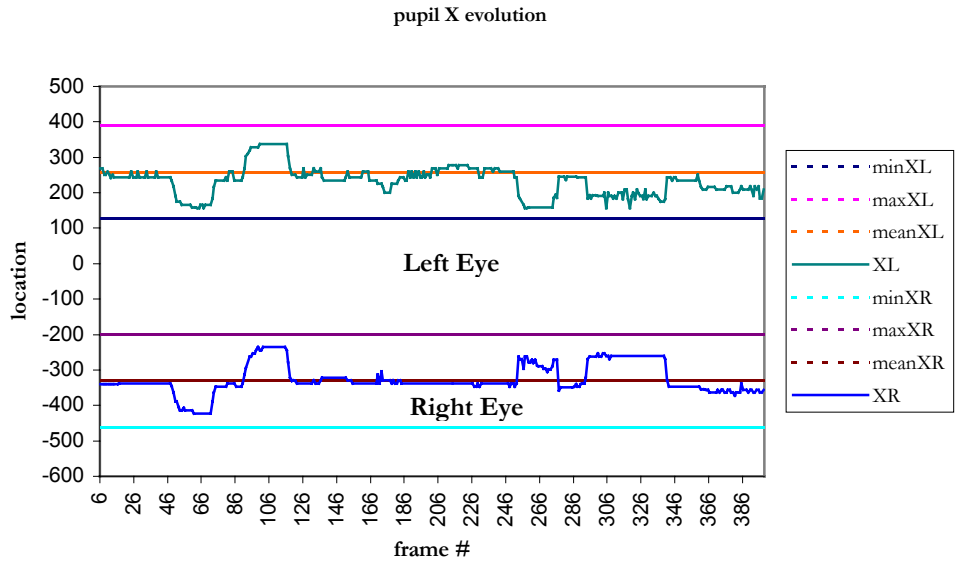
result was possible from the simple analysis because the X components of both eyes were too different.

Another effect that is visible after quantizing and filtering with the Temporal State Diagram is the suppression of some motion information if we do not obtain enough correct analyzed data. This result can be observed on frames ranged 136-160 in Figure IV-6b. The values assigned to the Y component for the 3% accuracy do not match the ones observed for both eyes in Figure IV-6a. This is due to the lack of data taken as correct with 3% accuracy; the algorithm does its best by applying the previous analysis result.

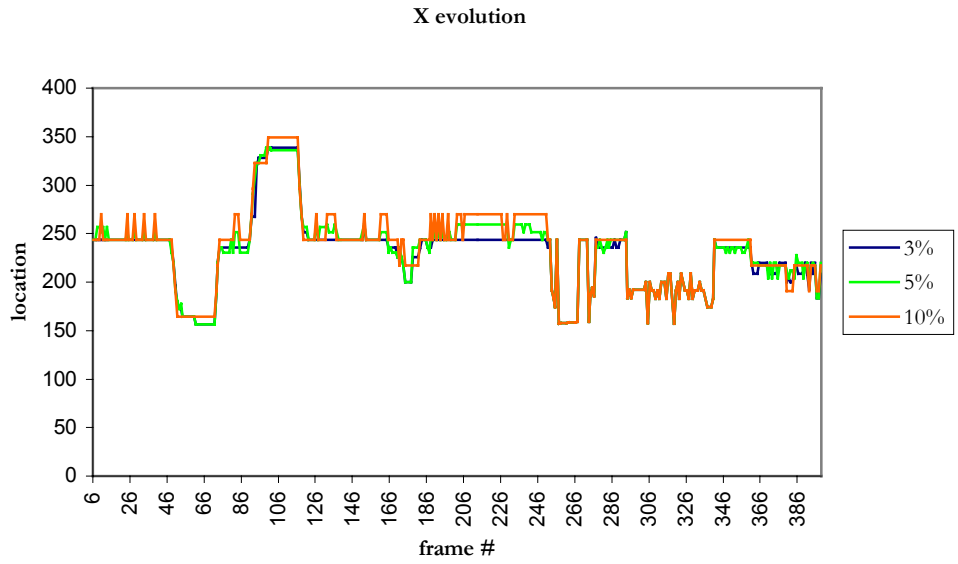
Although the Temporal State Diagram is applied over states defined accounting the x and y components of the pupil location, to understand the effects of the filtering on the eye analysis, we have plotted the X-component behavior separately from the Y-component behavior (Figure IV-5 and Figure IV-6 respectively).

Evaluation of the generated animation

The evaluation of the degree of naturalness generated by the animation of head models with the parameters obtained thanks to our eye analysis technique is presented in Chapter VI. The vector characteristic of the data extracted from the analysis enables the algorithmic extension of our method to analyze faces presenting different poses in front of the camera. The eye expression – pose coupling is also tested in Chapter VI.

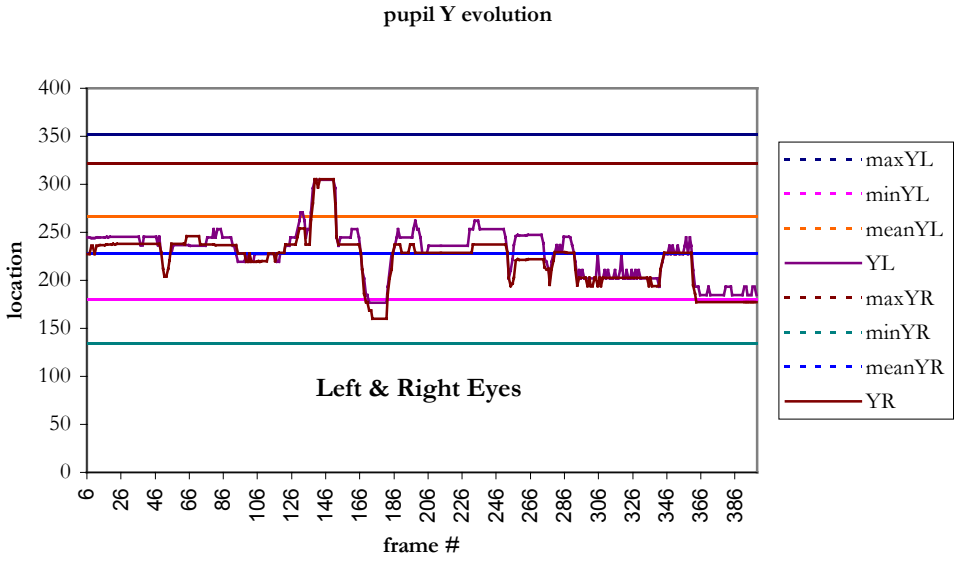


(a)

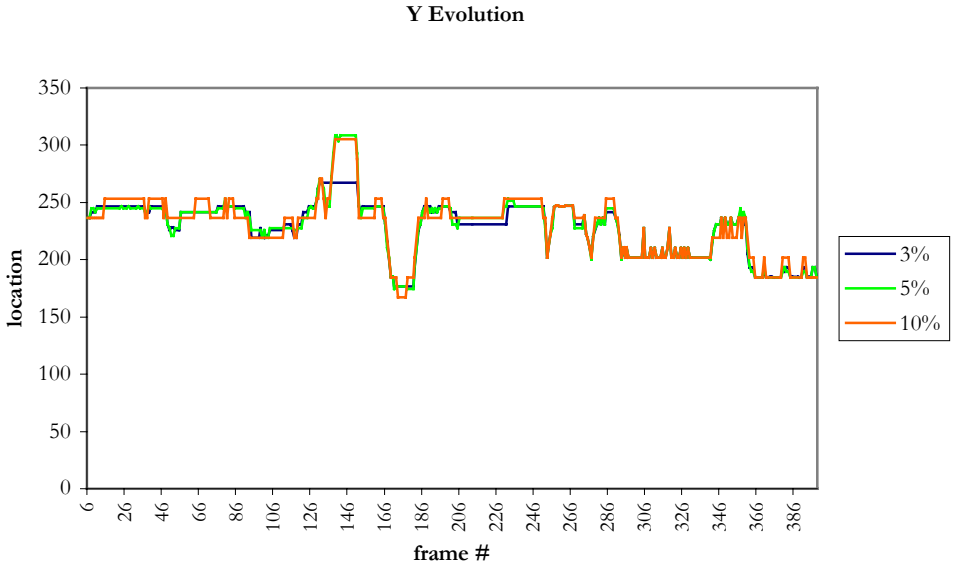


(b)

Figure IV-5. The upper graph depicts the evolution of the extracted data regarding the pupil X location for both eyes. The lower graph represents the resulting X location after applying the Temporal State Diagram. It shows the results for a $f(X,Y,W,H)$ that quantizes with an accuracy of 3%, 5% and 10% of W_{ROI} (example sequence: "NEON")



(a)



(b)

Figure IV-6. The upper graph depicts the evolution of the extracted data regarding the pupil Y location for both eyes. The lower graph represents the resulting X location after applying the Temporal State Diagram. It shows the results for a $f(X,Y,W,H)$ that quantizes with an accuracy of 3%, 5% and 10% of H_{ROI} (example sequence: "NEON")

IV.3 Introducing Color Information for Eye Motion Analysis

Several reasons motivated us to introduce color information as a possible source of reliable data for the analysis of eye motion. First, we wanted to study a new way to analyze eye motion that could allow us to complement the energy search algorithm previously described and to increase the accuracy of the motion analysis always profiting from the use of a Temporal State Diagram. Furthermore, the study would also point out the relevance of using color information to analyze motion, which we have rarely seen in the literature.

Our study has led us to a design of a specific eye-opening algorithm that becomes the source to generate the parameters for Y motion of the eyelid. It also makes possible to simplify the eye state algorithm by assigning the Y motion of the pupil based on the analyzed eyelid motion. We would then apply the dual interpretation of the correlation pupil-eyelid characteristic exploited in the previous approach. If the simplification is not made, then the combination eye-opening detection/gaze estimation is capable of detecting extreme expressions the same way the inter-feature complementary information from the eyebrow does (Section IV.4 contains these details).

IV.3.1 Eye opening detection

Color distribution analysis on the eye area shows that the eye can be clearly classified as different from the skin in terms of its hue and saturation components. We define the degree of eye opening as proportional to the inverse of the amount of skin contained within the analyzed ROI.

To measure the quantity of skin on the eye feature that we have extracted, we count the number of pixels we classify as skin pixels. The classification is made based on the probability of the pixel belonging to the skin. Every frame will be analyzed obtaining the opening as:

(IV-3) $$EyeOpening \propto \frac{1}{probSKIN}$$

where

(IV-4) $$probSKIN \propto \sum_{h}\sum_{s} NUMpel_{h,s} \cdot PDF_{SKIN}(H = h, S = s).$$

Since features extracted from different video sequences may have different size, $NUMpel_{h,s}$ is the total amount of pixels of determined hue and saturation normalized by the total number of analyzed pixels. PDF_{SKIN} is the Probability Density Function of the skin HS characteristics. The PDF is obtained by analyzing the pixel HS distribution of different skin images. Instead of using a general database for non-specific skin detection (Sahbi & Boujemaa, 2000), we use speaker-dependent data. In our approach, we approximate $probSKIN \approx probClosedEyes$ and we obtain the PDF from a sequence of frames of the closed eyes of the individual to be analyzed. We perform some training with sequences of closed eyes of the individual to be analyzed. The chrominance data of video images, hue and saturation, are highly dependent on the characteristics of the acquisition system utilized: camera, image card, etc., and on the person's nature (eye and skin color). They are less dependent on the recording environmental conditions, such as the lighting on the face. Our solution needs the described training step to ensure that the study we have performed reflects the usefulness of color information in the present context.

IV.3.2 Gaze detection simplification

Using the pupil-eyelid correlation hypothesis. We can restrict the analysis to study the horizontal movement of the eyes. Alternatively, we do not perform an exhaustive scan in a square zone but a horizontal sweep with a vertical rectangle of area $\alpha \cdot Weye_i \times Hfeature$. Equation (IV-1) is then transformed to only look for coordinate X, which indicates if the eye looks right or left:

(IV-5)

$$X = \min\left[\sum_{l=1}^{\alpha \cdot Weye\ Hfeature} \sum_{m=1} I^2(x + l - Weye/2, m)\right]$$

$$\forall x, y \ \exists \ \ x + l - Weye/2 > 0$$

IV.3.3 Analysis interpretation for parametric description

To be able to synthesize eye movements, we have parameterized the analysis data obtained so it can be interpreted. We set a tight cooperation between the two previously described analysis techniques in the Temporal State Diagram (Figure IV-8), that allows us to double-check possible erroneous results from the algorithms. Next sub-sections develop the complete process for the state diagram specification.

Parameterization of eye movements

We define parameters to describe the eye movement to be synthesized. These parameters are simple action units that mark how actions should be synthesized.

We have defined two parameters according to the two analysis techniques we use, eye-opening (EO) and horizontal pupil orientation (HPO). Each parameter takes different values depending on the action to perform during the synthesis. To test our procedure and to be able to evaluate its viability in real time, EO and HPO are quantified using 9 states that allow us to describe eye action with a minimum richness (see Figure IV-2 for the location distribution assigned to the 9 states). Table IV-2 depicts the actions and the corresponding values. As the table shows, no saccadic motion is perceptible by such a action description.

Table IV-2

ACTION UNIT DESCRIPTION.

	open	closed		left	center	right
EO	1	0	HPO	-1	0	1

Quantifying the results to parameterize them

The analysis algorithms described in the previous sections generate results that have to be paired to the proper parameter value.

Computing the *EyeOpening* along a sequence generates a function defining two levels. The function adopts the highest values when the eye is open (EO=1) and the lowest ones when the eye is closed (EO=0) (Figure IV-9). From sequence to sequence this difference in levels is fairly stable but the levels may be situated at different values. The values of the levels depend on the video camera and the lighting conditions. Since we analyze the sequence in a frame-by-frame basis and we do not count on a priori results, we define EO in relative terms. To do so, we compare the *EyeOpening* value of current frame i with the average *EyeOpening* values of the previous k frames (*avg*). If the difference, Δi_avg, is greater than a certain threshold (Tb_{EO}) the eye has opened, if it is smaller the eye has closed, otherwise it remains as in the previous frame.

Table IV-3

THE 36 COMBINATORY RESULTS FROM THE EYE ANALYSIS.

S_L		S_R			S_L		S_R			S_L		S_R		
EO	HPO	EO	HPO	(*)	EO	HPO	EO	HPO	(*)	EO	HPO	EO	HPO	(*)
0	-1	0	-1	**A**	0	1	0	-1	$\mathbf{S_1}$	1	0	0	-1	**A**
0	-1	0	0	**-**	0	1	0	0	**A**	1	0	0	0	**A**
0	-1	0	1	**-**	0	1	0	1	**A**	1	0	0	1	**A**
0	-1	1	-1	**A**	0	1	1	-1	**A**	1	0	1	-1	**A**
0	-1	1	0	**A**	0	1	1	0	**A**	1	0	1	0	$\mathbf{S_3}$
0	-1	1	1	**A**	0	1	1	1	**A**	1	0	1	1	**A**
0	0	0	-1	**A**	1	-1	0	-1	**A**	1	1	0	-1	**A**
0	0	0	0	**-**	1	-1	0	0	**A**	1	1	0	0	**A**
0	0	0	1	**-**	1	-1	0	1	**A**	1	1	0	1	**A**
0	0	1	-1	**A**	1	-1	1	-1	$\mathbf{S_2}$	1	1	1	-1	**A**
0	0	1	0	**A**	1	-1	1	0	**A**	1	1	1	0	**A**
0	0	1	1	**A**	1	-1	1	1	**A**	1	1	1	1	$\mathbf{S_4}$

$\boldsymbol{S_L S_R}$ = left&right analysis results; (*) = evaluation of the results; **(-)** = both results are erroneous; **A** = at lest one of them is correct & $S_L \neq S_R$; $\mathbf{S_i}$ = defined state ($\mathbf{S_1}$=S_{closed}, $\mathbf{S_2}$=look left, $\mathbf{S_3}$= look center, $\mathbf{S_4}$ = look right)

The parameter *X* that we obtain from the eyesight detection algorithm defines the horizontal location of the pupil in the feature. Finding its relative location regarding the eye on the feature image determines if the eye looks left, center or right (HPO=-1,0,1). Figure IV-7 shows on how we quantify the *X* value.

The more precise the synthesis is wanted, the more intermediate values both action units, EO and HPO, should take. In such cases, the quantization of the space of analysis results would change by adding more levels. The number of quantization levels must be chosen based on the capacity of synthesizing those details by the clone animator and the image quality and size of the feature. We must also evaluate if increasing the complexity of the quantization is appreciated when watching the real-time synthesis. This can be evaluated through an on-line analysis (see Chapter VI for details).

This first parameterization provides the preliminary analysis data, which might be erroneous. We estimate if our results are correct and decide which is the most suitable action by combining the information obtained from both eyes and both analysis algorithms in the Temporal State Diagram.

Applying the Temporal State Diagram

Table IV-3 shows all the possible combinations of analysis results, $S_L S_R$. They can be completely erroneous for both eyes *(-)*, different for each eye, in which maybe one is

wrong (A), or exactly the same for left and right (S_i). Applying the constraint of having the same behavior in the left and the right eye we have developed our diagram of states, see Figure IV-8.

The diagram cross-checks the behavior in both eyes and estimates the best current eye action (S_t^t) depending on the analysis result (A, *(-)*, S_i) and the previous eye state (S_t^{t-1}).

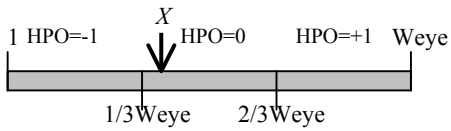


Figure IV-7. Quantization of HPO looking for left eye

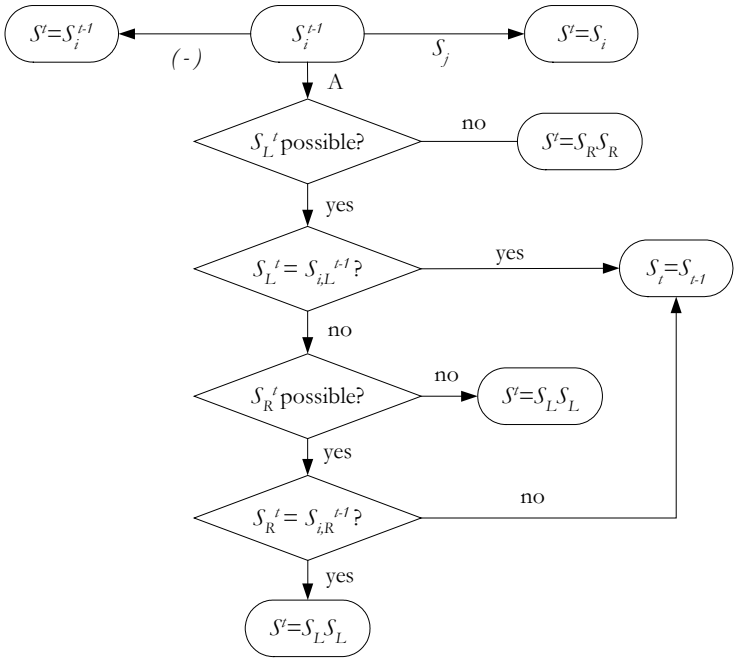


Figure IV-8. Temporal State Diagram for the eye action tracking with simplified gaze analysis. $S_{i\,(R/L)}^t$ represents a determined state *i* at time *t* for either the right of left eye and S^t the final result. Check Table IV-3 for the state combinations

IV.3.4 Experimental evaluation and conclusions

We have used two sets of images for our experiments; the experimental setting has being the same as in the previous section. To obtain the PDF of HS values, one set has the recorded closed eyes of the person. The other set contains the eyes of the recorded face that is analyzed. Both sets were obtained under uncontrolled lighting conditions, and to reduce the noise introduced by the camera we first filtered the analysis regions with low pass filters (a combination of 3x3 size median and average filters).

The PDF used for the final analysis is an average of the PDFs obtained from different sequences. To reduce the influence of noise on the results, we average the analysis area from N frames of each sequence and then we obtain its PDF.

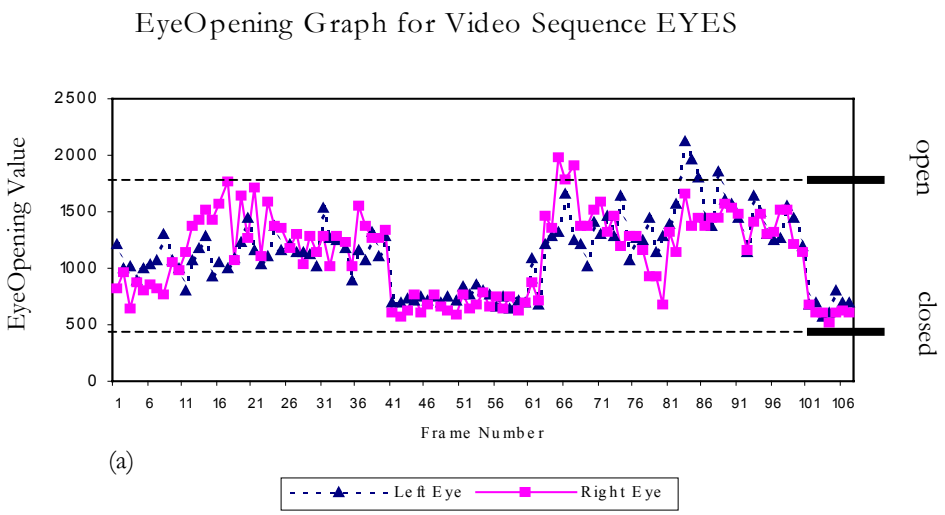
After testing our algorithms, the results were satisfactiory (Figure IV-9). In around 85% of the studied cases the *EyeOpening* algorithm could clearly provide the two expected levels for the open-close movement. The number of previous results accounted for state determination depends on the frame rate of the sequence. For 15 f/s we have used the previous three results. The *GazeDetection* algorithm is performing better, leading to positive results in around 98% of the tests. Applying the state diagram is convenient above all on those transitions areas where the *EyeOpening* algorithm changes from open to close, or vice versa.

Regarding the speed of performance, the heaviest computational part lies on the filtering and the component conversion from RGB to HSI of the video input. The importance of the filtering strongly depends on the graphics card output quality. Regarding the conversion, we could consider positive to adapt these algorithms to the YUV components (S:U, H:V, I:Y) since many graphics cards provide YUV output directly.

Since the speaker-dependent parameters – Th_{EO} and its skin PDF – need to be trained, we conclude that color considerations are useful if the skin characteristics of the person are available and geometrical constraints do not interfere. The probabilistic nature of the analysis does not allow us to extend the *EyeOpening* algorithm to analyze any other pose different from a near-to-frontal head orientation. The *EyeOpening* results represent a scalar value with no geometrical meaning and cannot be interpreted in 3D, thus they are not susceptible to be extended to be used in other poses. The *GazeDetection* values are of vector nature thus interpreted in 3D and adaptable to understand motion in a different pose.

No test has been made using a generic PDF. We believe that results should not change for all those individuals whose chromatic characteristics do not differ much from those represented in the generic PDF. We expect a decrease in performance on people

whose chromatic skin characteristics are extreme, above all if no specific quality can be guaranteed.



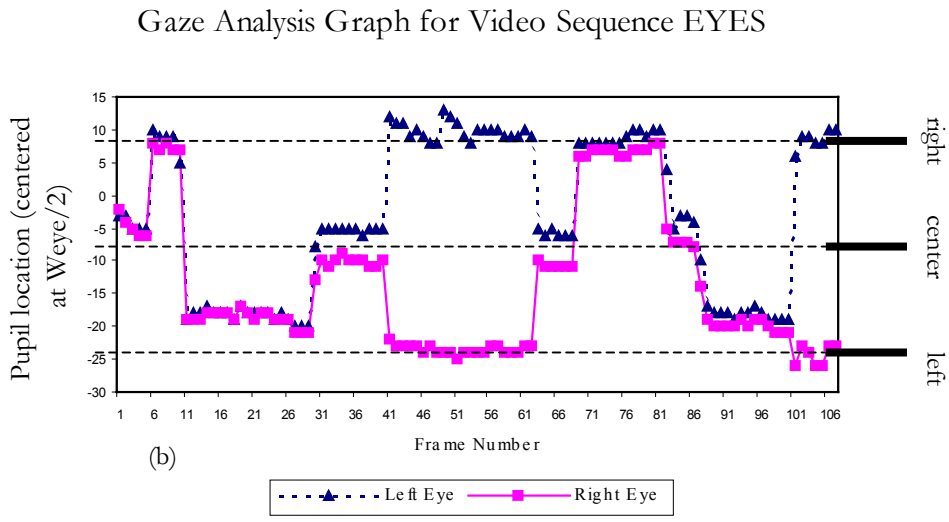


Figure IV-9. Analysis Graphs for a tested sequence. (a) *EyeOpening* (two quantization levels). (b) *GazeDetection* (three quantization levels). The upper row shows some frames from the analyzed video

IV.4 Eyebrow Motion Analysis Algorithm

Historically, eyebrow motion analysis has been less investigated than other feature analysis techniques (eyes and mouth). In the literature we find that the first works trying to analyze eyebrow behavior (Huang, C.-L. & Huang, Y.-M, 1997) are concerned by the search of expression information. More recently, Goto, Kshirsagar and Magnenat-Thalmann (2001) have also presented a method to analyze eyebrow motion to extract facial animation parameters. The analysis methodology followed is rather heuristic and when presenting the proposed approaches the influence of the environmental conditions is very often not discussed. Kampmann (2002) proposes a technique that is able to detect the eyebrows even if they are partially covered by hair. In general, we have not found any motion analysis technique that formally relates the image analysis processing results to the generation of the motion parameters.

In this section we describe an eyebrow motion analysis technique where the image processing involved tries to adapt its analysis to the characteristics of the user and the environmental conditions. We relate the results of this image analysis directly to a motion template that models eyebrow motion.

To study eyebrow behavior from video sequences we utilize a new image analysis technique based on an anatomical-mathematical motion model. This technique conceives the eyebrow as a single curved object (arch) that is subject to the deformation due to muscular interactions. The action model defines the simplified 2D (vertical and horizontal) displacements of the arch. Our video analysis algorithm recovers the needed data from the arch representation to deduce the parameters that deformed the proposed model.

Table IV-4

NOTATION CONVENTIONS USED IN THIS SECTION'S FORMULAE

x_n, y_n : real coordinate values of the eyebrow in their neutral position.
$x_n[\ i\ ]$, $y_n[\ i\ ]$: coordinate value of the pixel obtained from the video image analysis of the eyebrow in its neutral position at position i.
x, y : real coordinate values of the eyebrow in their current (analyzed frame) position.
$x[\ i\ ]$, $y[\ i\ ]$: coordinate value of the pixel obtained from the video image analysis of the current frame eyebrow at position i.
Δx, Δy : real coordinate difference between the current eyebrow arch and the neutral arch $x_{frame}-x_{neutral,}$ $y_{frame}-y_{neutral,}$ respectively.
$\Delta x[\ i\ ]$, $\Delta y[\ i\ ]$: real coordinate difference between the pixels from the current eyebrow arch being analyzed and those from the neutral arch at position i.
[0], [N] and [max] indicate the computed values for the first point (x = 0), the last point (x = last) and the point with the maximum vertical value ($x : y$ = max) on the arch.

IV.4.1 Anatomical-mathematical eyebrow movement modeling

To model the eyebrow movement, we define some mathematical expressions that superficially follow the muscular behavior and interaction when eyebrow motion exists.

Basically, four muscles control the eyebrow movement:

(i) Frontalis (F): that elevates them.

(ii) Corrugator (CS): that pulls them downwardly, produces vertical glabellar wrinkles.

(iii) Procerus: that lowers the eyebrows downwardly.

(iv) Orbicucularis Oculi (OO): that closes eyes and lowers eyebrows.

Figure IV-10. Several muscles generate the eyebrow movements. Upward motion is mainly due to the Frontalis muscle and downward motion is due to the Corrugator, the Procerus and the Orbicularis Oculi muscles[1]

Although the shape of an eyebrow is dependent on the physical appearance of the person, its motion is related to more general muscular actions. This enables us to represent eyebrows as arches, whose shape is specific to the person but whose motion can be mathematically modeled. We parameterize the arch movement as the displacement in the x, y and z-axis of each of its points compared to the initial neutral position, when no force is acting on it: $\Delta x = x_{frame} - x_{neutral}$, $\Delta y = y_{frame} - y_{neutral}$ and $\Delta z = z_{frame} - z_{neutral}$.

Two different behaviors exist in eyebrow motion, one when the expression goes upwards and another when the expression goes downwards. Different muscular action is involved in each of them and therefore different models control them. These

[1] Images and information based on data from www.oculoplastic.co.uk/eyebrows/anatomy.html

expressions have been derived from the observation of the muscular motion of the eyebrows and the empirical study of the optical flow behavior of the eyebrow area observed on real video sequences and adapting the parameters involved to the anatomical shape of the eyebrow.

Eyebrow Motion Expressions:

Upwards:

(IV-6) $$\Delta x = Ff_x \cdot e^{(-x_n/\alpha)}$$

(IV-7) $$\Delta y = Ff_y + Ff_y' \cdot e^{(-x_n/\alpha)}$$

Downwards:

(IV-8) $$\Delta y = -Fcs_y - Foo_y \cdot (|x_n - \beta| - \beta)^2$$

If $x_n < \beta$:

(IV-9a) $$\Delta x = -Fcs_x$$

If $x_n > \beta$:

(IV-4b) $$\Delta x = Foo_x \cdot (\beta - x_n) - Fcs_x$$

Ff, *Ff'*, *Fcs* and *Foo* are the magnitudes associated to the force of the Frontalis muscle, Corrugator muscle and the Orbicucularis Oculi respectively. The action of the Procerus muscle, being close and highly correlated to the one from the Corrugator, is included in the *Fcs* term. $_x$ and $_y$ indicate the different components, $\alpha = 2 \cdot \frac{w}{3}$ and $\beta = \frac{w}{2}$. All coordinates relate to the eyebrow local coordinate system. Figure IV-11 depicts the coordinate axis for the left eyebrow; the right eyebrow is symmetrical over an imaginary vertical line located between eyebrows.

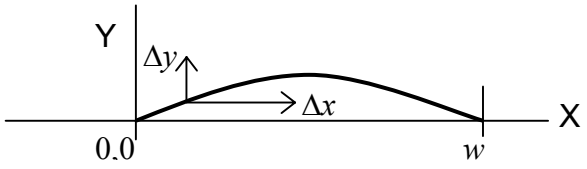


Figure IV-11. Eyebrow model arch for the left eye and its coordinate reference. The origin for the analysis algorithm is always situated at the inner extreme of the eyebrow (close to the nose) and defined for the eyebrow in its neutral state

$\Delta z = f(\Delta x, \Delta y)$ is difficult to model out of the frontal view of an eyebrow and does not provide critical information regarding the expression. Δz cannot be well estimated from a frontal orientation. If we want to synthesize realistically 3D eyebrow motion with information obtained from image analysis under these conditions, we may estimate the Δz movement by assuming that eyebrow motion follows the forehead surface, thus, simulating its natural behavior. The "displacement of texture coordinate" synthetic animation technique described in (Valente & Dugelay, 2000) illustrates this concept. This procedure simulates the eyebrow skin sliding motion on the skull. Changing the texture coordinates vertically and horizontally generates the animation; the model 3D coordinates remain unchanged, thus leaving the skull shape untouched.

Applying the formulae over two rounded arches with different parameter values, we obtain deformations that correspond to the expected eyebrow deformation due to those forces. Figure IV-12 shows the simulation of extreme modeled movement (downwards and upwards) on both eyebrows.

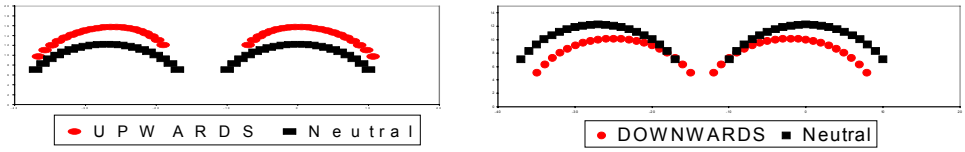


Figure IV-12. The action of the eyebrow behavior model applied over the neutral arch results on a smooth deformation. The graph on the left depicts eyebrow rising motion (*upwards*) for positive values of Ff_x, Ff_y and Ff_y'. The graph on the right represents eyebrow frowning (*downwards*) for positive values of Fcs_x, Foo_x, Fcs_y and Foo_y

IV.4.2 Image analysis algorithm: deducing model parameters

We achieve two goals by modeling the eyebrow movement. On the one hand, we simplify the eyebrow motion understanding to a point where we can derive its movements on images by comparing image data with the model parameters. On the other hand, this model is complete enough to generate the required information to create synthetic eyebrow expressions.

The developed image analysis algorithm tries to reduce the image of the eyebrow down to the proposed model in order to study the distribution of the points on the eyebrow arch. Then, it deduces the strength of the parameters involved. The analysis compares data extracted from the current video frame against the data obtained from the frame where the eyebrow is in a neutral position or *neutral frame*.

Algorithm procedure

Binarization:

Normally, eyebrows and skin are easy to separate in terms of hue, and with less accuracy, intensity. Under *regular* although *uncontrolled* lighting conditions we can differentiate eyebrows from skin and therefore binarize the feature image. We consider uncontrolled lighting, any illumination over the face that permits the eyebrow visual differentiation on the video frame.

Due to the anatomical nature of the head, eyebrows do not present the same aspect all over their arch. The area situated on the inner part of the *Superciliary* arch is generally better defined and easier to differentiate from the skin than the eyebrow arch that goes towards the joint with the *Superior Temporal Line*, because this last one is usually more sparse. Our analysis algorithm needs to detect the complete eyebrow because we are interested in studying the overall shape behavior. We have developed a binarization algorithm that analyzes the eyebrow in two different zones. One zone includes from the center of the face up to the point which is half way between the *Foramen* and the *Zygomatic* process (point where the eyebrow usually changes shape direction and texture) and the other zone goes from there to the external end of the eyebrow. We refer to Figure IV-13 to locate the different parts.

To perform the binarization we apply two different thresholds, one per zone. Each threshold is obtained by analyzing the histogram distribution of the corresponding area. The eyebrow is taken as the darkest part on the video image being analyzed.

(IV-10)
$$Th_i = \min_i + \frac{\max_i - \min_i}{3}$$

If pixel_value<Th_i, the pixel is considered as part of the eyebrow. The threshold has been chosen to be at a third of the intensity distribution because the analysis area covers three major intensity zones, which are well differentiated in most lighting conditions: the eye zone under the eyebrow, the eyebrow itself, and the forehead zone over the eyebrow. Usually, the darkest one belongs to the eyebrow. Figure IV-14a shows the histogram of one of the ROIs. We have marked the three different zones on the histogram. The reader must notice that the current binarization procedure is suitable for normal to dark haired people. The analysis of blond person should undergo the study of more specific criteria to binarize the image.

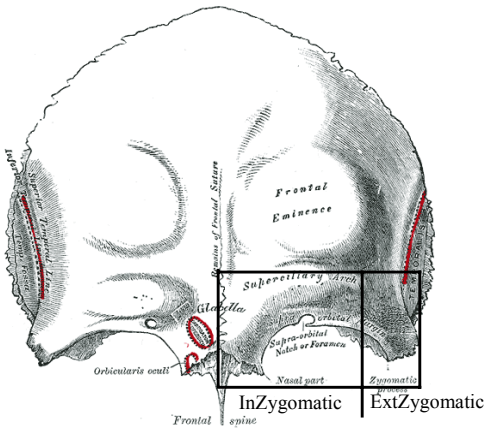


Figure IV-13. The eyebrow changes its hair density as it goes away from the inner extreme. The bone structure of the skull determines the shading difference along the eyebrow. We set two different binarization thresholds: Th_1 for the *InZygomatic* zone and Th_2 for the *ExtZygomatic*

Results of the binarization process:

This algorithm always detects the eyebrow even if, in some cases, it also introduces some artifacts. Eyes and hair are often labeled as being part of the eyebrow (see Figure IV-14b, where eye is marked as eyebrow). Both, eye and hair must not be taken into account when analyzing the ROI to extract the eyebrow arch. Due to predefined and fixed situation of the artifacts on the ROI, we have easily adapted the thinning algorithm to avoid taking them as part of the eyebrows.

(a) (b) (c)

Figure IV-14. The eyebrow 'two part' binarization leads to good determination of the eyebrow area but it also may introduce artifacts by labeling eyes or hair as part of the eyebrow. In the current eyebrow binary image we see how the eye has also been detected

Thinning:

We perform a vertical thinning over the binarized image to obtain the rounded arch that will define the eyebrow. The anatomical structure of the ocular cavity creates very dark shadows under extreme lighting conditions; therefore, sometimes the

binarization process can only make a rough estimation of the overall eyebrow shape. Under unknown conditions the eyebrow arch is robustly obtained by detecting the gradient change between forehead and eyebrows on the binarized feature image. This gradient change remains stable under most lighting conditions (see Figure IV-14c).

Parameter deduction:

The parameters that model eyebrow behavior are deduced from comparing the thinned arch at the current frame against the arch obtained from the analysis of the eyebrow in its neutral position (i.e. showing no action). The process starts by deducting the general eye behavior because our model formulates upward or downward motion with different expressions. The median vertical value of the arch (median of the y-component of the points shaping the arch) is compared against the median vertical value of the neutral arch. If the current median is greater than the neutral one, we conclude that we are analyzing upward expressions; otherwise, the downward model representation is used. After selecting the model, the most significant data from the arch are extracted and used to obtain the model parameters.

Parameter expressions:

- *UPWARDS:*

(IV-11a) $$Ff_x \approx \Delta x[0] \cdot e^{(x_n[0]/\alpha)} \approx \Delta x[0]$$

(IV-11b) $$Ff_y' \approx \frac{\Delta y[N] - \Delta y[0]}{(e^{(-x_n[N]/\alpha)} - 1)}$$

(IV-11c) $$Ff_y \approx \Delta y[0] - Ff_y'$$

- *DOWNWARDS:*

(IV-12a) $$Fcs_x \approx -\Delta x[0]$$

(IV-12b) $$Foo_x \approx \frac{\Delta x[N] + Fcs_x}{\beta - x_n[N]}$$

(IV-12c) $$Fcs_y \approx \frac{-\Delta y[0] - \Delta y[N]}{2}$$

(IV-12d) $$Foo_y \approx \frac{-Fcs_y - \Delta y[\max]}{\beta^2}$$

IV.4.3 Experimental evaluation and conclusions

To our knowledge, it does not exist a database of face images that completely suits the needs of our tests. Nevertheless, we have tried to test our procedure over more than one speaker; and specifically, we show the results from the analysis of three individuals of different eyebrow characteristics recorded under uncontrolled lighting conditions.

To test the correct behavior of the model and its application for eyebrow motion analysis, we applied the binarization-thinning technique over the left eye on the frames of several video sequences. Then, we deduced the model parameters contrasting the *frame arch* against the *neutral position arch*. To verify that the obtained parameters actually correspond to the eyebrow behavior, we have plotted the thinning results of each frame together with the obtained arch from applying the model over the *neutral position arch*, this resulting arch is the *modeled arch*. Figure IV-17 shows this arch comparison for the frames presented in the sequence of Figure IV-15; they also include the neutral position arch.

The best way to evaluate the performance of our techniques is to visually compare arch results; unfortunately, this procedure is not suitable to be applied over a large amount of data. To interpret the performance correctness of our approach, we have defined two different measurements:

(i) a *pseudo-area*:
$\tilde{a} = \sum_i \left| y_k[i] - y_m[i] \right|$, which can be understood as the area contained in-between arch k and m and it denotes the shape similarity between them; the closer $\tilde{a}$ is to 0 the more alike they are. We compare the area difference between the *neutral position arch* and the current *frame arch* against the shape difference between the *modeled arch* and the *current frame arch*. This measurement checks if the eye shape modeled by the extracted motion parameters follows the expected eyebrow obtained from the action; and

(ii) a mean difference comparison, where we compare the mean (average vertical value of the arch) difference between the current *frame arch* and the *neutral position arch* against the mean difference between the current *frame arch* and the *modeled arch*. This information helps us to evaluate why the analysis procedure was not able to detect the right eyebrow general action: up/down and it also provides information on the shape behavior of the modeled arch, by studying the sign of the measurement that gives the estimation of its vertical position.

We consider that the algorithm has worked correctly if the pseudo-area comparison shows that the *modeled arch* is closer to the current frame arch than the *neutral position arch*.

Test conclusions

Results show that this analysis technique positively deduces the eyebrow behavior. We are able to analyze video images and extract the few needed parameters to understand and to later synthesize eyebrow motion.

From the visual inspection of our results we conclude that errors come more from the image processing performance of the analysis than from the motion model used. Correct binarization and later thinning are critical to obtain accurate motion parameters. Figure IV-18 plots the measurement results of three different tests. The percentage of estimation success (better measurements over the modeled arch) is around 85% for those sequences where image quality and environment lighting conditions are standard. For low quality video input performance drops to around 50%. We must point out that the worst estimation usually happens for low expression movements, where the inaccuracy of the situation of the analysis area (the speaker may slightly move) is large enough to mislead the average results. In this case, like the *average difference* measurement shows, we may interpret an *up* movement as being *down* or vice versa.

Looking at Figure IV-16b we realize how important the correct and precise definition of the eyebrow analysis area is. The graph plots the results of one analyzed sequence along with the neutral analyzed frame of another sequence where head location and size were not exactly the same. Motion not due to the eyebrow expression but to the overall head pose leads to mistaken results. Our tests have been performed accepting that the head pose on the video sequence is known and frontal. The vector nature of the analysis results makes possible to adapt the method presented to extend its use to any head pose. Then, it fits into the analysis approach described in Section III.4 that is afterwards theoretically developed in Chapter V. Using the proposed extension technique also permits to track accurately the ROI on the video image thus minimizing the influence of the head pose on the analysis.

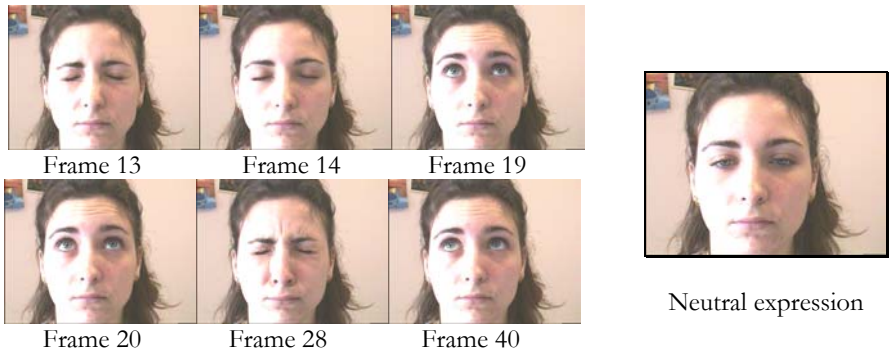
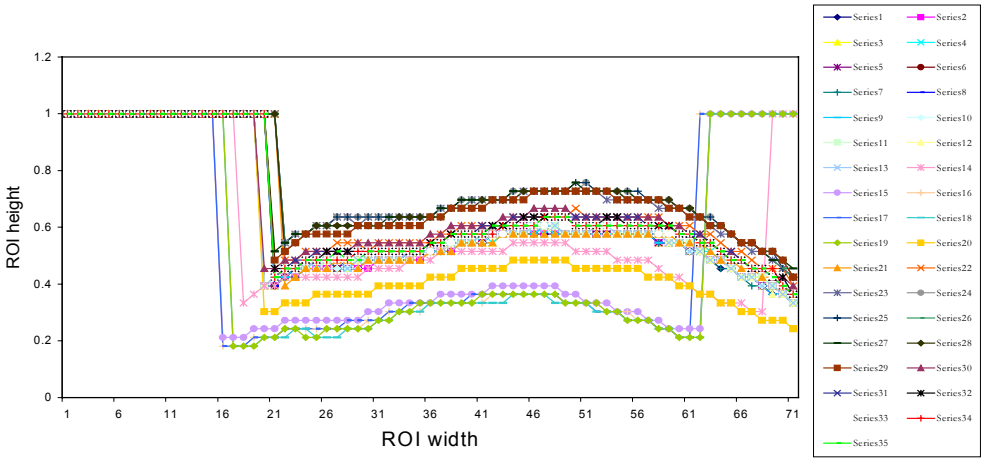
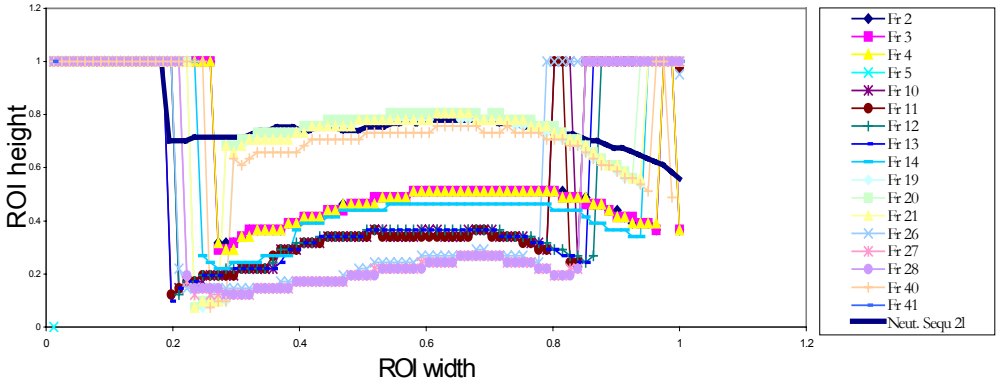


Figure IV-15. Our tests were performed over video sequences where the lighting over the face was not uniform. No environmental conditions were known besides the exact location of the ROI including the eyebrow feature, which remained unchanged through the sequence. Here we present the frames analyzed to obtained the results presented in Figure IV-17



(a)



(b)

Figure IV-16. Correct binarization and thinning clearly gives the data from which to extract the model parameters. Graph (b) plots the mixed results from the analysis of two different video sequences. Neut. Seq.2 is the analysis of a frame where the eyebrow was relaxed taken from a sequence different from the Fr sequence. This comparison simulates what would happen if the pose of the speaker changed during the analysis. The pose motion would cause the movement of the eyebrow but the algorithm would interpret it as a local eyebrow expression (being *upwards* when in reality it is neutral). We must control the pose of the user to completely exploit the algorithm in practical applications

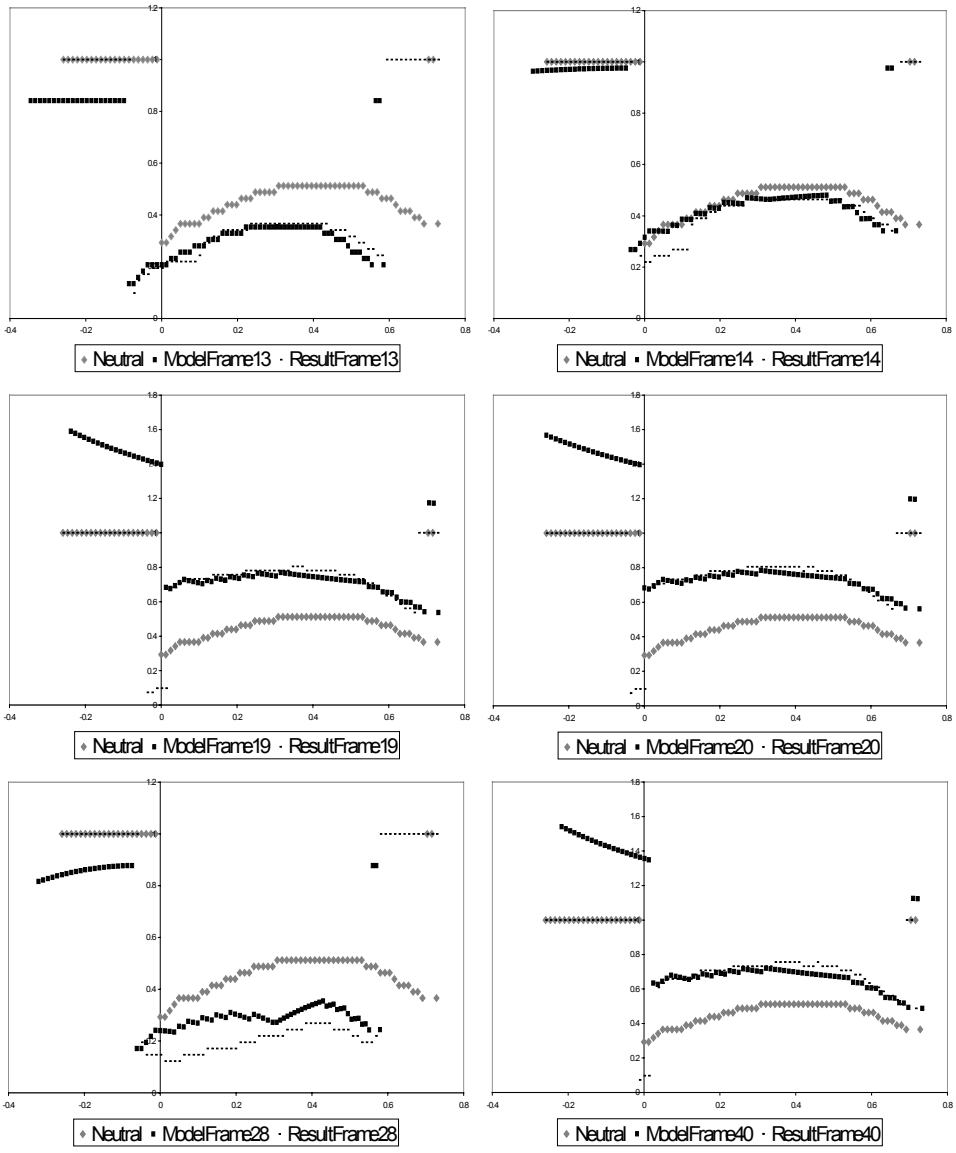


Figure IV-17. We have compared the arch extracted from the analysis (ResultFrame) with the arch resulting from applying the motion parameters on to the Neutral arch (ModelFrame). If the motion estimation is correct both should fall together. The anatomic-mathematical motion model nicely represents the eyebrow deformation. We see on frame 28 how the strange thinning result obtained at the beginning of the arch, probably due to the eyebrow-eye blending during binarization, worsens the algorithm accuracy. Although the obtained parameters still correctly interpret the general downward movement, showing fair robustness, they are no longer able to express the exact motion intensity

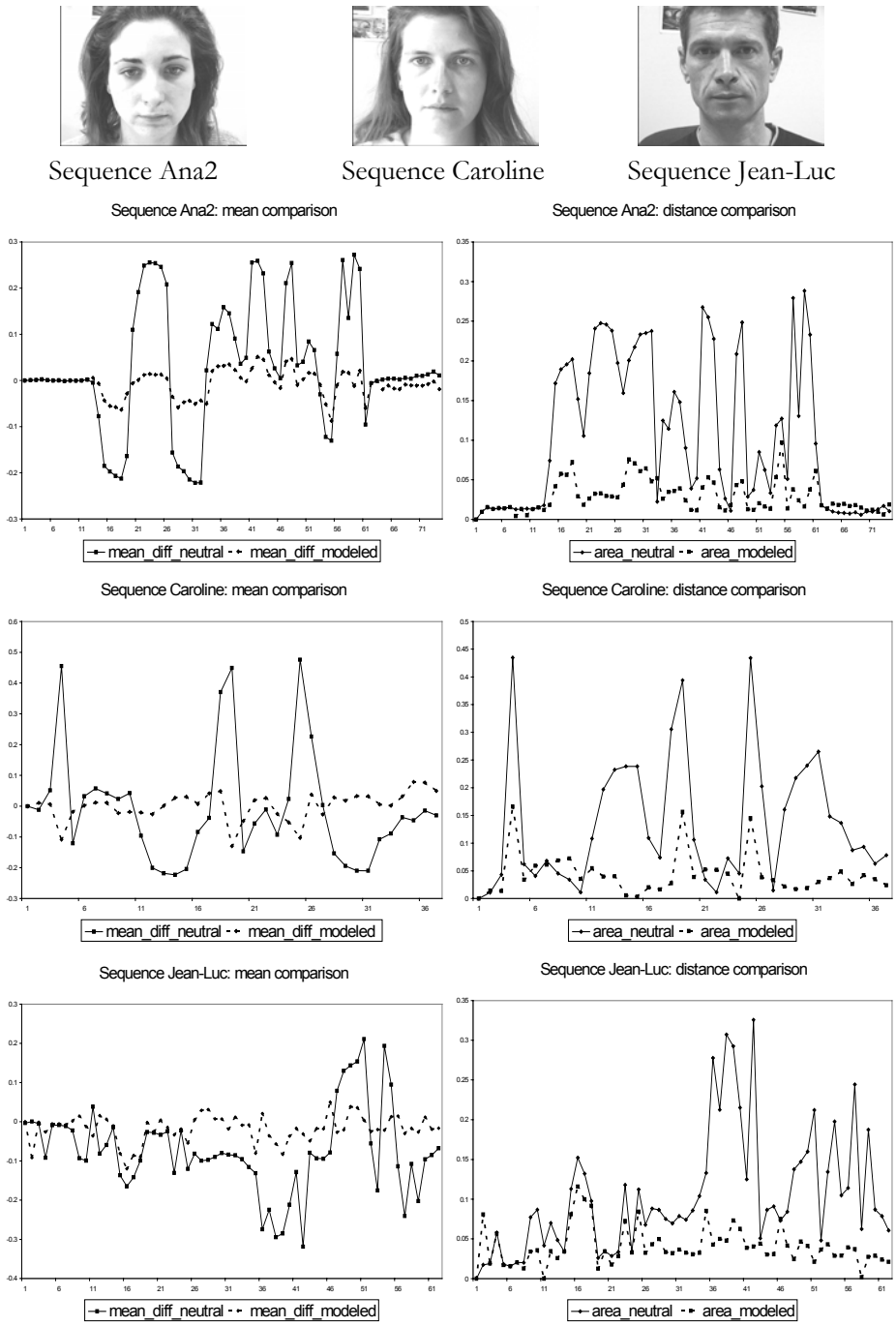


Figure IV-18. These plotted results from three different sequences: Ana2, Caroline and Jean-Luc illustrate the analysis behavior of the algorithm under different conditions. The algorithm proves to detect the right movement (the mean difference decreases) and to estimate the motion parameters correctly (the area decreases). We observe the best behavior for extreme eyebrow expressions. Ana2 sequence success rate: 90.71%, Caroline sequence success rate: 78.38% and Jean-Luc sequence success rate: 82.26%

IV.5 Eye-Eyebrow Spatial Correlation: Studying Extreme Expressions

Generally, the more complex motion models are, the less robust to unexpected environmental conditions their analysis becomes. Our analysis algorithms prove to perform robustly thanks to their simplicity. This simplicity limits the individual motion understanding to natural, coherent eye and eyebrow movements; these constraints are suitable in human-to-human communications but it may undesirably filter those details that add strength to the expression, above all, in the presence of extreme emotions (joy, anger, etc.).

To partially compensate this limitation, we also propose to exploit the existing eye-eyebrow motion correlation to enrich the overall ocular expression understanding from the individual analysis of each feature. When the eyes are closed the eyelids may behave in two different ways, they may be closed without any tension if the eyebrows are neutral or pulled up; or they may be tensely closed if the eyebrows are pushed down. When the eyes are open, the level of the eyebrow height indicates the degree of opening of the eyelid. Figure IV-20 illustrates this clear eyelid-eyebrow correlation. Extreme eyebrow actions determine and refine eye motion by:

(i) extending the information inside the eye Temporal State Diagram to include the interfeature constraints derived from eyebrow analysis. For instance, having a strong downwards eyebrow action will undoubtedly result in a close-eye action, even if the eye data is not reliable (Figure IV-19),

(ii) deriving the final eyelid synthetic behavior from adding to the position obtained from the pupil location an extra term accounting for the strength of the eyebrow movement:

(IV-13)
$$y_{eyelid}^{new} = y_{eyelid}^{former} + \mu \cdot fap + \eta$$

with

$$\mu = \frac{y_{eyelid}\big|_{MAX} - y_{eyelid}^{former}}{fap\big|_{MAX} - fap\big|_0} \text{ and } \eta = \frac{y_{eyelid}^{former} - y_{eyelid}\big|_{MAX}}{fap\big|_{MAX} - fap\big|_0} \cdot fap\big|_0 .$$

y_{eyelid}^{former} is the analyzed eyelid y-motion resulted from applying the standard eye-state analysis algorithm and y_{eyelid}^{new} is the eyelid y-motion obtained from adding the inter-frame constraint. The eyebrow vertical action parameter, denoted as *fap*, ranges from $fap\big|_0$ to $fap\big|_{MAX}$.

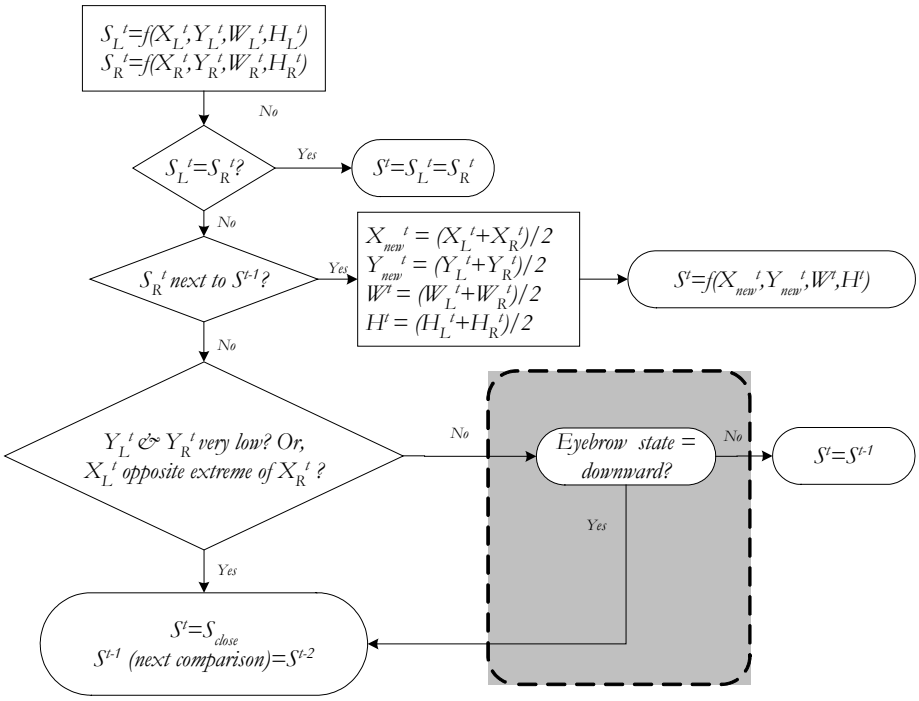


Figure IV-19. The basic Temporal State Diagram applied to eye analysis and built on only inter-eye constraints (Figure IV-3) can be complemented to take into account the data obtained from the eyebrow analysis.

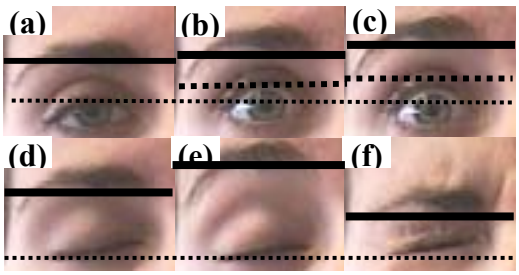


Figure IV-20. When the eye is closed (lower row), the eyelid change due to eyebrow action can be taken as some specific animation. When the eye is open (upper row) it must be taken into account to alter the standard y-motion of the eyelid

IV.5.1 Experimental evaluation and conclusions

During the eye-eyebrow cooperation tests, we have compared the evolution of the eyelid motion derived from the eye-state analysis against the results obtained from adding the influence of the eyebrow. In our practical scenario, see Figure IV-21, $fap\big|_0 = 0$, therefore (IV-13) is simplified to:

(IV-14) $$y_{eyelid}^{new} = y_{eyelid}^{former} + \mu' \cdot fap$$

with

$$\mu' = \frac{(y_{eyelid}\big|_{MAX} - y_{eyelid}^{former})}{fap\big|_{MAX}}.$$

The new eyelid vertical motion, y_{eyelid}^{new}, is the sum of its original standard value, y_{eyelid}^{former}, plus a term proportional to the eyebrow *fap* magnitude (ranged between 0 and fap_{MAX}). The proportion coefficient α is dependent on fap_{MAX}, the maximum value for the eyelid motion, and the analyzed eyelid location.

Looking at Figure IV-21 we can clearly notice the existing correlation between eye and eyebrow motion. The shadowed parts highlight the frames where the person was closing his eyes. The *fap* evolution, which depicts the action of the eyebrows, shows up that when eyebrows were moving up the eyes were open (Figure IV-20c is taken from frame 221), and when eyebrows where moving down, closed eyes were detected (Figure IV-20f is taken from frame 301).

Both Figure IV-21 and Figure IV-22 show that the added term to the final eyelid motion correctly estimates the increase in motion strength and it does not interfere with the eye analysis when no eyebrow motion is being detected.

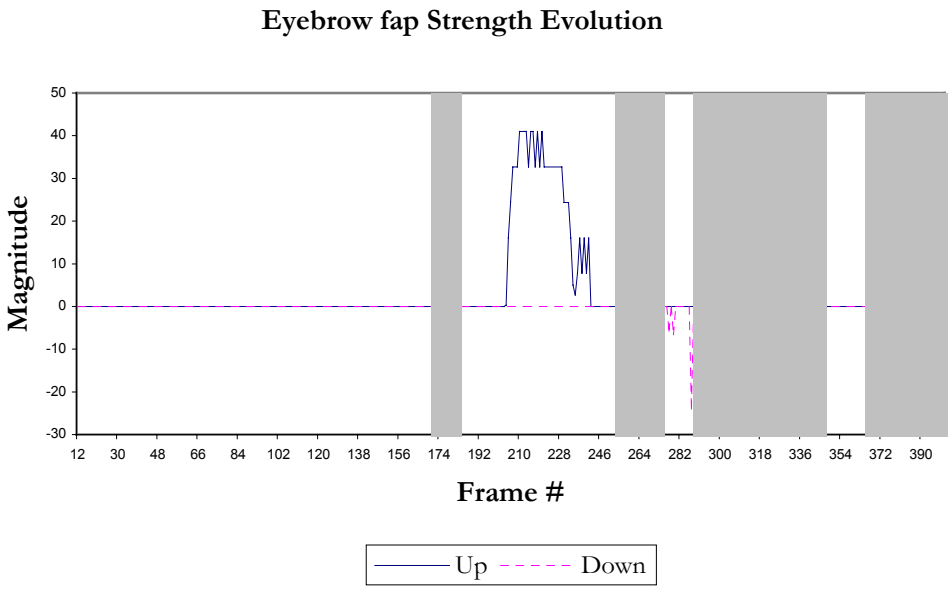


Figure IV-21. Eyebrow *fap* magnitude evolution (0-*fap*$|_{MAX}$) taken from sequence "NEON"

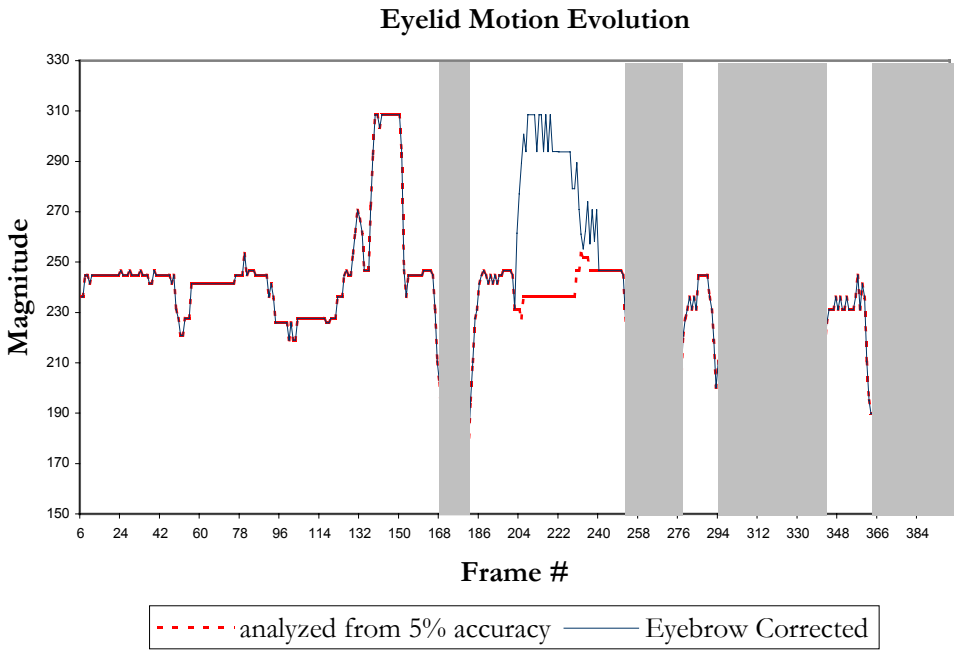


Figure IV-22. Eyelid standard analysis results compared to the corrected results after correcting the former with the eyebrow data. Analysis made on sequence "NEON"

IV.6 Analysis of mouth and lip motion

IV.6.1 Introduction

Mouth motion analysis has long been investigated in different domains. It has become a wide field of research because many of the techniques investigated aim at providing helpful tools for daily-constrained life, like for example, "automatic lip-reading" for the deaf. Related to the scope of research presented in this thesis, we focus on those algorithms that help to get more efficient ways of transmitting information in video-communication systems by substituting traditional face-to-face video by the animation of 3D clones of the speakers. Indeed, mouth analysis plays a major role in this scenario because the accuracy of mouth movement and the synchronization of the mouth actions with the audio generated during the conversation are crucial to obtain pleasant and natural communications.

We can consider that the overall movement of the mouth can be seen as the result of two factors:

$$M_{TOTAL} = M_{speech} + M_{expression}.$$

Where M_{speech} represents natural motion related to the articulation of sounds and phonemes while speaking and $M_{expression}$, is the part of the motion that shows the emotional expression and personal behavior of the individual. It is easy to discriminate the components of motion coming from expression when no speech is present. It is more difficult to deduce how actions from both natures interact when appearing together.

Looking at this issue from the inverse perspective, separating mouth motion components based on their nature (speech or expression) during the analysis, is also a hot topic of research in the Facial Animation community. During the creation of automatic motion to be synthesized on 3D head models (usually avatars), we combine phonetic motion information with expression motion data. This combination must be done in such a way that the resulting facial behavior acts in a natural human way. In most cases phonetic and expression interaction do not lead to pleasant and natural results. The knowledge of muscular interaction and natural facial behavior must be used to deduce the right motion and customize the animation generated after having analyzed the visual aspect of the mouth actions.

To develop a complete analysis framework, we have studied the advantages and drawbacks of most of the methods found in the literature. We have derived an approach that suits our scenario by developing a simple motion model to ensure that its action parameters will be robustly detected during the analysis, regardless of the environmental conditions.

Research background in this field

The aim of most of the analysis methods that were first developed to study global mouth motion was to generate animation patterns for the creation of automatic animation to use on avatars. Research was done to extract general motion behavior of mouths. In the one hand, these techniques aimed at matching speech and phonetic information obtained from real or synthetic speech with coherent and natural mouth movements (duality phoneme-viseme[2]); in the other hand, they focused on associating face expressions (joy, happiness, sadness, etc.) to some specific related mouth motion. Besides the hardware capture devices commonly used, analysis methods using image-processing were introduced. The first image-based techniques relied on markers and makeup that could easily be extracted from the images and then analyzed. The *Institut de la Communication Parlée* (ICP-Grenoble) has long been using this kind of methods to deploy realistic mouth motion on 3D head models (Odisio, Elisei, Bailly and Badin, 2001). Some other laboratories, like the Morishima Laboratory (Japan) also started developing mouth motion patterns from motion captured information before starting with other less invasive methods. It is, indeed, the invasive nature of the magnetic captors and markers along with the impossibility of easily deploying a usable system in non-testing environments that has pushed researchers towards the analysis of more or less naturally recorded video sequences, either monocular or from multiple points of view.

Many of the image analysis techniques focus on the study of lip motion evolution along the time. Some techniques use deformable contours, snakes or deformable models to define lip shape and then derive the lip action on the image (Lai, Ngo & Chan, 1996; Liévin, Delmas, Coulon, Luthon & Fristot, 1999). The models can be more or less complex and maybe based on some mesh structure that eases deriving motion information from the analyzed data. Chou, Chang and Chen (2001), for instance, use a mesh model to extract face animation parameters (FAPs). These FAPs are MPEG-4 compliant and thus usable for communication applications. This is one fine example of how developing flexible techniques of analysis enable practical usage of the obtained research results.

[2] Viseme: Lip, teeth and tongue natural motion synthesis (pre-established motion) when pronouncing a specific phoneme. See Pandzic, I. S., & Forchheimer, R. (Eds.). (2002).

Flexibility on the analysis implies a complete understanding of the analyzed image without controlling the environment conditions under which it has been recorded. This kind of analysis situation has brought up the development of several specific image processing techniques that study the mouth area in detail (Pahor and Carrato, 1999). The most performing results, those analysis techniques that offer the most realistic mouth animation are based on anatomic information of the face, mouth and jaw interaction. These methods relate the analyzed image data to some muscular motion parameters that exactly reproduce the behavior of the mouth. Morishima, Ishikawa and Terzopoulos (1998) started they research on monocular images following this idea. Unfortunately the methodology they exposed had one major weakness to be adapted to any circumstance: they use optical flow on their image processing, which makes their technique unstable to lighting changes.

Analyzing images to obtain information from the mouth is restricted by the fact that it is very difficult and sometimes even impossible to observe the motion of the inner parts: teeth and tongue. The animation of teeth and tongue is generally derived from the previous observation of natural human mouth action. King and Parent (2001) have developed a complete parametric tongue model for the speech animation. This kind of system along with phonetic analysis of speech coming from the speaker can help with reproducing the exact motion of the mouth while speaking. Unfortunately, its use is only possible when speech is present. Mouth analysis on images is fundamental to complement natural mouth synthesis in the absence of speech; it also helps to customize the standard mouth behavior provided by the phoneme-viseme mapping. Mouth motion image analysis techniques that completely neglect tongue and teeth interaction cannot aim at understanding real mouth behavior. Any robust image processing technique willing to obtain data to generate natural mouth movements must also consider analyzing teeth and tongue, in addition to lips.

Our approach to mouth analysis

The image processing techniques developed for mouth analysis try to take the most of some of the techniques revised in the literature, by analyzing the strengths and weaknesses and combining them in an efficient way. Based on the results already obtained in other similar studies of mouth motion we have developed our image analysis algorithms structurally:

(a) We start with a lip pixel color and intensity distribution study - analyzing the H&I distribution - of the mouth area. The goal of this analysis is to define those specific areas belonging to the mouth (teeth, lips and tongue) so they can be segmented and well separated. The complexity of the H&I based segmentation increases with very active mouth action and if lighting conditions are not known, therefore the algorithms involved are adapted along the time in a frame-by-frame basis.

(b) Pixel distribution, mouth shape, teeth or tongue location estimation can only be helpful for synthesis if there is a motion model associated to the analyzed image data. We have developed a mathematical mouth motion model to describe the movements from the image data obtained. This model is based on muscular interaction and tries to use the fewest number of control points susceptible of being extracted using flexible image processing. This model can only expect to estimate the projection of the motion on the analyzed plane; this implies that muscular information about mouth motion must be known on the synthesis part to create realistic reproductions. One of the differences of our model compared to others is that we try to also estimate jaw motion from the image of the mouth area.

(c) Finally, it is direct to relate the image data obtained from the segmentation process to the motion model needed to describe the action.

This section contains a new proposal for a mouth motion model that could be used in the context of our global facial feature analysis. We develop the model based in muscular intra-feature constraints. We also provide the technical evaluation of the image-processing algorithms proposed to extract the data related to this motion model. The vector nature of the data provided by the mouth motion template makes this technique susceptible of being extended to analyze any other view from the speaker in addition to the evaluated frontal head position.

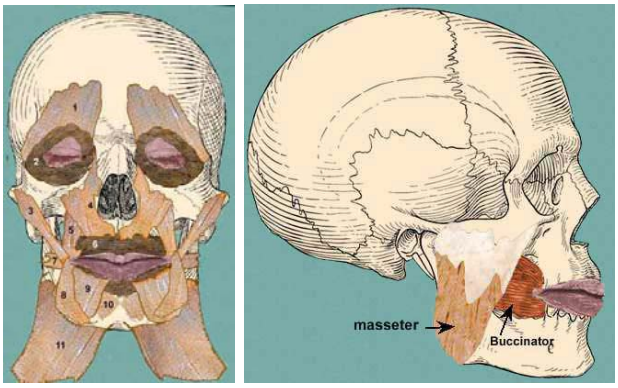
IV.6.2 Modeling lip motion with complete mouth action

Mouth movements derived from muscular interactions can be described in Cartesian 3D space as the total vector displacement of each of the points belonging to the components of the mouth along the x, y and z-axis. Muscular interaction during mouth motion is complex and the observed final shape of the mouth on the image plane only provides information about the x, and y components of the complete displacement. Depth information cannot be retrieved from a single perspective and therefore complete motion interaction must be deduced from the projected shape of the elements involved and the study of anatomical mouth behavior.

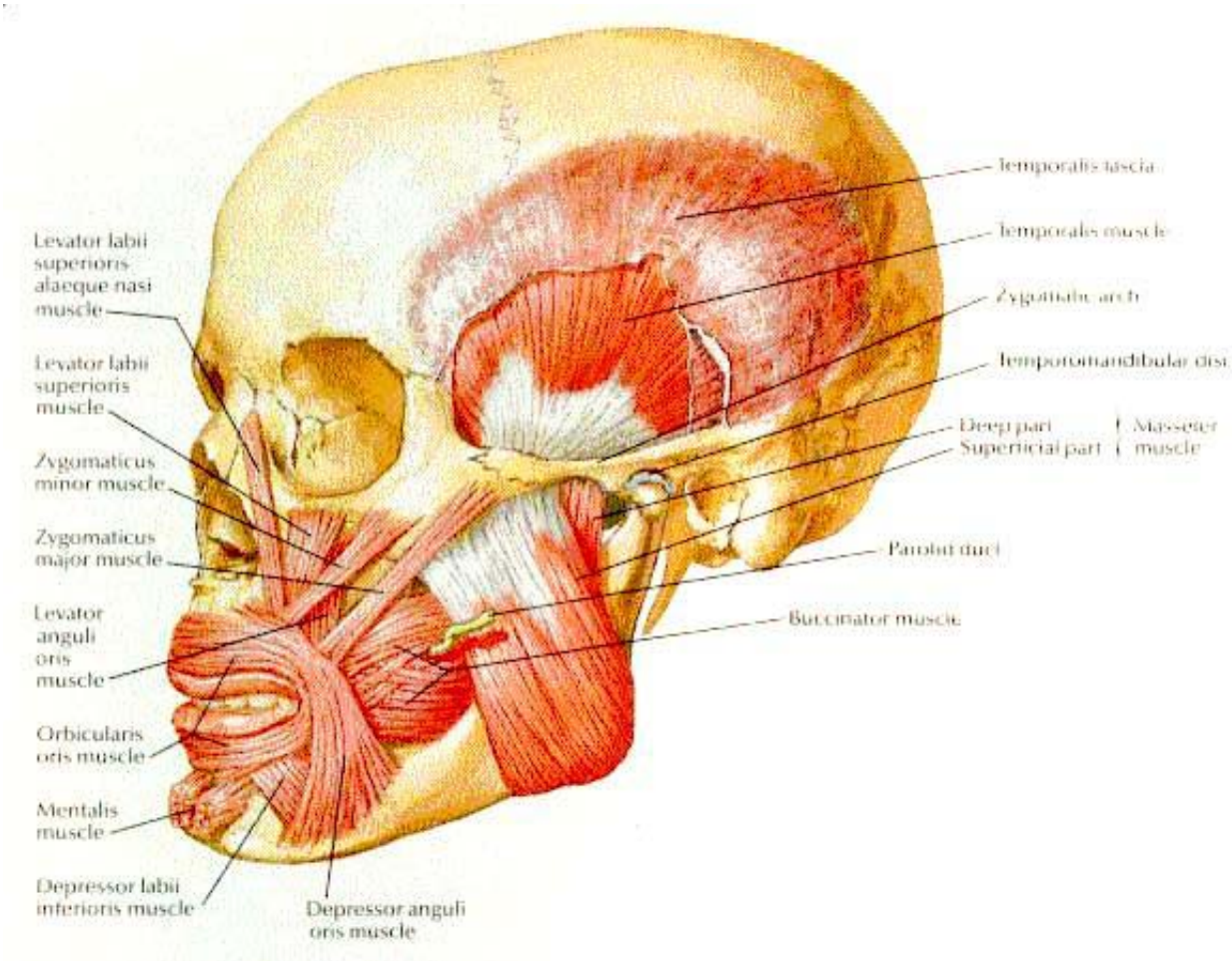
Knowing the dual nature of mouth comportment, monocular image analysis can be well complemented by the phoneme recognition of the person's speech. As seen in some practical scenarios (Goto, Kshirsagar & Magnenat-Thalmann, 2001; Chen, 2001), the extracted phoneme information from speech can be mapped to its viseme correspondence. Phonetic information can clearly help understanding mouth behavior but it is not enough. The phoneme-viseme mapping technique can only generate standard mouth behavior and no trace of emotion; expression or personalized action can be synthetically generated from it. Nevertheless, speech analysis is the only way to obtain motion information of those parts that remain invisible to the camera.

For our approach, we have studied and observed the interaction among muscles end bones of the head (Figure IV-23) that intervene when the mouth acts. We have mathematically modeled mouth muscular interaction to synthetically replicate mouth motion through the understanding of its projected appearance.

The interest of building a mathematical model for image analysis of mouth motion comes from the fact that if there is no reliable speech to be analyzed; the derivation of the model action from visually analyzed data can give us the best approximation to the studied motion. It also customizes mouth actions by generating more natural movements always directly related to the speaker individual way of speaking. This customization is very difficult to achieve by using only phonetic-based animation techniques.



(a) (b)



(c)

Figure IV-23. These illustrations present the bones and the muscles involved in the generation of mouth actions (from "Images of muscle and bones of the head", 2002; Simunek, 2003)

Two parts compose the motion model herein presented: the mathematical lip motion model and the structural jaw action model. To simplify the analysis requirements we have developed the lip model that contains the minimum control points needed to replicate major lip motion due to muscle actions. Most of current analysis solutions focus on the detailed tracking of lip movement along the time; fewer approaches take into account the motion from the teeth; and even less study the jaw action. The image analysis processing and its supporting motion model should not ignore the existence and interaction of teeth and jaw. Although difficult to analyze, their motion information is fundamental for the right interpretation and synthesis of mouth motion. The jaw model we have proposed tries to study the visual information extracted from the mouth analysis, basically teeth location, to deduce jaw motion.

Lip model

The proposed model for the lip motion tries to linearly describe the action of the muscles that play a major role in mouth motion (see Figure IV-23a to locate the muscles):

(a) Levator Labii Superioris [4]

(b) Zygomaticus Major [3]

(c) Joint action from: Zygomaticus Minor & Levator Anguli Oris [5]

(d) Joint action from : Buccinator & Risorius [7]

(e) Depressor Anguli Oris [8]

(f) Depressor Labii Inferioris [9]

(g) Orbicularis Oris [6]

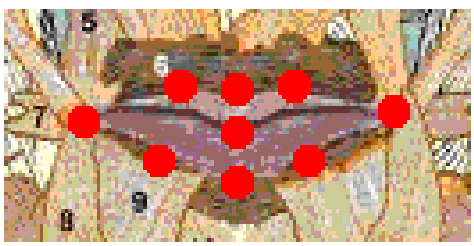


Figure IV-24. The chosen control points coincide with the ending extreme of the major muscles that intervene in mouth motion

After studying the anatomical structure of the mouth and its muscles we have detected eight critical points next to the lips (ten accounting for the replication of the interior part of the lip that moves correlated to the exterior part) where the muscles exert their final influence (look at Figure IV-24). We have developed the motion template presented in Figure IV-25.

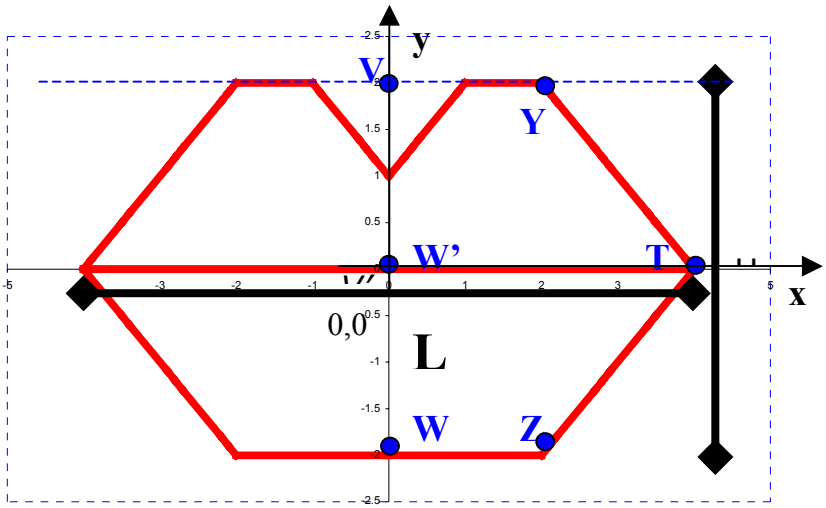


Figure IV-25. Schematic representations of the mouth lips and the control points acting on their left side

Since the movement of each of the selected control points is due to the action of specific groups of the major muscles whose ending point falls at that location of the lip area, we can derive the projected motion of this action studying the behavior of these control points along the time. Applying the following mathematical model, which gives a linear approximation of the behavior of the projected displacement (Δx & Δy) of the points belonging to the lip model due to muscular action exerted on the control points, we can reproduce some mouth shapes that estimate the muscular action created[3] :

From Buccinator & Risorius → control point: T, applied on both lips.

$$\Delta x_i = \left(\frac{x}{L/2}\right) \cdot \Delta x[T]$$

[3] This is the partial model to be applied to the left side part of the mouth, control points Y, Z & T must be duplicated on the right side and the motion model of the x-component of that side is symmetrical with respect to the y-axis.

From Zygomaticus Major or Depressor Anguli Oris → control point: T, applied on both lips.

$$\Delta y_i = \left(\frac{x}{L/2}\right) \cdot \Delta y[T]$$

From Levator Labii Superioris, Zygomaticus Minor & Levator Anguli Oris → control points: V&V', applied only on upper lip.

$$\Delta y_i = \left(\frac{x}{L/2}\right) \cdot \Delta y[V']$$

From Orbicularis Oris → control points Z&Y, applied to each lip separately

$$\text{Lower lip: } \Delta y_i = \Delta y[Z] \cdot \left(1 - abs\left(\frac{x}{L/4}\right) - 1\right) \;\&$$

$$\text{Upper lip: } \Delta y_i = \Delta y[Y] \cdot \left(1 - abs\left(\frac{x}{L/4}\right) - 1\right)$$

From Depressor Labii Inferioris → control point: W&W', applied only on the lower lip.

$$\Delta y_i = \left(\frac{x}{L/2}\right) \cdot \Delta y[W']$$

Control points Z & W do not only show the interaction amongst muscles but also the physical displacement of the jaw. Control point T also behaves differently when jaw rotation exists. It is very difficult to deduce jaw motion just by looking at the mouth evolution. A frontal view the face does not provide the best image perspective to appreciate jaw rotation but the teeth-lip location can help to deduce jaw motion.

There exist other motion template options for the mouth in the literature. Some of them, like the one proposed by Chen (2001), use a motion model that only controls the width and the height of the mouth; this model gives limited information regarding the action and cannot deduce complex mouth movements. Some others, like the one presented by Chou, Chang and Chen (2001), propose the use of the points that already belong to the mouth mesh for the synthesis. Their solution has the advantage of being scalable; they increase or decrease the number of control points depending on the synthetic model complexity. In their approach, they do not justify the number of the points they use from a muscular motion perspective and they do not determine the most suitable number of control points for the analysis. Our template tries to give a trade-off between simplicity, using a minimum number of control points, and performance, extracting the maximum motion information.

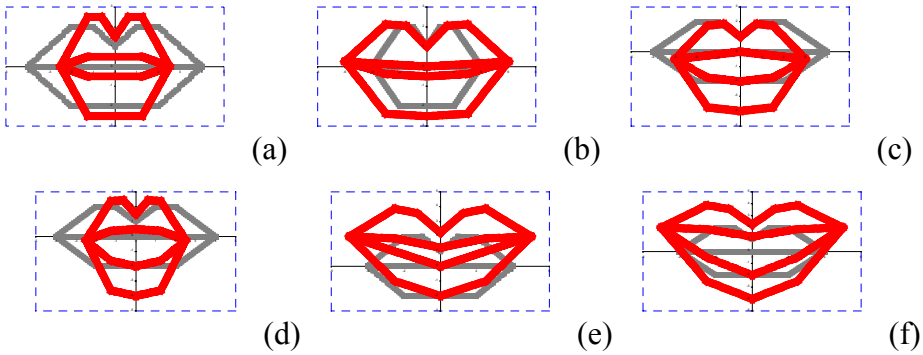


Figure IV-26. These images show the result (red color) of applying the displacements shown in the Table presented below onto the control points of a mouth on its neutral state (grey color). The global deformation of the mouth is obtained using the linear approximation proposed. Mouth proportions in neutral state: L=8 & H=4

(a)	Δx(T) =	-1.5;	Δy(T) =	0;	Δy(U') =	0;	Δy(U") =	0;	Δy(Z) =	-0.5;	Δy(Y) =	0.5;
(b)	Δx(T) =	2;	Δy(T) =	0.25;	Δy(U') =	0;	Δy(U") =	0;	Δy(Z) =	-0.5;	Δy(Y) =	0;
(c)	Δx(T) =	-1;	Δy(T) =	-0.5;	Δy(U') =	0;	Δy(U") =	-1;	Δy(Z) =	-1;	Δy(Y) =	0;
(d)	Δx(T) =	-1.5;	Δy(T) =	-0.25;	Δy(U') =	0;	Δy(U") =	-1;	Δy(Z) =	-1;	Δy(Y) =	0.5;
(e)	Δx(T) =	1;	Δy(T) =	2;	Δy(U') =	0.75;	Δy(U") =	0;	Δy(Z)	= 0;	Δy(Y) =	0.5;
(f)	Δx(T) =	1;	Δy(T) =	2;	Δy(U') =	0.75;	Δy(U") =	-1;	Δy(Z) =	-1;	Δy(Y) =	0.5;

In Figure IV-26, we present the visual results obtained from applying forces (represented by some specific magnitude values) on the control points of our motion model. With the designed motion template we can generate a rich variety of mouth expressions that will be sufficient to analyze standard mouth behavior.

Jaw motion: The importance of the reference coordinate system for global mouth motion understanding

The proposed mathematical model defines lip behavior independently of the origin of the forces for its deformation. The image analysis algorithm will have to evaluate if the movement is due to muscular action, jaw rotation or both. Due to natural constraints the actions coming from the jaw will be related to the degree of openness of the mouth and the proportion of teeth that are visible behind the lips.

The upper part of the mouth and the upper teeth remain always rigid and stable, their motion can only come from the rigid motion of the head; therefore they can be set as a proper point of reference for the non-rigid analysis. This information will help to deduce the complete action of the jaw, from the lips location regarding the upper and the lower teeth.

Figure IV-27 presents several jaw-motion combinations that illustrate the importance of tracking jaws during mouth analysis. In the first row, frontally similar lip-teeth projections are due to mouth-jaw motion of different nature. We show that visually similar lip behavior may come from very different mouth actions and that for complete mouth motion understanding, more than just lip tracking is needed.

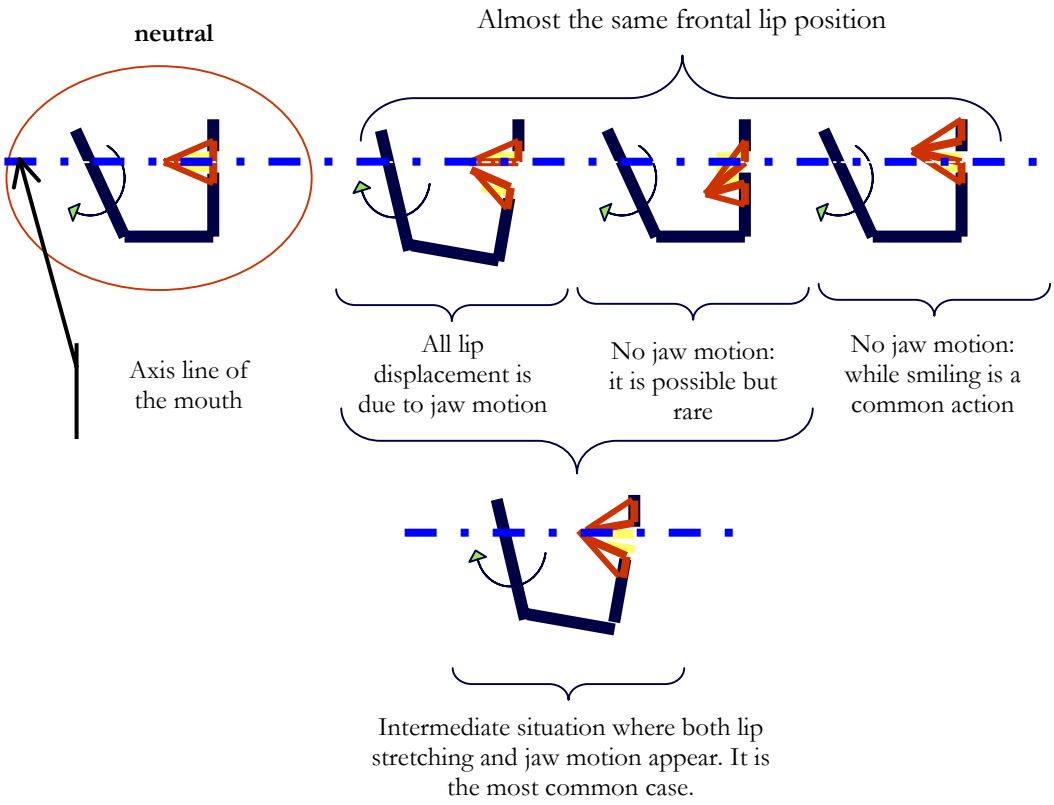


Figure IV-27. Schematic representation of how jaw motion influences the teeth-lip location plotted for some key mouth movements

Final Discussion

A lip/teeth/jaw model is capable of representing the most common mouth actions, strange mouth shapes are difficult to analyze and consequently are very difficult to model by justifying their muscular nature. For example, we consider unexpected mouth actions that count on the external intervention of the tongue.

Nevertheless, the complete feature analysis algorithm must be aware that the motion template is restricted to the analysis of lips, teeth and jaw but that there also exist other actions. Above all, it must detect when the mouth feature is not completely visible or does not show a shape directly derivable by the motion template. In such cases, the analysis must give the best approximation of the mouth shape, following natural mouth behavior constraints. It must not generate weird interpretations in the presence of elements that make the image analysis impossible.

IV.6.3 Image analysis of the mouth area: Color and intensity-based segmentation

To extract meaningful information from the mouth area of the face being analyzed, we have decided to build an image processing algorithm that robustly segments the region of the mouth in three major parts: lips, teeth and unknown dark area inside the mouth. Depending on the part we intend to segment the algorithm uses information from the color or the intensity of the pixel.

To build the segmentation algorithm we first study the histogram pixel distribution on HSI color space. Each part of the mouth shares common characteristics and are well located in each histogram distribution. The color/intensity histogram of the region of interest around the area is computed for each frame. From each of the histograms we deduce which zone of the image belongs to the lips, the teeth and the inner part of the mouth. Histograms computed frame by frame reflect the color and intensity distribution variations due to environmental changes, like for example the different lighting conditions of analysis. We have developed and tested two different algorithms to segment the mouth ROI:

(i) deducing the segmentation threshold on H and I from the evolution of the histogram

(ii) deducing the segmentation threshold on H and I from the statistical analysis of the histogram.

Histogram based algorithm for segmentation

To study the Hue and the Intensity pixel distribution of the mouth area we have plotted the histogram of the image of the mouth area for three different mouth configurations: close, open with no visible teeth, open with visible teeth.

We have preferred to use the simplified logarithmic hue transform proposed by Liévin and Luthon (2000)

$$\alpha = \arccos\left(\frac{2\cdot R - G - B}{(2\cdot\sqrt{(r-g)^2 + (r-b)\cdot(g-b)}}\right)$$

$$if\ \left(\tfrac{B}{I} > \tfrac{G}{I}\right) \quad \beta = 2\pi - \alpha$$

$$else \qquad \beta = \alpha$$

$$H = \frac{\beta\cdot 255}{2\pi}$$

rather than the traditional transform

$$H = \begin{cases} 256\cdot\dfrac{G}{R} & if\ G < R \\ 255 & if\ G \geq R \end{cases}$$

because it is more robust to lighting conditions. In their work, Liévin and Luthon prove that thanks to this transform, they are able to detect lips in situations where the traditional H transform could not. It also generates a histogram where the lip area is clearly differentiated from the other components of the mouth.

From the study of the different histograms we have concluded:

(a) The lower values of the I histogram clearly determine the dark area inside the mouth.

(b) We are no longer inside the dark area when the histogram values start to increase considerably.

(c) The H histogram shows two major hue concentrations. The biggest one belongs to the skin area of the ROI of the mouth and it decreases when the mouth is open, the smallest one belongs to the lip area and it remains stable regardless of the state of the motion of the mouth.

(d) Teeth are clearly detected observing the evolution of the hue histogram; the presence of teeth increases the amount of pixels around 255 strongly.

(e) The shape of the I histogram varies depending on the lighting conditions although the darkest area (lowest I values) always belongs to the inner part of the mouth.

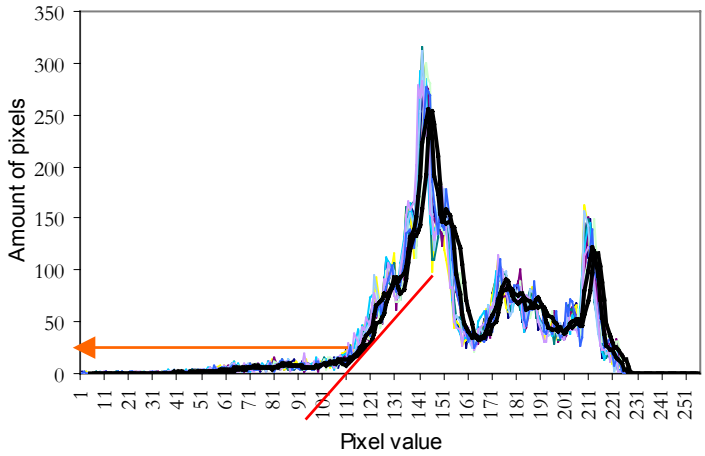
(f) The shape of the H histogram is quite stable against changes on the lighting conditions; it only changes depending on the natural skin characteristics of the individual being analyzed. If the color of the skin is close to the color of the lips, the distance between the maxima belonging to each of the two different H groups approach and they may even blend (making segmentation rather difficult). The overall location of the hue distribution (represented by its mean value) shifts to the left or the right also depending in the general skin characteristics of the person.

Figure IV-29 plots different I-histograms belonging to the same person obtained after analyzing frames of a closed mouth, an open mouth with teeth and an open mouth without teeth. Figure IV-30 shows the H-histograms for the same individual under the same analysis conditions.

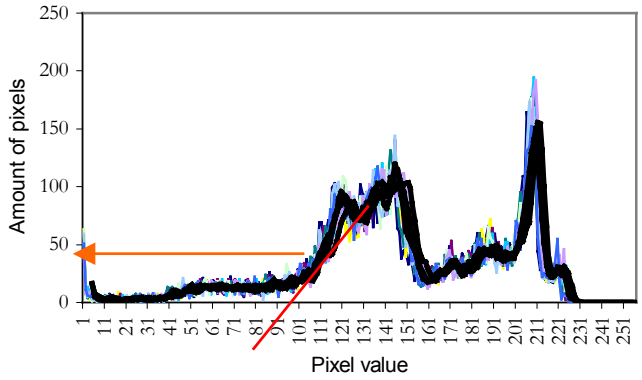
This histogram analysis is made on a tight area (ROI) surrounding the mouth in order to avoid the interference of undesirable external parts of the head like the hair, or objects from the background. From the histogram we can extract the threshold values that will determine the areas belonging to the lips, the dark inner part and the teeth.

After thresholds have been set we must label the pixels of the mouth zone. The labeling is performed on a wider area that covers a larger extension of the face (above all towards the chin). The area of analysis for this process is extended because mouth movements could be extreme and go outside the safe zone for the analysis of the mouth hue and intensity properties (see Figure IV-28).

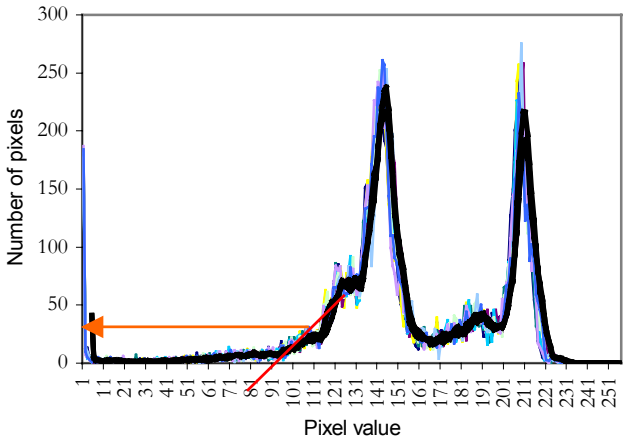
Figure IV-28. Areas delimited for the histogram study and for the mouth motion analysis



(a) closed mouth

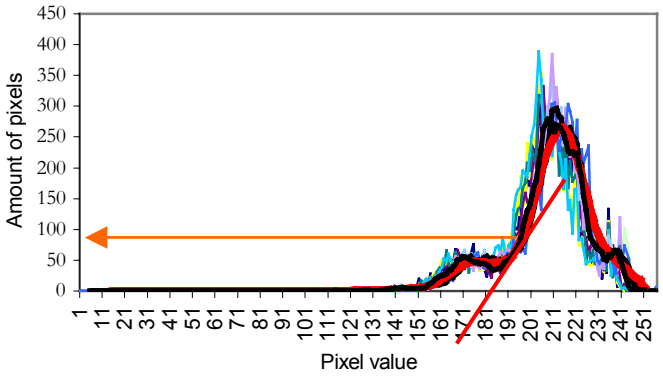


(b) open mouth with teeth

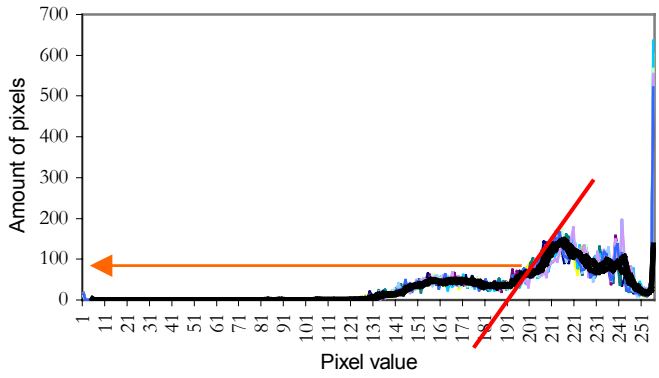


(c) open mouth no teeth

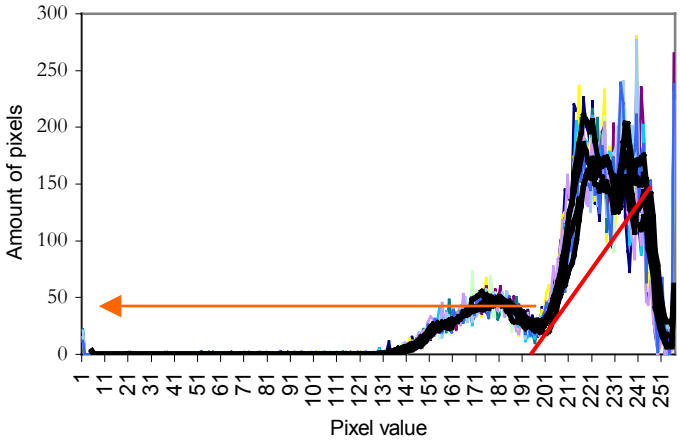
Figure IV-29. Intensity histograms



(a) closed mouth



(b) open mouth with teeth



(c) open mouth with no teeth

Figure IV-30. Hue histograms

To determine the thresholds for the labeling process we propose two possibilities:

- Tangent evolution of the histograms to determine the threshold for segmentation:

This approach analyzes the tangent of the histogram at each point looking for the sudden slope increase that determines the change from the darkness area to the rest of the mouth area in the I histogram. The same approach is utilized to detect the change of slope that separates the lip area from the rest of the mouth in the H histogram (See group of graphs in Figure IV-29 and Figure IV-30). Since histograms are very noisy the tangent computation is made after having smoothed the histogram by computing the local average of every value (black curves on the graphs). The two empirically chosen thresholds are:

$$th_I = I : \tan(Hist(I)) = 50\,\text{deg}$$

$$th_H = H : \tan(Hist(H)) = 40\,\text{deg}$$

These values have been chosen based on the analysis of mouths homogeneously illuminated and where clear segmentation could be done. Segmenting with these thresholds has been afterwards tested onto other images with unknown lighting conditions to check its robustness.

- Statistical analysis of the histograms to derive the thresholds for segmentation:

We propose another approach that intends to deduce the threshold for the segmentation process by using the statistical values of the histograms (min, max, mean, mode and standard deviation). Instead of analyzing the local behavior of the histogram to deduce the optimal value of the threshold we study its global characteristics. The two empirically chosen thresholds are:

$$th_I = \min(I) + \text{stdev}(I)$$

$$th_H = \text{mode}(H)\text{-stdev}(H)$$

Once again, these thresholds have been chosen after the analysis of mouth homogenously illuminated and they have been tested on the same block of face recorded under unknown condition to compare its robustness with the previous approach.

Results about the tests performed to study the performance of both approaches are given in subsection "*Evaluating the performance* (…)"

Labeling mouth parts

We segment the different parts that belong to the mouth following these criteria:

$$pixel_i \in lips \quad \text{if} \quad pixel_i H < th_H$$

$$pixel_i \in darkness \quad \text{if} \quad pixel_i I < th_I$$

Teeth determination (combination of hue and intensity knowledge)

1st approach: $pixel_i \in teeth \quad if \quad pixel_i \notin lips \;\&\; pixel_i H > th_{teeth}$

where th_{teeth} depends on the intensity distribution of the mouth area and has been chosen to be: $\max(H) - (\max(H)\text{-}\min(H))/20$.

2nd approach: $pixel_i \in teeth \quad if \quad pixel_i \notin lips \;\&\; pixel_i H > \text{mode}(H)$

Evaluating the performance of the proposed approaches:

To compare both approaches and deduct which one is more convenient we have utilized a video database provided by the Mathematics and Computer Science Department ate the UIB, University of the Balearic Islands, (2002). This database contains 60 videos of several seconds of faces of 60 people - majority of Caucasian skin characteristics. Each individual was recorded under homogeneous lighting but illumination conditions differ from case to case.

The following table presents the algorithmic performance for the first and second approach; it is shown as the percentage of video sequences where positive expected lip, teeth and darkness detection and segmentation was observed. The results were obtained after studying qualitatively the segmentation process.

Table IV-5

QUALITATIVE ANALYSIS EXPRESSED AS THE PERCENTAGE OF PERFORMANCE SUCCES ON UIB'S DB

	1st approach	2nd approach
Lips	59.32% (1) – 30.50% (2)	62.71% (1) – 23.72% (2)
Teeth	2.32% (1) – 23.25% (2)	72.27% (1) – 18.18% (2)
Darkness	66.10% (1) – 22.03% (2)	83.05% (1) – 13.55% (2)

COMPARING THE 1ST APPROACH WITH THE SECOND APPROACH MEASURING THE SUCCESS AT:
(1) RIGHT DETECTION & RIGHT SEGMENTATION IN MOST FRAMES
(2) RIGHT DETECTION BUT INCORRECT SEGMENTATION IN MOST FRAMES

We refer the reader to Appendix IV-I to find details on the tests performed, where results and comments on the characteristics of the people analyzed have been recorded.

Figure IV-31 contains some shots from a few sequences of the database used where the areas labeled as being lips have been surrounded and the dark part has been detected and marked in black.

Conclusions

Comparing approaches:

After studying the two approaches proposed, the second gives better results. The stability of the histogram is too weak locally and the applied smoothing depends on the analysis conditions. Changing the degree of smoothness also implies changing the threshold values. Therefore thresholds obtained from the analysis of the tangent evolution of the histogram are not stable enough. Statistical data seem to remain more robust to all cases and are less dependent on noisy characteristics of the histogram.

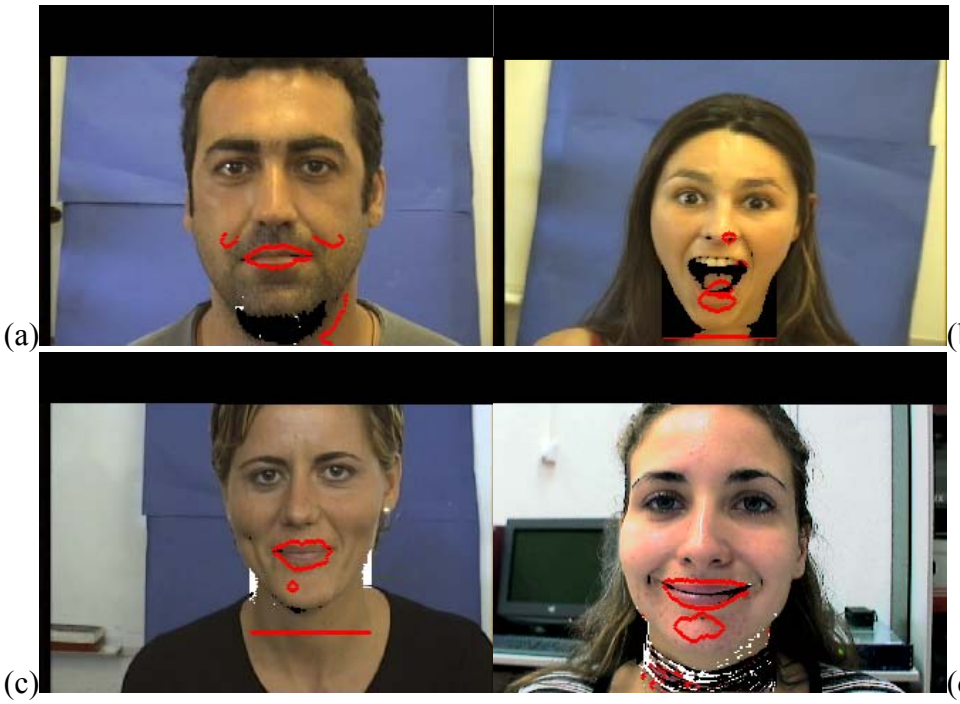


Figure IV-31. Screen shots of some of the 60 videos analyzed for the tests. In the images the lips areas are surrounded by read and the lip separation and darker inner part of the mouth detected in black. The second approach was used for the analysis of the presented shots. (a), (c) and (d) show the right regimentation of the lips. (a), (b) and (d) illustrate the correct segmentation of the darkest part of the mouth. In the four cases we also observe the segmentation of other unwanted areas, thus creating artifacts that ought not to be taken into account during motion analysis

- Regarding the study of the hue characteristics of the image:

Lip segmentation based on hue analysis is an efficient technique as long as there is a noticeable tone difference between skin and lips. There are cases where this tone difference does not exist: tanned people, lips so thin they do not appear on the image, etc. Hue characteristics can also change with the influence of the color nature of the source of lighting on the face (neon light versus yellow bulb light, for instance). The change of color on the surface could interfere and change the overall hue distribution histogram.

- Detection of teeth:

The first approach, which is based on plain histogram characteristic study, is able to detect teeth if they are present but it does not segment the complete teeth zone well.

The second approach, based on the study of the histogram statistics, detects and segments the complete teeth area but produces over segmentation introducing noise on the segmented image by assigning the label of teeth to some sparse pixels on the skin but never on the lips.

To extract useful and complete information the second approach should be used but the analysis process must take into account the existence of noise data situated around the mouth

- Reliability of the obtained segmentation:

If the influence of the lighting on the hue characteristics is known all the segmented data (from teeth, darkness and lips) have the same level of reliability. Since the natural source of lighting is not generally known, intensity values determining darkness and teeth are in most cases the most reliable information and motion analysis should start by extracting useful information from these segmented parts. Statistical analysis methods are more reliable than distribution-based analysis techniques.

V Extending the Use of Frontal Motion Templates to any other Pose

Assuming that we control the user's pose is an important restriction when doing analysis for videoconferencing purposes. In real scenarios this assumption becomes a major drawback. Yet, many virtual telecommunication schemes try to avoid the pose-expression coupling issue by minimizing its effects. In this case, for the analysis algorithms to remain robust, these schemes only allow the user slight changes in pose.

In this chapter we describe a technique that extends the previously detailed 'near-to-frontal' feature analysis algorithms to any given pose of the head to allow the user more freedom of movement in front of the camera.

V.1 Introduction

In the literature we have found two major approaches to adapt frontal facial motion and expression analysis algorithms to any pose:

1. **Designing one feature template per each pose**: after developing and testing motion templates on frontal faces, they are redefined based on different predetermined face poses. For instance, this is the solution given by Tian, Kanade and Cohn (2001). They overcome the pose limitation in their analysis by defining a "multiple state face model", where different facial component models are used for different head states (left, left-front, right, down, etc.). This analysis strategy is limited. The complexity of this solution increases with the number of states, which will be large if much freedom of movement is wanted.
2. **Rectifying the input image**: the image to be analyzed is transformed to obtain an approximation of the face viewed from a frontal perspective. Then, the image processing algorithms defined for frontal faces analyze this new image to obtain the corresponding feature templates (Chang, et al., 2000). This solution works nicely for slightly rigid movements. Significant rotations and translations cannot be compensated with simple image transformations because:
    - the appearance of each face feature does not only depend on the projection due to the pose but also on its 3D shape, therefore a 2D rectification done without acknowledging the 3D nature of the feature cannot be accurate;
    - the rectified image may be missing some areas occluded on the original image; and
    - 2D rectification may alter the lighting perception and the anatomical shape of the features, which is very important in feature-based image analysis.

We propose a different approach to do frontal motion analysis adaptation. Our solution uses the knowledge of the head pose and the user physiognomy to interpret the expressions in 3D space instead of on the image plane.

V.2 Feature Template Adaptation

The algorithmic adaptation process follows these steps:

(a) We first redefine the motion model, region of interest (ROI) and image processing parameters associated with each feature template in 3D, assuming that the head is facing the camera, in its neutral pose.

(b) Next, we use information regarding the rigid-motion of the head on the analyzed frame to project the 3D defined ROIs and other analysis constraints of each feature on the video image. Then, we apply the image processing required to extract the data for the model.

(c) Finally, we inverse the projection and the pose transformation of those data to obtain their 3D equivalent that will be ready to be contrasted against the motion models already defined in 3D.

Figure V-1 presents a graphical interpretation of the adaptation process applied to the analysis of the eye features.

For the adapted analysis we must define:

(i) **an observation model**. To develop the adaptation, we consider our analysis scenario: one 3D object (head) in front of one camera that acquires the video images that are analyzed. We establish the neutral pose of the head, when the face is completely centered on the image and statically looking towards the camera center. The observation model mathematically describes the relationship between the coordinates of the head object in its neutral pose and the final view of the face on the video or image plane. This mathematical model enables us to interpret data associated to the head from the modeled 3D space to the image 2D space and vice versa.

(ii) **a 3D model of the head.** The template motion analysis techniques defined for a frontal view assume to know the location of the face features on the image plane. Similarly, during the adaptation we need to know the physiognomy of the person facing the camera so to be able to accurately locate the features in 3D space. We use the vertex data of a highly realistic 3D representation and of the person and its model texture to determine the position of the ROI of each feature.

(iii) **a convenient surface approximation per feature.** The analysis templates are originally defined to analyze the information on the image plane. We can easily adapt these motion models by directly mapping each one of them on a

surface parallel to the image plane and situated on the determined location of the feature on the 3D head in its neutral pose. To obtain the most suitable parallel plane, we develop the linear approximation of the surface that covers the region of motion of each feature.

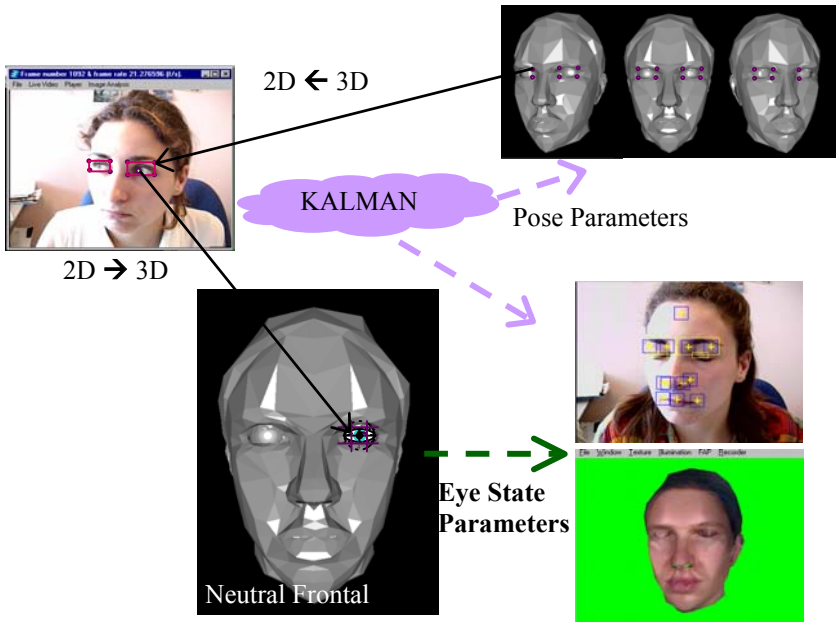


Figure V-1. This diagram illustrates the general adaptation process applied to the eye analysis algorithm. First, the vertices that define the 3D ROI on the linear surface model are projected onto the image plane. Then the image-processing algorithm retrieves the desired information by analyzing inside the delimited area. To understand the motion of the feature, data are interpreted in 3D space, over the motion model that has been defined on the linear surface approximation of the eye feature viewed from a frontal perspective. Once the motion is interpreted it can be reproduced on a synthetic head model. The projection and the understanding of the image information are possible because the system controls the 3D pose of the head with respect to the camera

V.3 Observation Model

The observation model utilized to relate the head in its neutral pose (facing the camera) and its projected representation takes into account the rigid motion (translations and rotations) of the head observed from the reference origin and the projection due to the camera. Although the acquisition camera is not calibrated because we do not control the nature of the input sequences, we can still consider that it makes a perspective projection, and not an orthogonal one.

The reference origin is situated along the optical axis of the camera and on the image plane. The image plane represents the video image where the face is focused. The focal distance *F*, represents the distance from that plane to the optical center of the camera. To describe the rigid motion of the head we have defined three translations, along the X, Y and Z-axis, and three rotations, around these same axes. Figure V-2 presents the graphical interpretation of the model and the orientation of the reference axes.

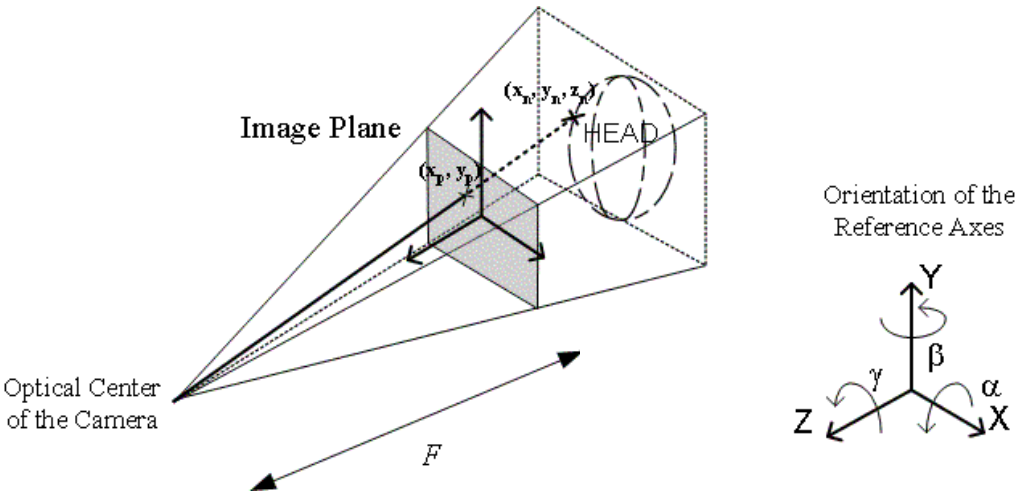


Figure V-2. Schema of the reference system and camera model used (of focal length *F*) for the adaptation process. It establishes the relationship of a point in the Euclidean space $\mathbf{x}_n = (x_n, y_n, z_n)^T$ and its projected counterpart on the camera image plane $\mathbf{x}_p = (x_p, y_p)^T = \left(\frac{F \cdot x_n}{F - z_n}, \frac{F \cdot y_n}{F - z_n} \right)^T$. The axis orientation is such that the camera only sees the negative part of the Z-axis

V.3.1 Mathematical description of the model

We describe points using their homogenous coordinates to be able to describe a perspective transform linearly and easily derive the relationship between 3D neutral coordinates and 2D projections.

Any vector $(x, y, z, w)^T$ is a homogenous point if at least one of its elements is not 0. If *a* is a real number and is not 0, $(x, y, z, w)^T$ and $(ax, ay, az, aw)^T$ represent the same homogenous point.

The relationship between a point in 3D or 2D Euclidean space and its homogenous representation is:

$(x, y, z)_{3D} \rightarrow (x, y, z, 1)_H$ and $(x, y)_{2D} \rightarrow (x, y, 0, 1)_H$

We can obtain the Euclidean representation of a homogenous point only if $w \neq 0$:

$(x, y, z, w)_H \rightarrow (x/w, y/w, z/w)_{3D}$ and $(x, y, z, w)_H \rightarrow (x/w, y/w)_{2D}$

Transformation matrices that describe rigid motion

- Translation following vector $(t_X, t_Y, t_Z)^T$:

$$\mathbf{T}_{(t_X, t_Y, t_Z)} = \begin{bmatrix} 1 & 0 & 0 & t_X \\ 0 & 1 & 0 & t_Y \\ 0 & 0 & 1 & t_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

- Rotation by an angle of α rad around the X-axis:

$$\mathbf{R}_{\alpha, X} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha & 0 \\ 0 & s_\alpha & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

- Rotation by an angle of β rad around the Y-axis:

$$\mathbf{R}_{\beta, Y} = \begin{bmatrix} c_\beta & 0 & s_\beta & 0 \\ 0 & 1 & 0 & 0 \\ -s_\beta & 0 & c_\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

- Rotation by an angle of γ rad around the Z-axis:

$$\mathbf{R}_{\gamma,Z} = \begin{bmatrix} c_\gamma & -s_\gamma & 0 & 0 \\ s_\gamma & c_\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Observation equation

The final location of the head regarding the reference origin is obtained applying the translation and rotation matrices upon the coordinates of the head in its neutral pose.

$$\mathbf{x}_{trans}{}^T = \mathbf{G} \cdot \mathbf{x}_n{}^T$$

where $$\mathbf{G} = \mathbf{T}_{(t_X,t_Y,t_Z)} \cdot \mathbf{R}_{\alpha,x} \cdot \mathbf{R}_{\beta,y} \cdot \mathbf{R}_{\gamma,z}$$

Then, the position "head is facing the camera" is defined when $(t_X, t_Y, t_Z) = (0,0,0)$, $\alpha = 0$, $\beta = 0$ and $\gamma = 0$.

The observed projection on the image plane is:

(V-1) $$\mathbf{x}_p{}^T = \mathbf{P}_F \cdot \mathbf{T}_{(0,0,-F)} \cdot \mathbf{x}_{trans}{}^T,$$

where $$\mathbf{P}_F \cdot \mathbf{T}_{(0,0,-F)} = \begin{bmatrix} F & 0 & 0 & 0 \\ 0 & F & 0 & 0 \\ 0 & 0 & -1 & -2F \\ 0 & 0 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -F \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} F & 0 & 0 & 0 \\ 0 & F & 0 & 0 \\ 0 & 0 & -1 & -F \\ 0 & 0 & -1 & F \end{bmatrix}$$

represents the complete projection from the combination of the perspective projection matrix, $\mathbf{P}_F$, whose origin is located on the optical center of the camera and the translation $-F$ along the Z-axis, $\mathbf{T}_{(0,0,-F)}$, that relocates the origin of the reference axis on the image plane (just like it is in our observation model in Figure V-2).

We obtain the following expression to relate the homogenous coordinates of the points belonging to the head in its neutral pose and their observed equivalent representation on the image plane:

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ w_p \end{bmatrix} = \begin{bmatrix} Fc_\beta c_\gamma & -Fc_\beta s_\gamma & Fs_\beta & Ft_X \\ F(c_\alpha s_\gamma + s_\alpha s_\beta c_\gamma) & F(c_\alpha c_\gamma - s_\alpha s_\beta s_\gamma) & F(-s_\alpha c_\beta) & Ft_Y \\ c_\alpha s_\beta c_\gamma - s_\alpha c_\gamma & -c_\alpha s_\beta c_\gamma - s_\alpha c_\gamma & -c_\alpha c_\beta & -t_Z - F \\ c_\alpha s_\beta c_\gamma - s_\alpha c_\gamma & -c_\alpha s_\beta c_\gamma - s_\alpha c_\gamma & -c_\alpha c_\beta & -t_Z + F \end{bmatrix} \cdot \begin{bmatrix} x_n \\ y_n \\ z_n \\ w_n \end{bmatrix}$$

After transforming the homogenous coordinates to Euclidean space coordinates ($w_n = 1$ and z_p is not taken into account), the observation $(x_p, y_p)_{2D}{}^T$ on the image plane of a given point $(x_n, y_n, z_n)_{3D}{}^T$ belonging to the face in its neutral pose is:

(V-2)
$$\begin{bmatrix} x_p \\ y_p \end{bmatrix}_{2D} = \frac{F}{N}\begin{bmatrix} c_\beta c_\gamma x_n - c_\beta s_\gamma y_n + s_\beta z_n + t_X \\ (s_\alpha s_\beta c_\gamma + c_\alpha s_\gamma)x_n - (s_\alpha s_\beta s_\gamma - c_\alpha c_\gamma)y_n - s_\alpha c_\beta z_n + t_Y \end{bmatrix}$$

$$N = (c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma)x_n + (-c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma)y_n - c_\alpha c_\beta z_n - t_z + F$$

V.4 Model Inversion

To find which is the original neutral coordinate of a given point from the video image of a facial feature, we need to invert the previous projection and pose transformations.

- Rigid motion transformation inverse:

$$\mathbf{x}_n^{\,T} = \mathbf{G}^{-1} \cdot \mathbf{x}_{trans}^{\,T}$$

$$\mathbf{G}^{-1} = \mathbf{R}_{\gamma,z}^{\;-1} \cdot \mathbf{R}_{\beta,y}^{\;-1} \cdot \mathbf{R}_{\alpha,x}^{\;-1} \cdot \mathbf{T}_{(t_X,t_Y,t_Z)}^{\;-1} = \mathbf{R}_{-\gamma,z} \cdot \mathbf{R}_{-\beta,y} \cdot \mathbf{R}_{-\alpha,x} \cdot \mathbf{T}_{(-t_X,-t_Y,-t_Z)}$$

$$= \begin{bmatrix} c_\gamma c_\beta & s_\gamma c_\alpha + c_\gamma s_\beta s_\alpha & s_\gamma s_\alpha - c_\gamma s_\beta c_\alpha & -c_\gamma c_\beta t_X - (s_\gamma c_\alpha + c_\gamma s_\beta s_\alpha) t_Y - (s_\gamma s_\alpha - c_\gamma s_\beta c_\alpha) t_Z \\ -s_\gamma c_\beta & c_\gamma c_\alpha - s_\gamma s_\beta s_\alpha & c_\gamma s_\alpha + s_\gamma s_\beta c_\alpha & s_\gamma c_\beta t_X - (c_\gamma c_\alpha - s_\gamma s_\beta s_\alpha) t_Y - (c_\gamma s_\alpha + s_\gamma s_\beta c_\alpha) t_Z \\ s_\beta & -c_\beta s_\alpha & c_\beta c_\alpha & -s_\beta t_X + c_\beta s_\alpha t_Y - c_\beta c_\alpha t_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This inverse transform defines a bijective operation in the Euclidean 3D space, one given neutral point, $\mathbf{x}_n^{\,T}$, and relates to one and unique transformed point, $\mathbf{x}_{trans}^{\,T}$.

- Projection inverse:

$$\mathbf{x}_{trans}^{\,T} = [\mathbf{P}_F \cdot \mathbf{T}_{(0,0,0-F)}]^{-1} \cdot \mathbf{x}_p = \begin{bmatrix} 1/F & 0 & 0 & 0 \\ 0 & 1/F & 0 & 0 \\ 0 & 0 & -1/2 & -1/2 \\ 0 & 0 & -1/2F & 1/2F \end{bmatrix} \cdot \begin{bmatrix} x_p \\ y_p \\ z_p \\ w_p \end{bmatrix}$$

This inverse transform does not define a bijective operation in the Euclidean 3D space. Inverting the projection generates a straight line that goes through the optical center of the camera and that defines the ray of possible solutions in 3D space for a given projected point.

By isolating the neutral coordinates in Equation (V-2):

(V-3)

$$\begin{bmatrix} (c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma) x_p - F c_\beta c_\gamma & (-c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma) x_p + F c_\beta s_\gamma & -c_\alpha c_\beta x_p - F s_\beta \\ (c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma) y_p - (s_\alpha s_\beta c_\gamma + c_\alpha s_\gamma) F & (-c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma) y_p + (s_\alpha s_\beta s_\gamma - c_\alpha c_\gamma) F & -c_\alpha c_\beta y_p + s_\alpha c_\beta F \end{bmatrix} \cdot \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix}_{3D} = \begin{bmatrix} (t_X - x_p) F + t_Z x_p \\ (t_Y - y_p) F + t_Z y_p \end{bmatrix}$$

we point out the non bijective nature of the observation model. The accurate 3D coordinates of one specific point are the solution to the intersection of the ray of solutions with the 3D surface to which the point belongs.

V.4.1 Feature surface approximation

To be able to interpret the data obtained from the image plane we need to know the neutral 3D surface to which these data belong. This surface will be the needed constraint that will enable the system to find a solution to the inversion of the observation model.

The motion template will be directly adapted on this surface, therefore the desired characteristics of each feature surface are:

(1) it must completely cover the region of interest of the facial feature;

(2) it has to be as close as possible to the real feature surface ; and

(3) it must be defined on the head on its frontal position as a parallel surface to the image plane.

Forcing these requirements we ensure that there will be a bijective relationship between the described neutral 3D space of the template motion model and the 2D image of each feature on the video.

To obtain the desired surface we study the anatomical structure of the feature; then, we model it with a surface that covers the analysis area and that is tangent to the motion that is going to be analyzed; and finally, we give a linear approximation of such a surface that is parallel to the image plane. A realistic 3D head model of the person being analyzed is used to study the specific physiognomy. This 3D head model will also be the reference 3D object that will determine the observation model used, and its focal length F.

The ROI of each feature and its template parameters are defined on the obtained plane (Sections V.5 and V.6 cover both issues in detail).

Inversion solution for a general plane

The general expression of a plane is: $Ax + By + C\tilde{z} + D = 0$. If it is described in homogenous coordinates, it can be seen as the solution to the equation $\mathbf{p}^T \cdot \mathbf{x}_n = 0$, where $\mathbf{p}^T = [A \quad B \quad C \quad D]^T$.

This surface constraint is added to equation system (V-3) obtaining

(V-4)

$$\begin{bmatrix} A & B & C \\ (c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma)x_P - F c_\beta c_\gamma & (-c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma)x_P + F c_\beta s_\gamma & -c_\alpha c_\beta x_P - F s_\beta \\ (c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma)y_P - (s_\alpha s_\beta c_\gamma + c_\alpha s_\gamma)F & (-c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma)y_P + (s_\alpha s_\beta s_\gamma - c_\alpha c_\gamma)F & -c_\alpha c_\beta y_P + s_\alpha c_\beta F \end{bmatrix} \cdot \begin{bmatrix} x_n \\ y_n \\ \tilde{z}_n \end{bmatrix}_{3D} = \begin{bmatrix} -D' \\ (t_X - x_P)F + t_Z x_P \\ (t_Y - y_P)F + t_Z y_P \end{bmatrix}$$

whose solution gives a unique correspondence $(\mathbf{x}_n{}^T)_{3D}$ for a given $(\mathbf{x}_p{}^T)_{2D}$ observed on the image plane.

Inversion solution for a plane parallel to the image plane

Surfaces parallel to the image plane are those that have $A = B = 0$, $C \neq 0$ and $D \neq 0$.

Imposing $z_n = \dfrac{-D}{C} = M$, equation system (V-4) is then simplified:

(V-5)
$$\begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \cdot \begin{bmatrix} x_n \\ y_n \end{bmatrix}_{3D} = \begin{bmatrix} a_4 - Ma_3 \\ b_4 - Mb_3 \end{bmatrix} \;\&\; z_n = M$$

$$\begin{aligned} a_1 &= -x_p \cdot (c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma) + F \cdot c_\beta c_\gamma & b_1 &= -y_p \cdot (c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma) + F \cdot (s_\alpha s_\beta c_\gamma + c_\alpha s_\gamma) \\ a_2 &= x_p \cdot (c_\alpha s_\beta s_\gamma + s_\alpha c_\gamma) - F \cdot c_\beta s_\gamma & b_2 &= y_p \cdot (c_\alpha s_\beta s_\gamma + s_\alpha c_\gamma) + F \cdot (-s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma) \\ a_3 &= x_p \cdot (c_\alpha c_\beta) + F \cdot s_\beta & b_3 &= y_p \cdot (c_\alpha c_\beta) - F \cdot s_\alpha c_\beta \\ a_4 &= -x_p \cdot (t_Z - F) - F \cdot t_X & b_4 &= -y_p \cdot (t_Z - F) - F \cdot t_Y \end{aligned}$$

To simplify the visual presentation, c_φ stands for $\cos(\varphi)$, s_φ stands for $\sin(\varphi)$ and t_φ stands for $\tan(\varphi)$, ^{p} for projected coordinates and $_n$ for neutral 3D-coordinates. Capital letters represent matrices and vectors, and lower case letters coordinates and vector components. F stands for the focal distance value of the projection system.

V.5 3D Definition of Feature ROIs

Defining the ROIs over the 3D head model allows the analysis system to control the evolution and changes of the analysis areas on the video sequence caused by the changes on the head pose. We obtain the area to analyze by projecting these 3D regions on the image plane. This procedure automatically reshapes the areas on the video images following the feature appearance.

The expression of the area of one feature (Figure V-3) is:

$$Area = base \cdot height = \mathbf{A}_p \cdot \mathbf{B}_p \cdot \sin(\arccos(\frac{(x_p^3 - x_p^1)(x_p^2 - x_p^4) + (y_p^3 - y_p^1)(y_p^2 - y_p^4)}{\mathbf{A}_p \cdot \mathbf{B}_p}))$$

where $\mathbf{A}_p = \left\| 3_p - 1_p \right\|$, $\mathbf{B}_p = \left\| 2_p - 4_p \right\|$ and $i_p = (x^i{}_p, y^i{}_p)$.

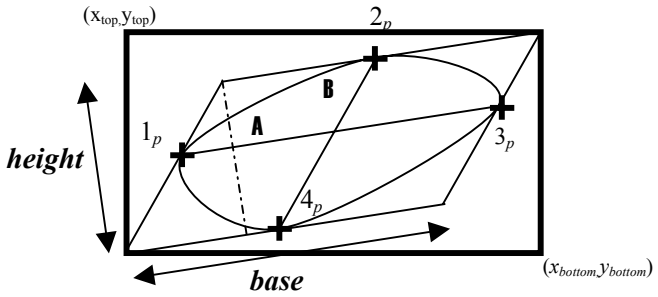


Figure V-3. Example of deformation and framing of one feature ROI

Controlling the size and shape of the feature's projection also permits to foresee if the information that will be obtained from the analysis of the targeted zone will be relevant. After observing the graphs plotted in Figure V-4, we realize that the area of analysis projected on the image plane basically reaches its maximum value when the head is around its neutral pose, although the exact maximum depends on the ROI's 3D location. Then, it presents a decreasing evolution when the head moves. Motion along the optical axis of the camera does not follow this global behavior. This increasing trend in the ROI area represents the consequence of approaching the camera.

If the ROI on the image plane decreases, the amount of details that the face feature will present on the image will diminish as well. Our analysis algorithms may not be able to extract data from certain features if they are too small. Knowing the pose and how the feature size behaves, it can help to prevent performing analysis that will not succeed. For instance, we can define a threshold area under which the algorithm will not perform because we consider that there will not be enough visible surface. This

threshold is dependent on the practical implementation of feature analysis system: image-processing technique used, size of the input video, etc.

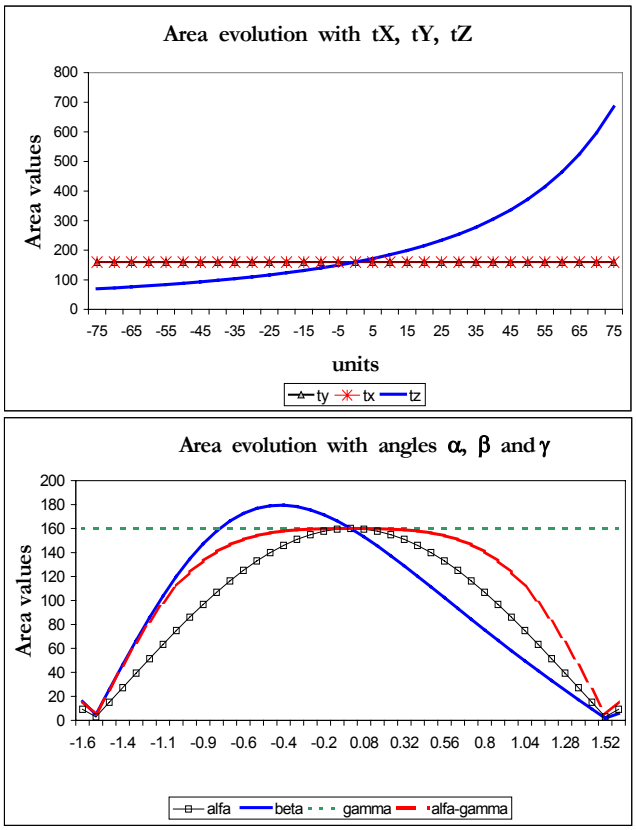


Figure V-4. These graphs depict the evolution of the feature's projected ROI depending on the pose of the head. We observe the influence of each of the pose parameters independently and the angles $\alpha - \gamma$ jointly. The studied observation model simulates the ROI of an eye. It has $F = 15 \cdot \mathbf{A}$ units, the major axis ($\mathbf{A}$) and the minor axis ($\mathbf{B}$) are defined from: $1_n = (20,0,4.8)$; $2_n = (25,7.5,4.8)$; $3_n = (30,0,4.8)$ and $4_n = (25,-7.5,4.8)$; the area of the ROI surface computed in 3D is 150 units

For practical purposes, to make the deformed areas more suitable for image analysis we enclose them in video analysis rectangles $(x_{top}, y_{top}) \rightarrow (x_{bottom}, y_{bottom})$:

$$(x_{top}, y_{top}) = (\min(x) \in ROI, \max(y) \in ROI);$$

$$(x_{bottom}, y_{bottom}) = (\max(x) \in ROI, \min(y) \in ROI)$$

V.6 3D Template Modeling for Eye, Eyebrows

For each one of the facial features that we want to analyze (eyes, eyebrows and mouth) the adaptation of the motion models is done after having developed the plane parallel to the image plane of the observation model.

The chosen plane is the basis of the new adapted template parameters. Originally, the parameters where determined based upon a local reference system related to the ROI of the feature itself (and the image frame in general). The first step is to relocate the ROI coordinates on the surface approximation for the specific surface and then determine their 3D coordinates on the plane $(x_n^{ROI}, y_n^{ROI}, M)$. x_n^{ROI} and y_n^{ROI} are obtained from the projection of the 3D-ROI points selected from the head model surface. Any template analysis parameters related to the image processing involved (it also includes the ROI), or any other anatomical motion restrictions, are attached to the physiognomy of the person and described by the 3D coordinates that relates them to reference system of the observation model.

During the analysis, not only the coordinates of the ROI are projected but also the coordinates of all those parameters that determine the image analysis and that are related to the feature anatomical structure. The points that define those parameters are obtained from the head surface model and also orthogonally projected onto the template plane.

This section presents the surface approximation designed for the implementation of our algorithmic extension to 3D. Section V.8 will discuss how other surface designs are possible and what are the implications of using non-linear surfaces.

V.6.1 Eye

We model the eye as the sphere — $(x_n - x_0)^2 + (y_n - y_0)^2 + (z_n - z_0)^2 = Rad^2$ — that better suits the eye on the 3D head neutral representation (see Figure V-5). We have chosen the pupil in its neutral position (frontal) as the point around which we will develop the linear approximation. This choice implies developing the plane tangent to the sphere at the pupil point (see Figure V-5 for a graphical representation). A plane tangent to the sphere is such that its normal is vector $\mathbf{r}$, which is given by:

$$\mathbf{r} = (A, B, C) = (x_n - x_0, y_n - y_0, z_n - z_0)$$

where (x_n, y_n, z_n) are the expressions of the coordinates in the sphere general equation. The group of parallel planes tangent to the sphere can be described by this normal as $Ax + By + Cz + D = 0$. From this family we take those that are parallel to the image plane that is, with the form $z_n = M$, which generates a family of circles:

$$f(x_n, y_n) = (x_n - x_0)^2 + (y_n - y_0)^2 = Rad^2 - (M - z_0)^2$$

from this family we are interested in those whose radius is equal to 0 because they define the points of the plane that are tangent to the sphere.

$$f(x_n, y_n) = 0 : M = z_0 + Rad \ \ and \ M = z_0 - Rad$$

The final surface is the plane nearest to the camera optical center:

$$z_n = z_0 + Rad$$

Regarding the adaptation of the image-processing algorithms involved, we know that all the measurements needed for the analysis of the eye area in the search of the minimum point of energy were already dependent on the ROI dimensions. Since the ROI is reshaped after its projection on the image plane, the parameters are automatically adapted as well.

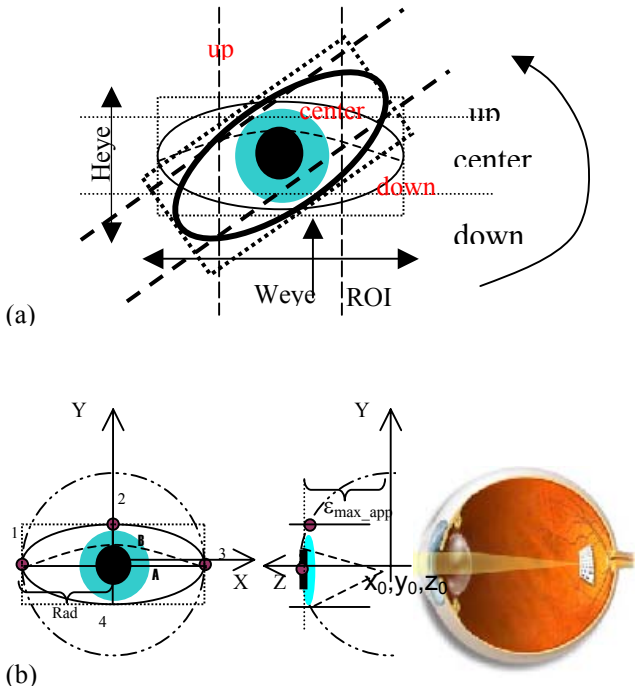


Figure V-5. The eye ROI on the image must follow and reshape according to the view that we have of the feature for a given head pose. Figure (a) schematically shows how the originally designed eye state analysis algorithm cannot be applied directly on the eye as soon as there exist a rigid motion component of the final movement. In Figure (b) the eye model and its linear surface approximation are presented.

V.6.2 Eyebrow

To generate the surface, $f(x_n, y_n, z_n)$, that covers the area of the eyebrow, we are interested in developing the surface tangent to the vertical movement of the eyebrow arch. One of the surface characteristics is that $\frac{\partial f(x_n, y_n, z_n)}{\partial y_n} = 0$, the other one, is that it must follow the shape of the individual forehead (see Figure V-6). The procedure to find the most suitable plane approximation for the development of the template starts by determining the 3D ROI coordinates on that plane, $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ and $\mathbf{x}_4$; then, we take $z_n = M = \frac{1}{4}(z_1 + z_2 + z_3 + z_4)$.

The eyebrow image-processing algorithm also needs to determine the point of eyebrow density change. We obtain points **a** and **b** by projecting orthogonally the actual coordinates of the changing point on the plane. The determination of these two points automatically divides the ROI in the two required analysis areas for the binarization.

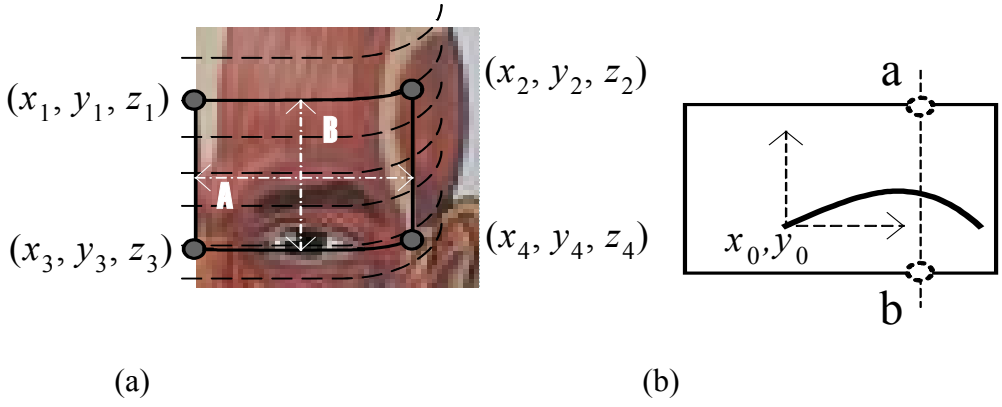


Figure V-6. The eyebrow could be approximated by the surface that tangently follows the eyebrow moment along the forehead. Its plane approximated is the average z value of the points that delimit the eyebrow ROI

V.7 Accuracy of the Adaptation: A Theoretical Study

We study equation system (V-5), to understand the theoretical performance of the adaptation process. First, we are interested in evaluating the position that we obtain on the modeled plane ($\mathbf{x}_n$) for a given point retrieved from the video image ($\mathbf{x}_p$), knowing that the point comes from the analysis of the image of a human head projection and not from its modeling. Second, we want to estimate the degree of dependence on the pose parameters that the complete analysis has.

Let us develop the solution to the system (V-5) (compacted as $\mathbf{A}\cdot\mathbf{x}_n = \mathbf{B}$):

$$x_n = \frac{1}{\det(\mathbf{A})}\cdot\left[M\cdot B\cdot(s_\alpha s_\beta c_\gamma + c_\alpha s_\gamma) + M\cdot C\cdot c_\beta c_\gamma + M\cdot F^2\cdot(s_\alpha s_\gamma - c_\alpha s_\beta c_\gamma) + B\cdot t_Z\cdot(c_\alpha s_\beta s_\gamma + s_\alpha c_\gamma) + F^2\cdot t_X\cdot(s_\alpha s_\beta s_\gamma - c_\alpha c_\gamma) - F^2\cdot t_Y c_\beta s_\gamma + C\cdot H\cdot(c_\alpha c_\gamma - s_\alpha s_\beta s_\gamma) - C\cdot t_Y\cdot(c_\alpha s_\beta s_\gamma + s_\alpha c_\gamma) + B\cdot H\cdot c_\beta s_\gamma\right]$$

$$y_n = \frac{1}{\det(\mathbf{A})}\cdot\left[M\cdot B\cdot(c_\alpha c_\gamma - s_\alpha s_\beta s_\gamma) - M\cdot C\cdot c_\beta c_\gamma + M\cdot F^2\cdot(s_\alpha c_\gamma + c_\alpha s_\beta c_\gamma) + B\cdot t_X\cdot(c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma) - F^2\cdot t_X\cdot(s_\alpha s_\beta c_\gamma + c_\alpha c_\gamma) - F^2\cdot t_Y\cdot c_\beta c_\gamma - C\cdot H\cdot(s_\alpha s_\beta c_\gamma + c_\alpha c_\gamma) + C\cdot t_Y\cdot(s_\alpha s_\gamma - c_\alpha s_\beta c_\gamma) + B\cdot H\cdot c_\beta c_\gamma\right]$$

$$z_n = M$$

where

$$B = -y_p\cdot F\ ;\ C = -x_p\cdot F\ ;\ H = t_Z - F \text{ and } \det(\mathbf{A}) = F\cdot(F\cdot c_\alpha c_\beta - x_p\cdot s_\beta + y_p\cdot s_\alpha c_\beta).$$

It will lead to the development of the expressions analyzed for our accuracy study.

V.7.1 Influence of the surface modeling

Stability of the inversion

Determining when $\det(\mathbf{A}) = 0$, we can evaluate in which circumstances the method is unstable because the inversion does not have a solution:

(V-6) $$F\cdot(F\cdot c_\alpha c_\beta - x_p\cdot s_\beta + y_p\cdot s_\alpha c_\beta) = 0.$$

Analyzing the geometrical nature of system (V-5) (see Figure V-2) we restrict the study to the case where F is a positive number and vector $\vec{\mathbf{r}} = \mathbf{x}_p\big|_{3D} - (0,0,F) = (x_p, y_p, -F)$ (Figure V-7) is the ray of solutions of the projection

inversion. The expression (V-6) represents the family of planes that are parallel to this ray. The combination of rigid motion parameters that generates such a plane leads to the singular case where the system will not be able to deduce the pose.

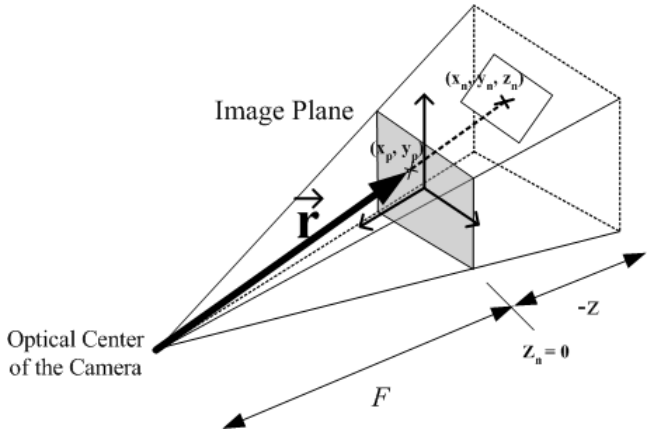


Figure V-7. There exists a solution to the inversion as long as the plane that approximates the feature surface does not take, after the rigid motion of the head has been applied, an orientation parallel to vector $\vec{\mathbf{r}}$

Geometrical interpretation of the analysis:

1. The stability of the system is independent of the angle γ, which is the angle that indicates the rotation around Z-axis. This is because in our observation model the Z-axis and the camera optical axis coincide.
2. The system is unstable for those combinations of angles α and β that transform the plane parallel to the image plane to a plane that contains vector $\vec{\mathbf{r}}$.
3. Some illustrative examples of unstable results:
   - If $x_p = 0$ & $y_p = 0$ (retrieved data along the optical axis) and $\alpha = \pi/2$ & $\beta = \pi/2$ then $\det(\mathbf{A}) = 0$
   - If $x_p = 0$ and $\frac{y_p}{F} = -\tan(\alpha)$ then $\det(\mathbf{A}) = 0$
   - If $y_p = 0$ and $\frac{x_p}{F} = \cos(\alpha) \cdot \tan(\beta)$ then $\det(\mathbf{A}) = 0$
4. As the pose of the plane approaches the conditions under which the system becomes unstable, the projected image of the feature being analyzed starts to concentrate towards the same point. Although mathematically the plane is an infinite surface, we are only interested on that part of the plane containing the feature template. Therefore, the instability of the system can also be controlled from the analysis of the ROI projection; as its area decreases, the data on the

template plane starts concentrating until it reaches the unstable point. Figure V-4 also illustrates the system stability dependence on the estimated pose parameters.

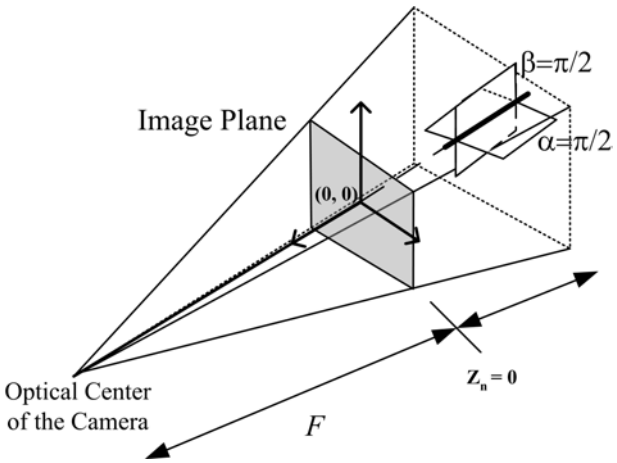


Figure V-8. With $\alpha = \pi / 2$ or $\beta = \pi / 2$, the plane, as it is located in the presented example, generates an undetermined solution that does not permit the inversion of the system around the observed point

Linear approximation accuracy

There exists a precision error due to the model linearization. This inaccuracy is known and limited:

$$\left|\varepsilon_{\max}\right| < \arg\max(M - z_i) \text{ where } i : \text{point} \in \{\textit{feature surface}\}$$

For instance, for the proposed eye feature modeling:

$$\left|\varepsilon_{\max\_eye}\right| < Rad$$

V.7.2 Error propagation

The inaccuracy of the obtained $\mathbf{x}_n$ comes from two sources:

(a) the image processing is not precise and does not retrieve the correct projected location; or

(b) the pose parameters that describe the rigid motion are not accurate enough.

To study the propagation of these two sources of error, we evaluate the real value $\tilde{\mathbf{x}}_n$ obtained as the approximation of the theoretical value $\mathbf{x}_n$ under the influence errors that we have separated into a multiplicative error and an additive error:

$$\tilde{\mathbf{x}}_n = \mathbf{x}_n \cdot \boldsymbol{\varepsilon}_{mult} + \boldsymbol{\varepsilon}_{add} \ .$$

We perform our study by developing the error expressions of each source acting independently. To interpret the error expressions we also provide a numerical study for a specific case.

Influence of the analyzed data precision

Error expressions:

We express the retrieved data as a function of the error due to the inaccuracy during the image processing as:

$$\boxed{\tilde{x}_p = x_p + \varepsilon_x \text{ and } \tilde{y}_p = y_p + \varepsilon_y}.$$

Expressions of the obtained interpreted data as a function of previous errors:

- $\tilde{x}_n = x_n.\varepsilon_{x\_mult} + \varepsilon_{x\_add}$

where

$$\varepsilon_{x\_mult} = \frac{1}{1 + \dfrac{-\varepsilon_x \cdot s_\beta + \varepsilon_y \cdot s_\alpha c_\beta}{-x_p \cdot s_\beta + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta}}$$

$$\varepsilon_{x\_add} = \frac{\dfrac{-\varepsilon_y((M\cdot(s_\alpha s_\beta c_\gamma + c_\alpha s_\gamma) + t_x \cdot (c_\alpha s_\beta s_\gamma + s_\alpha c_\gamma) + H \cdot c_\beta s_\gamma) - \varepsilon_x(H\cdot(c_\alpha c_\gamma - s_\alpha s_\beta s_\gamma) - M\cdot c_\beta c_\gamma - t_y \cdot (s_\alpha c_\gamma + c_\alpha s_\beta s_\gamma))}{-x_p s_\beta + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta}}{1 + \dfrac{-\varepsilon_x s_\beta + \varepsilon_y s_\alpha c_\beta}{-x_p \cdot s_\beta + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta}}$$

- $\tilde{y}_n = y_n.\varepsilon_{y\_mult} + \varepsilon_{y\_add}$

where

$$\varepsilon_{y\_mult} = \frac{1}{1 + \dfrac{-\varepsilon_x \cdot s_\beta + \varepsilon_y \cdot s_\alpha c_\beta}{-x_p \cdot s_\beta + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta}}$$

$$\varepsilon_{y\_add} = \frac{\dfrac{-\varepsilon_y((M\cdot(c_\alpha c_\gamma - s_\alpha s_\beta s_\gamma) + t_x \cdot (c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma) + H \cdot c_\beta c_\gamma) - \varepsilon_x(H\cdot(c_\alpha s_\gamma - s_\alpha s_\beta c_\gamma) + M\cdot c_\beta c_\gamma + t_y \cdot (s_\alpha s_\gamma - c_\alpha s_\beta c_\gamma))}{-x_p s_\beta - F \cdot c_\alpha c_\beta + y_p s_\alpha c_\beta}}{1 + \dfrac{-\varepsilon_x \cdot s_\beta + \varepsilon_y \cdot s_\alpha c_\beta}{-x_p \cdot s_\beta + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta}}$$

Influence of the pose estimation

Error expressions:

Expressions of the obtained interpreted data as a function of the inaccuracy on the estimation of the rigid motion parameters:

$$\boxed{\widetilde{\alpha} = \alpha + \varepsilon_\alpha}$$

- $\widetilde{x}_n = x_n \cdot \varepsilon_{\alpha\_x\_mult} + \varepsilon_{\alpha\_x\_add}$

$$\varepsilon_{\alpha\_x\_mult} = \frac{1}{1 - \dfrac{F \cdot s_\alpha c_\beta - y_p \cdot c_\alpha c_\beta}{-x_p + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta} \cdot t_{\varepsilon_\alpha} + \dfrac{(1 - c_{\varepsilon_\alpha})}{(c_{\varepsilon_\alpha})} \cdot \dfrac{-x_p s_\beta}{-x_p + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta}}$$

$$\varepsilon_{\alpha\_x\_add} = \frac{\dfrac{1}{F \cdot (-x_p + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta)} \cdot \left[\dfrac{(1 - c_{\varepsilon_\alpha})}{c_{\varepsilon_\alpha}} \cdot (M \cdot C \cdot c_\beta c_\gamma - F^2 \cdot t_Y c_\beta s_\gamma + B \cdot H \cdot c_\beta s_\gamma) + T \cdot t_{\varepsilon_\alpha} \right]}{1 - \dfrac{F \cdot s_\alpha c_\beta - y_p \cdot c_\alpha c_\beta}{-x_p + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta} \cdot t_{\varepsilon_\alpha} + \dfrac{(1 - c_{\varepsilon_\alpha})}{(c_{\varepsilon_\alpha})} \cdot \dfrac{-x_p \cdot s_\beta}{-x_p + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta}}$$

where

$$T = [M \cdot B \cdot (c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma) + M \cdot F^2 \cdot (s_\alpha s_\beta c_\gamma + c_\alpha s_\gamma) + B \cdot t_X \cdot (-s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma) +$$
$$F^2 \cdot t_X (c_\alpha s_\beta s_\gamma + s_\alpha c_\gamma) - C \cdot H \cdot (c_\alpha s_\beta s_\gamma + s_\alpha c_\gamma) + C \cdot t_Y (s_\alpha s_\beta s_\gamma - c_\alpha c_\gamma)]$$

- $\widetilde{y}_n = y_p \cdot \varepsilon_{\alpha\_y\_mult} + \varepsilon_{\alpha\_y\_add}$

$$\varepsilon_{\alpha\_y\_mult} = \frac{1}{1 - \dfrac{F \cdot s_\alpha c_\beta - y_p \cdot c_\alpha c_\beta}{-x_p + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta} \cdot t_{\varepsilon_\alpha} + \dfrac{(1 - c_{\varepsilon_\alpha})}{(c_{\varepsilon_\alpha})} \cdot \dfrac{-x_p s_\beta}{-x_p + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta}}$$

$$\varepsilon_{\alpha\_y\_add} = \frac{\dfrac{1}{F \cdot (-x_p + F \cdot c_\alpha c_\beta + y_p s_\alpha c_\beta)} \cdot \left[\dfrac{(1 - c_{\varepsilon_\alpha})}{c_{\varepsilon_\alpha}} \cdot (-M \cdot C \cdot c_\beta c_\gamma - F^2 \cdot t_Y \cdot c_\beta c_\gamma + B \cdot H \cdot c_\beta c_\gamma) + T \cdot t_{\varepsilon_\alpha} \right]}{1 - \dfrac{F \cdot s_\alpha c_\beta - y_p c_\alpha c_\beta}{-x_p + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta} \cdot t_{\varepsilon_\alpha} + \dfrac{(1 - c_{\varepsilon_\alpha})}{(c_{\varepsilon_\alpha})} \cdot \dfrac{-x_p \cdot s_\beta}{-x_p + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta}}$$

where

$$T = [M \cdot B \cdot (-c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma) - M \cdot F^2 \cdot (s_\alpha s_\beta s_\gamma - c_\alpha c_\gamma) - B \cdot t_X (s_\alpha s_\beta c_\gamma + c_\alpha s_\gamma) -$$
$$F^2 \cdot t_X (c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma) - C \cdot H \cdot (c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma) + C \cdot t_Y (s_\alpha s_\beta c_\gamma + c_\alpha s_\gamma)]$$

$$\boxed{\widetilde{\beta} = \beta + \varepsilon_\beta}$$

- $\widetilde{x}_n = x_n \cdot \varepsilon_{\beta\_x\_mult} + \varepsilon_{\beta\_x\_add}$

$$\varepsilon_{\beta\_x\_mult} = \frac{1}{1 - \frac{-x_p \cdot c_\beta + F \cdot c_\alpha s_\beta - y_p \cdot s_\alpha s_\beta}{-x_p + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta} \cdot t_{\varepsilon_\beta}}$$

$$\varepsilon_{\beta\_x\_add} = \frac{\frac{1}{F(-x_p + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta)} \cdot \left[\frac{(1 - c_{\varepsilon_\beta})}{c_{\varepsilon_\beta}} \cdot (M \cdot B \cdot c_\alpha s_\gamma + M \cdot F^2 \cdot s_\alpha s_\gamma + (C \cdot H - F^2 \cdot t_X) \cdot c_\alpha c_\gamma + (B \cdot t_X - C \cdot t_Y) \cdot s_\alpha c_\gamma) + T \cdot t_{\varepsilon_\beta} \right]}{1 - \frac{-x_p \cdot c_\beta - F \cdot c_\alpha s_\beta - y_p \cdot s_\alpha s_\beta}{-x_p + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta} \cdot t_{\varepsilon_\beta}}$$

where

$$T = M \cdot B \cdot s_\alpha c_\beta c_\gamma - M \cdot C \cdot s_\beta c_\gamma - M \cdot F^2 \cdot c_\alpha c_\beta c_\gamma + (F^2 \cdot t_X - C \cdot H) \cdot s_\alpha c_\beta s_\gamma +$$
$$F^2 \cdot t_Y \cdot s_\beta s_\gamma + (B \cdot t_X - C \cdot t_Y) \cdot c_\alpha c_\beta s_\gamma - B \cdot H \cdot s_\beta s_\gamma$$

- $\widetilde{y}_n = y_n \cdot \varepsilon_{\beta\_y\_mult} + \varepsilon_{\beta\_y\_add}$

$$\varepsilon_{\beta\_y\_mult} = \frac{1}{1 - \frac{-x_p \cdot c_\beta - F \cdot c_\alpha s_\beta - y_p \cdot s_\alpha s_\beta}{-x_p + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta} \cdot t_{\varepsilon_\beta}}$$

$$\varepsilon_{\beta\_y\_add} = \frac{\frac{1}{F \cdot (-x_p + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta)} \cdot \left[\frac{(1 - c_{\varepsilon_\beta})}{c_{\varepsilon_\beta}} \cdot (M \cdot B \cdot c_\alpha c_\gamma + M \cdot F^2 \cdot s_\alpha c_\gamma + (C \cdot H - F^2 \cdot t_X) \cdot c_\alpha s_\gamma - (B \cdot t_X - C \cdot t_Y) \cdot s_\alpha c_\gamma) + T \cdot t_{\varepsilon_\beta} \right]}{1 - \frac{-x_p \cdot c_\beta - F \cdot c_\alpha s_\beta - y_p \cdot s_\alpha s_\beta}{-x_p + F \cdot c_\alpha c_\beta - y_p \cdot s_\alpha c_\beta} \cdot t_{\varepsilon_\beta}}$$

where

$$T = -M \cdot B \cdot s_\alpha c_\beta s_\gamma + M \cdot C \cdot s_\beta c_\gamma + M \cdot F^2 \cdot c_\alpha c_\beta s_\gamma - (F^2 \cdot t_X + C \cdot H) \cdot s_\alpha c_\beta c_\gamma +$$
$$F^2 \cdot t_Y \cdot s_\beta c_\gamma + (B \cdot t_X - C \cdot t_Y) \cdot c_\alpha c_\beta c_\gamma - B \cdot H \cdot s_\beta c_\gamma$$

$$\widetilde{\gamma} = \gamma + \varepsilon_{\gamma}$$

- $\widetilde{x}_n = x_n \cdot \varepsilon_{\gamma\_x\_mult} + \varepsilon_{\gamma\_x\_add}$

$$\varepsilon_{\gamma\_x\_mult} = c_{\varepsilon\gamma}$$

$$\varepsilon_{\gamma\_x\_add} = \frac{s_{\varepsilon_\gamma}}{F\cdot(-x_p + F\cdot c_\alpha c_\beta + y_p\cdot s_\alpha c_\beta)}\cdot[M\cdot B\cdot(c_\alpha c_\gamma - s_\alpha s_\beta s_\gamma) - M\cdot C\cdot c_\beta s_\gamma + M\cdot F^2\cdot(s_\alpha c_\gamma + c_\alpha s_\beta s_\gamma) + B\cdot t_X(c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma) +$$
$$F^2\cdot t_X\cdot(s_\alpha s_\beta c_\gamma + c_\alpha s_\gamma) - F^2\cdot t_Y\cdot c_\beta c_\gamma - C\cdot H\cdot(s_\alpha s_\beta c_\gamma + c_\alpha s_\gamma) - C\cdot t_Y\cdot(c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma) + B\cdot H\cdot c_\beta c_\gamma$$

- $\widetilde{y}_n = y_n \cdot \varepsilon_{\gamma\_y\_mult} + \varepsilon_{\gamma\_y\_add}$

$$\varepsilon_{\gamma\_y\_mult} = c_{\varepsilon\gamma}$$

$$\varepsilon_{\gamma\_y\_add} = \frac{s_{\varepsilon_\gamma}}{F\cdot(-x_p + F\cdot c_\alpha c_\beta + y_p\cdot s_\alpha c_\beta)}\cdot[-M\cdot B\cdot(c_\alpha s_\gamma + s_\alpha s_\beta c_\gamma) + M\cdot C\cdot c_\beta s_\gamma - M\cdot F^2\cdot(s_\alpha s_\gamma - c_\alpha s_\beta c_\gamma) - B\cdot t_X(c_\alpha s_\beta s_\gamma + s_\alpha c_\gamma) +$$
$$F^2\cdot t_X(s_\alpha s_\beta s_\gamma - c_\alpha c_\gamma) + F^2\cdot t_Y\cdot c_\beta s_\gamma + C\cdot H\cdot(s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma) + C\cdot t_Y\cdot(c_\alpha s_\beta s_\gamma + s_\alpha c_\gamma) - B\cdot H\cdot c_\beta s_\gamma$$

$$\widetilde{t}_x = t_x + \varepsilon_{t_x}$$

- $\widetilde{x}_n = x_n + \varepsilon_{tx\_x\_add}$

$$\varepsilon_{tx\_x\_add} = \varepsilon_{t_x}\cdot\frac{F^2\cdot(s_\alpha s_\beta s_\gamma - c_\alpha c_\gamma) + B\cdot(c_\alpha s_\beta s_\gamma + s_\alpha c_\gamma)}{F\cdot(-x_p + F\cdot c_\alpha c_\beta + y_p\cdot s_\alpha c_\beta)}$$

- $\widetilde{y}_n = y_n + \varepsilon_{tx\_y\_add}$

$$\varepsilon_{tx\_y\_add} = \varepsilon_{t_x}\cdot\frac{-F^2\cdot(s_\alpha s_\beta s_\gamma - c_\alpha c_\gamma) + B\cdot(c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma)}{F(-x_p + F\cdot c_\alpha c_\beta + y_p\cdot s_\alpha c_\beta)}$$

From the estimated:

$$\widetilde{t}_y = t_y + \varepsilon_{t_x}$$

- $\widetilde{x}_n = x_n + \varepsilon_{ty\_x\_add}$

$$\varepsilon_{ty\_x\_add} = -\varepsilon_{t_y} \cdot \frac{F^2 \cdot c_\beta s_\gamma + C \cdot (c_\alpha s_\beta s_\gamma + s_\alpha c_\gamma)}{F(-x_p + F \cdot c_\alpha c_\beta + y_p s_\alpha c_\beta)}$$

- $\widetilde{y}_n = y_n + \varepsilon_{ty\_y\_add}$

$$\varepsilon_{ty\_y\_add} = -\varepsilon_{t_y} \cdot \frac{-F^2 \cdot c_\beta c_\gamma - C \cdot (c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma)}{F \cdot (-x_p + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta)}$$

$$\widetilde{t}_z = t_z + \varepsilon_{t_z}$$

- $\widetilde{x}_n = x_n + \varepsilon_{t_z\_x\_add}$

$$\varepsilon_{t_z\_x\_add} = -\varepsilon_{t_z} \cdot \frac{B \cdot c_\beta c_\gamma - C \cdot (s_\alpha s_\beta s_\gamma - c_\alpha c_\gamma)}{F(-x_p + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta)}$$

- $\widetilde{y}_n = y_n + \varepsilon_{t_z\_y\_add}$

$$\varepsilon_{t_z\_y\_add} = -\varepsilon_{t_z} \cdot \frac{B \cdot c_\beta c_\gamma - C \cdot (s_\alpha s_\beta c_\gamma + c_\alpha s_\gamma)}{F \cdot (-x_p + F \cdot c_\alpha c_\beta + y_p \cdot s_\alpha c_\beta)}$$

Numerical interpretation

Error expressions depend on the specific projection system parameters (F), the pose that the head model shows (α, β, γ, t_X, t_Y, t_Z) and the value of the projected coordinates obtained from the analysis (x_p, y_p).

The following tables illustrate the error evolution when the analyzed head is almost in neutral position ($\alpha = \beta = \gamma = t_X = t_Y = t_Z \approx 0$), assuming only one source of error at a time. Additive and multiplying error behavior evolves with the changes in the additive error of the analyzed results.

Table V-1 and Table V-2 show the error evolution in terms of percentage of error committed when analyzing the projected data (x_p, y_p). Table V-6, Table V-7 and Table V-8 present the percentage of error from the translation parameter estimation; and Table V-3, Table V-4 and Table V-5 show the error in radian units due to estimation inaccuracy over the pose angles.

$f(), f'(), g()$ and $g'()$ are different functions that depend on the specific projection system parameters. $h(), h'(), h''(), h'''(), k(), k'(), k''(), k'''(), m(), m'(), n(), n'(), q(), q'(), p()$ and $p'()$ are functions that depend on the system parameters and the analyzed values x_p and y_p.

Table V-1

EVOLUTION OF THE ERROR DUE TO THE INACCURACY ON THE X COMPONENT OF PROJECTED DATA ANALYSIS.

error % x_p	error % x_{mult}	error % x_{add}	error % y_{mult}	error % y_{add}
10	0	10.f(system)	0	10.f'(system)
5	0	5.f(system)	0	5.f'(system)
1	0	f(system)	0	f'(system)
0.5	0	0.5.f(system)	0	0.53.f'(system)
0.1	0	0.1.f(system)	0	0.1.f'(system)
0.05	0	0.05.f(system)	0	0.05.f'(system)
0.01	0	0.01.f(system)	0	0.01.f'(system)

Table V-2

EVOLUTION OF THE ERROR DUE TO THE INACCURACY ON THE *Y* COMPONENT OF PROJECTED DATA ANALYSIS.

error % y_p	error % x_{mult}	error % x_{add}	error % y_{mult}	error % y_{add}
10	0	10.g(system)	0	10.g'(system)
5	0	5.g(system)	0	5.g'(system)
1	0	g(system)	0	g'(system)
0.5	0	0.5.g(system)	0	0.5.g'(system)
0.1	0	0.1.g(system)	0	0.1.g'(system)
0.05	0	0.05.g(system)	0	0.05.g'(system)
0.01	0	0.01.g(system)	0	0.01.g'(system)

Table V-3

EVOLUTION OF THE ERROR DUE TO THE INACCURACY ON THE α POSE PARAMETER PRECISION.

error α rad	error x_{mult}	error x_{add}
1	1/(1-*h*(sys,coor). 0.0175)	(0.000152.*h*'(sys,coor)+0.0175.*h*"(sys,coor))/ (1-*h*(sys,coor). 0.0175)
0.5	1/(1-*h*(sys,coor).0.00873)	(0.0000381.*h*'(sys,coor)+0.00873.*h*"(sys,coor))/ (1-*h*(sys,coor).0.00873)
0.1	1/(1-*h*(sys,coor).0.00175)	(0.00000152.*h*'(sys,coor)+0.00175.*h*"(sys,coor))/ (1-*h*(sys,coor).0.00175)
0.05	1/(1-h(sys,coor).0.000873)	(0.000000381.*h*'(sys,coor)+0.000873.*h*"(sys,coor))/ (1-*h*(sys,coor).0.000873)
0.01	1/(1-h(sys,coor).0.000175)	(0.0000000152.*h*'(sys,coor)+0.000175.*h*"(sys,coor))/ (1-*h*(sys,coor).0.000175)
error α rad	error y_{mult}	error y_{add}
1	1/(1-*h*(sys,coor). 0.0175)	(0.000152.*h*'"(sys,coor)+0.0175.*h*"(sys,coor))/ (1-*h*(sys,coor).0.0175)
0.5	1/(1-*h*(sys,coor).0.00873)	(0.0000381.*h*'"(sys,coor)+0.00873.*h*"(sys,coor))/ (1-*h*(sys,coor).0.00873)
0.1	1/(1-*h*(sys,coor).0.00175)	(0.00000152.*h*'"(sys,coor)+0.00175.*h*"(sys,coor))/ (1-*h*(sys,coor).0.00175)
0.05	1/(1-*h*(sys,coor).0.000873)	(0.000000381.*h*'"(sys,coor)+0.000873.*h*"(sys,coor))/ (1-*h*(sys,coor).0.000873)
0.01	1/(1-*h*(sys,coor).0.000175)	(0.0000000152.*h*'"(sys,coor)+0.000175.*h*"(sys,coor))/ (1-*h*(sys,coor).0.000175)

Table V-4

EVOLUTION OF THE ERROR DUE TO THE INACCURACY ON THE β POSE-PARAMETER PRECISION.

error β rad	error x_{mult}	error x_{add}
1	1/(1-*k*(sys,coor).0.0175)	(0.000152.*k*"(sys,coor)+0.0175.*k*"(sys,coor))/ (1-*k*(sys,coor).0.0175)
0.5	1/(1-*k*(sys,coor).0.00873)	(0.0000381.*k*'(sys,coor)+0.00873.*k*"(sys,coor))/ (1-*k*(sys,coor).0.00873)
0.1	1/(1-*k*(sys,coor).0.00175)	(0.00000152.*k*'(sys,coor)+0.00175.*k*"(sys,coor))/ (1-*k*(sys,coor).0.00175)
0.05	1/(1-*k*(sys,coor).0.000873)	(0.000000381.*k*'(sys,coor)+0.000873.*k*"(sys,coor))/ (1-*k*(sys,coor).0.000873)
0.01	1/(1-*k*(sys,coor).0.000175)	(0.0000000152.*k*'(sys,coor)+0.000175.*k*"(sys,coor))/ (1-*k*(sys,coor).0.000175)
error β rad	error y_{mult}	error y_{add}
1	1/(1-*k*(sys,coor).0.0175)	(0.000152.*k*'"(sys,coor)+0.0175.k"(sys,coor))/ (1-*k*(sys,coor).0.0175)
0.5	1/(1-*k*(sys,coor).0.00873)	(0.0000381.*k*'"(sys,coor)+0.00873.*k*"(sys,coor))/ (1-*k*(sys,coor).0.00873)
0.1	1/(1-*k*(sys,coor).0.00175)	(0.00000152.*k*'"(sys,coor)+0.00175.*k*"(sys,coor))/ (1-*k*(sys,coor).0.00175)
0.05	1/(1-*k*(sys,coor).0.000873)	(0.000000381.*k*'"(sys,coor)+0.000873.*k*"(sys,coor))/ (1-*k*(sys,coor).0.000873)
0.01	1/(1-*k*(sys,coor).0.000175)	(0.0000000152.*k*'"(sys,coor)+0.000175.*k*"(sys,coor))/ (1-*k*(sys,coor).0.000175)

Table V-5

EVOLUTION OF THE ERROR DUE TO THE INACCURACY ON THE γ POSE-PARAMETER PRECISION.

error γ rad	error x_{mult}	error x_{add}	error y_{mult}	error y_{add}
1	0.99984	0.0175.*m*(sys,coor)	0.0175	0.99984.*m*'(sys,coor)
0.5	0.999961	0.00873.*m*(sys,coor)	0.00873	0.999961.*m*'(sys,coor)
0.1	0.9999984	0.00175.*m*(sys,coor)	0.00175	0.9999984.*m*'(sys,coor)
0.05	0.99999961	0.000873.*m*(sys,coor)	0.000873	0.99999961.*m*'(sys,coor)
0.01	0.999999984	0.000175.*m*(sys,coor)	0.000175	0.999999984.*m*'(sys,coor)

Table V-6

EVOLUTION OF THE ERROR DUE TO THE INACCURACY ON THE T_X POSE-PARAMETER PRECISION.

error % t_x	error % x_{mult}	error % x_{add}	error % y_{mult}	error % y_{add}
10	none	10.*n*(system,coor)	none	10.*n*'(system,coor)
5	none	5.*n*(system,coor)	none	5.*n*'(system,coor)
1	none	*n*(system,coor)	none	*n*'(system,coor)
0.5	none	0.5.*n*(system,coor)	none	0.5.*n*'(system,coor)
0.1	none	0.1.*n*(system,coor)	none	0.1.*n*'(system,coor)
0.05	none	0.05.*n*(system,coor)	none	0.05.*n*'(system,coor)
0.01	none	0.01.*n*(system,coor)	none	0.01.*n*'(system,coor)

Table V-7

EVOLUTION OF THE ERROR DUE TO THE INACCURACY ON THE T_Y POSE-PARAMETER PRECISION.

error % t_y	error % x_{mult}	error % x_{add}	error % y_{mult}	error % y_{add}
10	none	10.*p*(system,coor)	none	10.*p*'(system,coor)
5	none	5.*p*(system,coor)	none	5.*p*'(system,coor)
1	none	*p*(system,coor)	none	*p*'(system,coor)
0.5	none	0.5.*p*(system,coor)	none	0.5.*p*'(system,coor)
0.1	none	0.1.*p*(system,coor)	none	0.1.*p*'(system,coor)
0.05	none	0.05.*p*(system,coor)	none	0.05.*p*'(system,coor)
0.01	none	0.01.*p*(system,coor)	none	0.01.*p*'(system,coor)

Table V-8

EVOLUTION OF THE ERROR DUE TO THE INACCURACY ON THE T_Z POSE-PARAMETER PRECISION.

error % t_z	error % x_{mult}	error % x_{add}	error % y_{mult}	error % y_{add}
10	none	10.*q*(system,coor)	none	10.*q*'(system,coor)
5	none	5.*q*(system,coor)	none	5.*q*'(system,coor)
1	none	*q*(system,coor)	none	*q*'(system,coor)
0.5	none	0.5.*q*(system,coor)	none	0.5.*q*'(system,coor)
0.1	none	0.1.*q*(system,coor)	none	0.1.*q*'(system,coor)
0.05	none	0.05.*q*(system,coor)	none	0.05.*q*'(system,coor)
0.01	none	0.01.*q*(system,coor)	none	0.01.*q*'(system,coor)

Discussion and conclusion from the complete evaluation

Bearing in mind that the previous tables represent the results on a specific situation, we can still study the influence trends over the error behavior and which parameters seem to be more critical for a correct coupling result. Exact error values depend on the instant system characteristics but specific functions that represent these characteristics have the same order of magnitude ($O(10^0)$), independently of the system itself and the obtained results.

In the evolution study, we see that the instant pose can greatly influence the error behavior. This fact is noticeable, for instance, when we observe that the projected-data multiplying-error disappears when the head is in its neutral position (Tables IV-1 and IV-2). In general, for small inaccuracy errors the overall error behavior shows a clean linear evolution. Pose parameters related to the X and Y-axis (α, β, t_X and t_Y) have similar comportment. Rotations around the X and the Y-axis have stronger action over the error results than the rotation related to the Z-axis. In fixed pose conditions γ rotation, t_X and t_Y do not change the image appearance of the studied ROI and therefore errors due to inaccuracy in their prediction have less impact on obtaining the neutral coordinates.

The pose-parameter estimation proves to be critical for the template adaptation. In addition to directly influencing the interpretation of the data analyzed on the image, it is also directly responsible of the success and accuracy of the image-processing analysis. The motion template analysis techniques implemented depend on the correct delimitation of the ROI of the feature that is going to be analyzed. Since image ROIs are obtained from projections, pose parameters also intervene in determining their final location. This implies that the accuracy ($\varepsilon_x, \varepsilon_y$) of the analyzed values ($\tilde{x}_p$, $\tilde{y}_p$), although apparently not directly dependent on the pose, is indirectly dependent on it because of the determination of the template ROI.

From our theoretical evaluation, we conclude that the template adaptation procedure can only be feasible if the rigid motion study of the head on the image generates the required pose parameters with a minimum degree of accuracy. The level of precision is set so that ROIs are properly tracked and do not become the major source of error for the image-processing analysis techniques involved.

V.8 Using other surfaces for the algorithmic 3D-extension

The motion model extension to 3D space has been designed to utilize a linear approximation of the feature surface because we seek to take advantage of the computational benefits linear surfaces provide. In this section, we would like to discuss how this system could also be implemented using other surface approximations. We can always aspire to build the surface that minimizes the error derived from the imprecision.

Let us recall equation system (V-5) in its extended form:

(V-7)
$$\begin{cases} a_1 \cdot x_n + a_2 \cdot y_n + a_3 \cdot z_n = a_4 \\ b_1 \cdot x_n + b_2 \cdot y_n + b_3 \cdot z_n = b_4 \end{cases}$$

This system represented the projection and the pose transformation inversion needed to recover the motion data obtained from the image plane. The parametric solution to the system could be:

$$z_n(y_n) = \frac{A + B \cdot y_n}{C}$$

$$x_n(y_n) = \frac{a_4 - a_2 \cdot y_n - a_3 \cdot z_n}{a_1}$$

with $A = a_4 \cdot b_1 - b_4 \cdot a_1$; $B = a_1 \cdot b_2 - b_1 \cdot a_2$ and $C = a_3 \cdot b_1 - b_3 \cdot a_1$.

Let us express the surface that represents the feature ROI in one of its parametric forms:

$$x_n(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} dx_{i,j} \cdot N_i^k(u)(v) \cdot N_j^l(u)(v)$$

$$y_n(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} dy_{i,j} \cdot N_i^k(u)(v) \cdot N_j^l(u)(v)$$

$$z_n(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} dz_{i,j} \cdot N_i^k(u)(v) \cdot N_j^l(u)(v).$$

Where $N_i^k(u)(v)$ and $N_j^l(u)(v)$ are B-Spline values (NURBS) for the chosen surface control points and $dx_{i,j}$, $dy_{i,j}$ and $dz_{i,j}$ the weights given to those values.

We would like to add the surface constraint to system (V-7) to solve for x_n, y_n and z_n. We do so by describing $z_n(u,v)$ as

$$z_n(y(u,v)) = \frac{A}{C} + \frac{A}{C} \cdot \sum_{i=o}^{n} \sum_{j=0}^{m} dy_{i,j} \cdot N_i^k(u)(v) \cdot N_j^l(u)(v) .$$

Solving the system is the equivalent to finding a solution for u and v in the following expression:

(V-8) $$z_n(u,v) = \frac{A}{C} + \frac{B}{C} \cdot y_n(u,v) .$$

One way to set u and v is by using Newton's numerical method. To do so, the function to optimize

(V-9) $$f(u,v) = z_n(u,v) - \frac{A}{C} + \frac{B}{C} \cdot y_n(u,v) = 0$$

must fit these criteria:

1. It must be C^1, both in u and v.
2. Its Jacobian must be invertible at the point (u_0,v_0) of the expected solution, that is: $\exists J^{-1} f(u_0,v_0) \neq 0$.

These criteria are automatically met if the surface studied is C^2.

Besides the parametric surfaces, other curves could be used: Cubic, Quadratic, etc. B-Splines could be more advantageous because they are already widely used in Computer Graphics to define surfaces and could present interesting side aspects to consider. In all cases, numerical methods, like the one we have suggested, will have to be used to find a proper solution to the system. It will always imply loss in computation speed.

Brief review of Newton's method

Newton's method is a root-finding algorithm which uses the first few terms of the Taylor series of a function $f(x)$ in the vicinity of a suspected root to zero in order to find an approximation to that root. It is also called the Newton-Raphson method. For $f(x)$ a polynomial, Newton's method is essentially the same as Horner's method. The Taylor series of $f(x)$ about the point $x = x_0 + \varepsilon$ is given by

$$f(x_0 + \varepsilon) = f(x_0) + f'(x_0) \cdot \varepsilon + \tfrac{1}{2} f''(x_0) \cdot \varepsilon^2 + \ldots$$

Keeping terms only to first order,

(V-10) $$f(x_0+\varepsilon)\approx f(x_0)+f'(x_0)\cdot\varepsilon\,.$$

This expression can be used to estimate the amount of offset ε needed to land closer to the root starting from an initial x_0. Setting $f(x_0+\varepsilon)=0$ and solving (V-10) for $\varepsilon-\varepsilon_0$ gives

$$\varepsilon_0=-\frac{f(x_0)}{f'(x_0)},$$

which is the first-order adjustment to the root's position. By letting $x_1=x_0+\varepsilon_0$, calculating a new ε_1, and so on, the process can be repeated until it converges to a root using

$$\varepsilon_n=-\frac{f(x_n)}{f'(x_n)}.$$

The method is easily extended to solve for two unknowns:

$$\begin{bmatrix}u\\v\end{bmatrix}=\begin{bmatrix}u_0\\v_0\end{bmatrix}-J^{-1}f(u_0,v_0)\cdot f(u_0,v_0)$$

where, in the case of our surface, we know that

$$J^{-1}f(u_0,v_0)=\left[\begin{array}{c}\dfrac{\partial^{-1}f(u,v)}{\partial u}\\[2ex]\dfrac{\partial^{-1}f(u,v)}{\partial v}\end{array}\right]\Bigg|_{u_0,v_0}=\left[\begin{array}{c}\left(\dfrac{\partial z_n(u,v)}{\partial u}-\dfrac{B}{C}\cdot\dfrac{\partial y_n(u,v)}{\partial u}\right)^{-1}\\[2ex]\left(\dfrac{\partial z_n(u,v)}{\partial v}-\dfrac{B}{C}\cdot\dfrac{\partial y_n(u,v)}{\partial v}\right)^{-1}\end{array}\right]\Bigg|_{u_0,v_0}.$$

VI Technical Evaluation of Coupling Facial Expression Analysis and Head Pose Tracking

This Chapter compiles the practical development and the experimental evaluation of the facial motion analysis system detailed in the previous chapters. We have tried to study the performance, accuracy and robustness of the proposed head-pose & expression analysis coupling. The testbed has been set to correspond as much as possible to the deployment of a real acquisition system for teleconferencing.

VI.1 Introduction

The work compiled in this thesis report has been the natural continuation of the scientific research on facial analysis for synthetic animation that the Image Group of the Multimedia Communications Department at the Institut Eurécom has been doing during the past 6 years.

Regarding rigid facial motion analysis, the research group had developed an algorithm that utilizes a feedback loop inside a Kalman filter to obtain precise information about the person's location in space. Kalman filtering has been applied to head tracking giving good results (Cordea, Petriu, E. M., Georganas, D., Petriu, D. C., & Whalen, T. E., 2001; Ström, 2002) and it enables the prediction of the translation and rotation parameters of the head from the 2D tracking of specific points of the face on the image plane. For non-rigid facial motion analysis, some interesting techniques for expression analysis had already been tested (Valente, 1999) but the PCA-based approach originally taken had too many restrictions when extending its use to any other pose. This led us to develop the pose-expression coupling technique investigated here.

The main practical drawback of our head-tracking system is the need for 3D information about the shape of the head that we are tracking. This implies that we must use a model that provides accurate 3D coordinates of the points whose projection is tracked on the image and fed to the filter to obtain the prediction of the pose parameters. Very often, a general head model is used. The apparent drawback can become a strong advantage if a realistic 3D synthetic representation of the user is available. In (Valente & Dugelay, 2001), we showed that improvement in the amount of freedom of movement in front of the camera is possible if using the speaker's clone during the tracking. Models have to be a precise 3D representation of the speaker, in shape and texture, because our approach compares head models and video frames at the image level.

We have inserted the feature motion analysis algorithms presented in this report inside the original head-tracking framework. Since the speaker's realistic head model is needed for the tracking, and therefore also available during the analysis, the 3D data required for extending the use of the motion analysis templates is obtained from them.

For the adaptation process to be possible, the head-pose tracking algorithm and the image processing must share the same observation model. Some other performing tracking algorithms work using local 2D image reference systems on which they can only estimate the user's position on the screen (e.g. Bradski's Cam Shift algorithm (1998)). For the sequence of processes to follow, i.e., detection of the features to be analyzed,

analysis and interpretation of the analyzed results, this information is not accurate enough.

In our approach, the algorithm operating on the image plane extracts the 2D features to be tracked on the video sequence from the synthesized image of the model, onto which the predicted pose parameters have already been applied. At this point, it provides an adjusted view of the user in its future pose. The system structure also enables to project points belonging to the modeled areas of the face (those used for feature expression analysis) and track the evolution of each feature ROI. Figure VI-1 shows two screen shots where the online adaptation of the 3D model view is observed.

Details about the Laboratory's previous and current research work can be found at the group's website (Video Cloning, 1999).

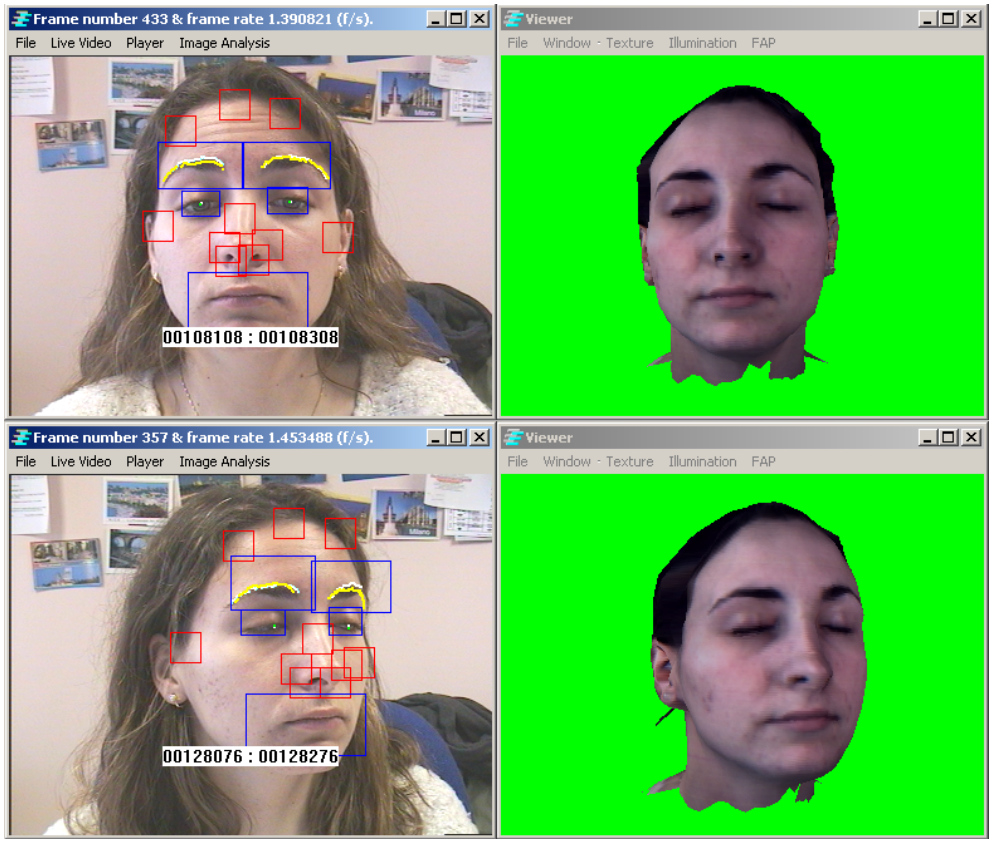


Figure VI-1. Two screen shots of the test settings. On the most left window we present the video input, the projection of the analysis results, and the evolution of the ROIs, suitable for visual inspection of the feature analysis performance; on the most right window the synthetic reproduction (projected using OpenGL) of the user's clone is represented, allowing us to control the evolution of the head tracking algorithm. Since we use a highly realistic model to perform the tracking we utilize its 3D data to do the algorithmic adaptation: we redefine the motion animation models and their ROIs on it

VI.2 Description of the System

The test-bed for the technical evaluation of the proposed system comprises two main environments, the video input window and the synthetic rendering window (Figure VI-1). We consider the following standard acquisition conditions: a small camera situated in front of the speaker on top of the monitor around 75 cm away from him (Figure VI-2).

Figure VI-2. Settings for the technical evaluation: just one computer and one camera are used

The video-input window shows either life input or a video sequence recorded by the camera. It also presents the graphical feedback from the different analysis methods thus allowing us to visually verify the correctness of the expression analysis.

The rendering window displays the synthetic reproduction of the speaker's head model. The model is rendered after the obtained pose parameters have been applied on it. This allows us to check if the tracking algorithm still keeps track of the head correctly.

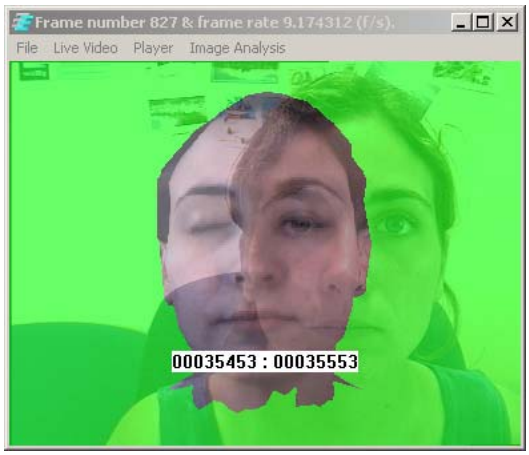
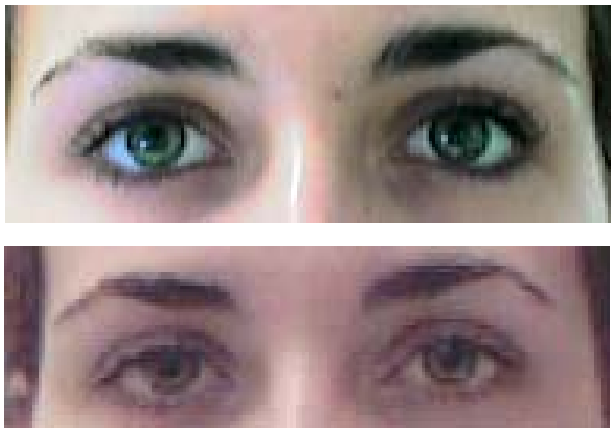


Figure VI-3. Synthetic images and video input are blended to be able to initialize the analysis system

Both windows are the same size because the synthetic world is meant to reproduce exactly the real world. Indeed, this fact is utilized to initialize the complete system. For the first acquired frame the head pose is completely unknown and the filter has not yet been initialized. Therefore, an initialization step is required. During this step, the face features to be tracked in 2D must be detected on the frame and the setting of the pose parameters to the first head pose is required. Since the development of this procedure is not the purpose of the research presented in this thesis report, we decided to manually initialize the system by blending the real video input with the synthetic view of the model. This way, we can ask the speaker to move and set himself in the neutral pose (see Figure VI-3).

VI.2.1 Characteristics of video input

The quality of the video input is very important. Different cameras have different optical characteristics. Our system does not take those characteristics into account to be as environmentally independent as possible. Each system may also use different acquisition equipment and during acquisition, the image can be significatly altered, sometimes resulting in the inclusion of undesired artifacts. As we observe in Figure VI-4, the image resulting from capturing the same object with different equipment may differ highly. The first image (a) has been acquired with a standard acquisition card and a low distortion camera. Color and shape are well maintained. Image (b) has been recorded with a typical web cam. We clearly see that the compression methods needed to lower the network payload during video streaming on the Internet, generally jpeg, damage the image quality.


(a)

(b)

Figure VI-4. Image (a) was recorded with a standard camera and acquisition card. Image (b) was obtained from a typical web camera

The algorithms developed during our research have been tested over video input whose quality falls in category (a). The acquisition system did not distort the final image. Nevertheless, we did not assume certain image quality with respect to color, contrast and any other characteristics.

Characteristics of the complete acquisition system

Camera:	Sony EVI-D31
Acquisition Card:	Osprey 100 Video Capture Device
Input Stream:	PAL-BGHDI
Output Stream:	BGR32 : [B:8 bytes][G:8 bytes][R:8 bytes][transparency:8 bytes]
Max frame rate:	33 f/s
Video Input Size:	384x288 (~) [*w*x*h*]

We used ® DirectX® (2003) technology to deploy video acquisition and rendering. The DirectShow® filter strategy was utilized. One of the system characteristics was that images were recorded by sweeping from down to top, thus inverting the data information in memory. The reference system to access data is illustrated in Figure VI-5. One pixel has the following coordinates (x_{video}, y_{video}).

All these details will be taken into account during the deployment of the algorithms. Comparing the input image to the observation model coordinate system that is used (System V-2) and that has been adapted for the synthetic world (VI-6), we know that:

$$x_p^{OGL} = 2 \cdot \frac{x_{video}}{w} - 1 \qquad y_p^{OGL} = 2 \cdot \frac{y_{video}}{h} - 1$$

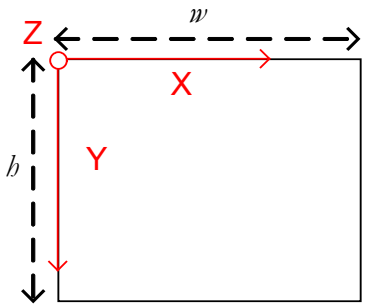


Figure VI-5. Reference system of the video image in memory

VI.2.2 Head model synthesis

Our image analysis techniques need geometrical information about the speaker's head to perform well. This information is obtained from the speaker's 3D model, which can be the same as the model utilized to represent the speaker during animation but not necessarily.

To create the head models we used the data generated by a Cyberware™ 3D scanner (Cyberware, 2003). It provided us with a cloud of points that represented the surface of the person's head. It also retrieved a cylindrical image representation of the surface texture which was automatically mapped onto the 3D points. Then, this dense cloud of points was triangulated to obtain a rich wireframe. The wireframe was used to extract all the precise 3D data needed to redefine the motion models used during the analysis in 3D (Figure VI-6a; Figure VI-6c) . We used a reduced version of the original wireframe (Figure VI-6b) to track the head on the video because we wanted to be able to render both video and synthetic feedback in a reasonable time.

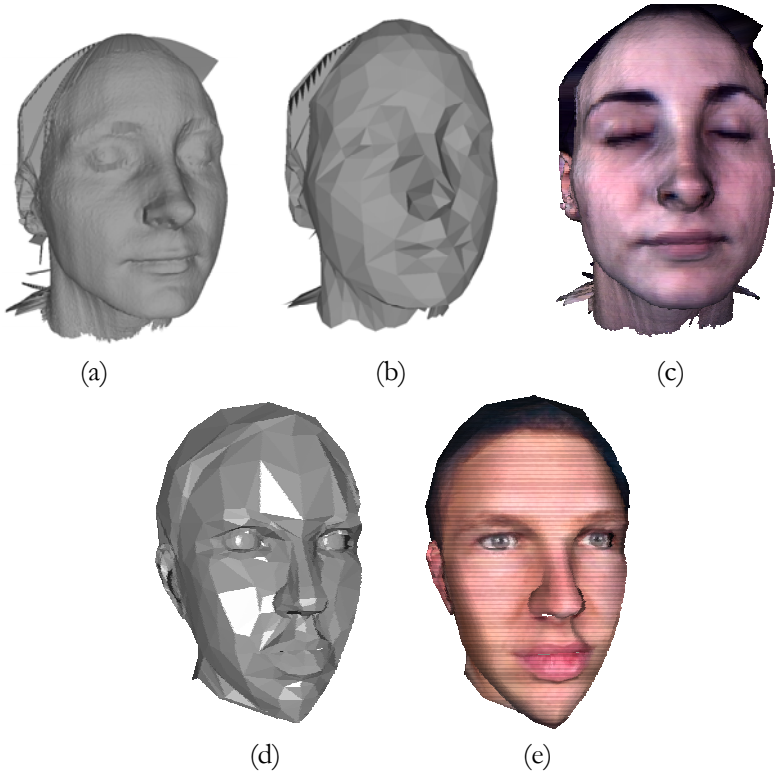


Figure VI-6. Different models were used for the practical implementation of the algorithms. Very dense wireframe models were utilized to extend the use of our expression analysis algorithms (a-c), a less heavy version of these models was rendered during the coupling of head tracking and expression analysis. An animated avatar substituted the realistic head model of the speaker to evaluate the naturalness of the animation synthesis created from the parameters obtained from the analysis

These complete models were used only for the analysis because no animation, besides rigid motion, was given with them. Their visual feedback helped to control head pose tracking. To test the naturalness created from online analysis, we used a simpler model: *Olivier* (Figure VI-6d; Figure VI-6e). This model, which we consider an avatar, was provided by France Telecom R&D – Rennes. Due to its wireframe simplicity it was easy to implement some customized animation. For example, by using *Olivier* we could observe the results from synthesizing eye actions analyzed from video input.

MPEG-4 based head models

Our head models have been generated from different 3D data acquisition methods but they all have been coded in .mp4 format. This is a binary format that includes the geometry, the texture and the animation rules of 3D heads packed following the MPEG-4 standard. We decided to adopt the semantics and syntax that MPEG-4 provides for Facial Animation (and not FACS or proprietary semantics and syntax) because our analysis/synthesis techniques are integrated inside a system that aims at providing telecommunications services, as it is explained in depth in Chapter III.

The representation of synthetic visual objects in MPEG-4 is based on the prior VRML (2003) standard using nodes such as Transform, which defines rotation, scale or translation of an object, and IndexedFaceSet describing 3-D the shape of an object by an indexed face set. However, MPEG-4 is the first international standard that specifies a compressed binary representation of animated synthetic audio-visual objects. Appendix VI-J contains some details related to the different nodes involved; this information has been taken from ISO/IEC 144496-2 MPEG-4 (1999).

Specification and animation of faces

For a complete face object, MPEG-4 specifies a face model in its neutral state, a number of feature points on this neutral face as reference points, and a set of FAPs, each corresponding to a particular facial action deforming a face model in its neutral state. The FAP value for a particular FAP indicates the magnitude of the corresponding action, e.g., a big versus a small smile or deformation of a mouth corner. For an MPEG-4 terminal to interpret the FAP values using its face model, it has to have predefined model specific animation rules to produce the facial action corresponding to each FAP. The terminal can either use its own animation rules or download a face model and its associated face animation tables (FAT) to have a customized animation behavior. Since FAPs are required to animate faces of different sizes and proportions, the FAP values are defined in face animation parameter units (FAPU). The FAPU are computed from spatial distances between major facial features on the model in its neutral state.

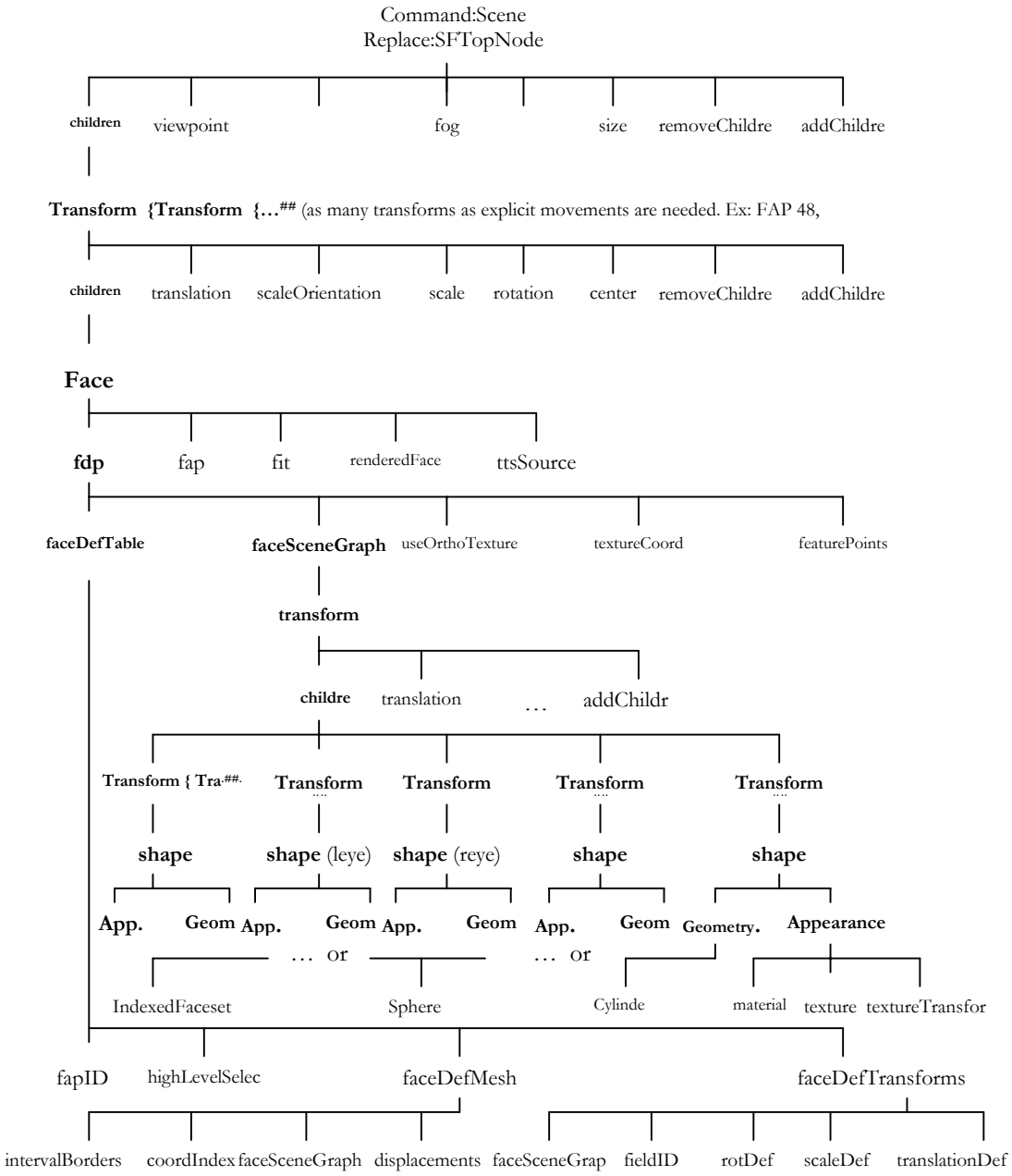


Figure VI-7. Tree structure of the MPEG-4 coded head models

In order to use facial animation in the context of MPEG-4 systems, a BIFS scene graph (ISO/IEC 14496-1 MPEG-4 STD, 1998) has to be transmitted to the decoder. The minimum scene graph contains a Face node and a FAP node. The FAP decoder writes the amplitude of the FAPs into fields of the FAP node. The FAP node might have the children Viseme and Expression which are FAPs requiring a special syntax. This scene graph would enable an encoder to animate the proprietary face model of the decoder. If a face model is to be controlled from a TTS system, an AudioSource node is to be attached to the face node.

In order to download a face model to the decoder, the face node requires an FDP node as one of its children. This FDP node contains the position of the feature points in the downloaded model, the scene graph of the model and the FaceDefTable, FaceDefMesh and FaceDefTransform nodes required to define the action caused by FAPs. The typical structure of the data being coded per each model can be found in Figure VI-7.

Neutral face and Facial Animation Parameter Units

Our model is considered to be in its neutral state (see Figure VI-8) when:

- the coordinate system is right-handed; head axes are parallel to the world axes
- gaze is in direction of Z axis,
- all face muscles are relaxed,
- eyelids are tangent to the iris, the pupil is one third of the diameter of the iris,
- lips are in contact; the line of the lips is horizontal and at the same height of lip corners,
- the mouth is closed and the upper teeth touch the lower ones,
- the tongue is flat, horizontal with the tip of tongue touching the boundary between upper and lower teeth.

An FAPU and the feature points used to derive the FAPUs are defined with respect to the face in its neutral state. The FAPUs allow interpretation of the FAPs on any facial model in a consistent way, producing reasonable results in terms of expression and speech pronunciation. The measurement units are shown in Table VI-1.

Table VI-1

FACIAL ANIMATION PARAMETER UNITS AND THEIR DEFINITIONS.

	Description	FAPU Value
IRISD0 = 3.1.y – 3.3.y = 3.2.y – 3.4.y	Iris diameter (by definition it is equal to the distance between upper ad lower eyelid) in neutral face	IRISD = IRISD0 / 1024
ES0 = 3.5.x – 3.6.x	Eye separation	ES = ES0 / 1024
ENS0 = 3.5.y – 9.15.y	Eye - nose separation	ENS = ENS0 / 1024
MNS0 = 9.15.y – 2.2.y	Mouth - nose separation	MNS = MNS0 / 1024
MW0 = 8.3.x – 8.4.x	Mouth width	MW = MW0 / 1024
AU	Angle Unit	10^{-5} rad

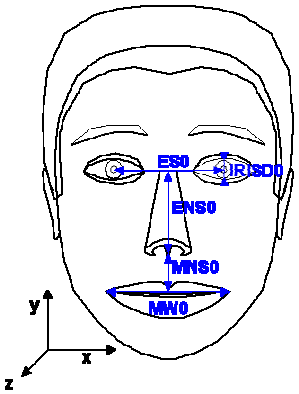


Figure VI-8. A face model in its neutral state and the feature points used to define FAP units (FAPUs). Fractions of distances between the marked key features are used to define FAPU. © MPEG-4

Face Definition Points

MPEG-4 specifies 84 feature points, or Face Definition Points (FDPs) on the neutral face. Feature points are arranged in groups like cheeks, eyes, and mouth. The location of these feature points has to be known for any MPEG-4 compliant face model. The feature points on the model should be located according to Figure VI-9.

The FDPs are normally transmitted once per session, followed by a stream of compressed FAPs. If the decoder does not receive the FDPs, the use of FAPUs ensures that it can still interpret the FAP stream. This insures minimal operation in broadcast or

teleconferencing applications. The FDP set is specified in BIFS syntax. The FDP node defines the face model to be used at the receiver. Two options are supported:

- calibration information is downloaded so that the proprietary face of the receiver can be configured using facial feature points and optionally a 3D mesh or texture;
- a face model is downloaded with the animation definition of the Facial Animation Parameters. This face model replaces the proprietary face model in the receiver.

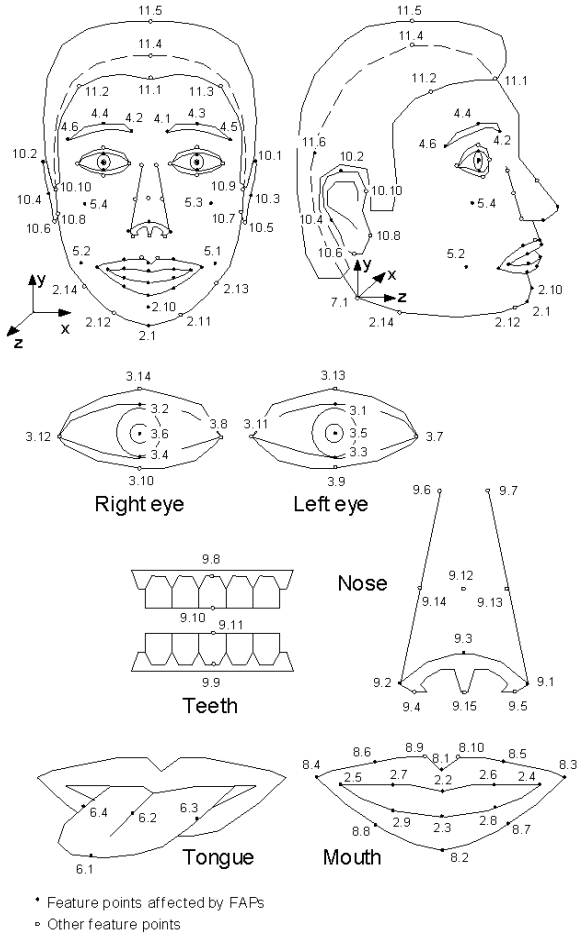


Figure VI-9. At least the Face Definition Points must be specified on the head model wireframe to define it to allow all animation systems to customize their own models. Our models include these points in their mesh, ensuring the correct understanding of MPEG-4 animation parameters. © MPEG-4

Face Animation Parameters

The FAPs are based on the study of minimal perceptible actions and are closely related to muscle actions. The 68 parameters are categorized into 10 groups related to parts of the face. FAPs can also be used to define facial action units. Exaggerated

amplitudes permit the definition of actions that are normally not possible for humans, but are desirable for cartoon-like characters.

The FAP set contains two high level parameters: visemes and expressions. A viseme is a visual correlate to a phoneme. The viseme parameter allows viseme rendering (without having to express them in terms of other parameters) and enhances the result of other parameters, ensuring the correct rendering of visemes. Only static visemes which are clearly distinguished are included in the standard set. Additional visemes may be added in future extensions of the standard. Similarly, the expression parameter allows the definition of high level facial expressions. The facial expression parameter values are defined by textual descriptions. To facilitate facial animation, FAPs that can be used together to represent natural expression are grouped together in FAP groups, and can be indirectly addressed by using an expression parameter. The expression parameter allows for a very efficient means of animating faces.

FAP 1 (visemes) and FAP 2 (expressions) are high-level animation parameters. A face model designer creates them for each face model. Using FAP 1 and FAP 2 together with low-level FAPs 3-68 that affect the same areas as FAP 1 and 2, may result in unexpected visual representations of the face. Generally, the lower level FAPs have priority over deformations caused by FAP 1 or 2. When specifying an expression with FAP 2, the encoder may send an init_face bit that deforms the neutral face of the model with the expression prior to superimposing FAPs 3-68. This deformation is applied with the neutral face constraints of mouth closure, eye opening, gaze direction and head orientation. Since the encoder does not know how FAP 1 and 2 are implemented, it is recommended to use only those low-level FAPs that will not interfere with FAP 1 and 2.

Our analysis techniques can generate facial animation parameters related to the FAPs included in Table VI-2. You must notice that all FAPs involve facial expression synthesis except numbers 48, 49, 50, 101, 102 and 103. These latest describe rigid head motion. FAPs 101, 102 and 103 are not specified in the standard. We have added them to include pose translation, t_X, t_Y and t_Z (respectively) as if they were FAPs instead of having implemented a transform node on top of the complete face object. FAPs 48, 49 and 50 represent the rotations α, β and γ respectively.

MPEG-4 FAP are commutative action units whose center of coordinate is intimately related to the head model. Kalman's pose tracking projection-transform world origin is related to the situation of the camera. This implies that all actions are expressed as translations and rotations from the camera perspective. MPEG-4 defines movement from the head perspective. It does not provide FAP for the translation of the head because it considers it an external movement exerted over the head element.

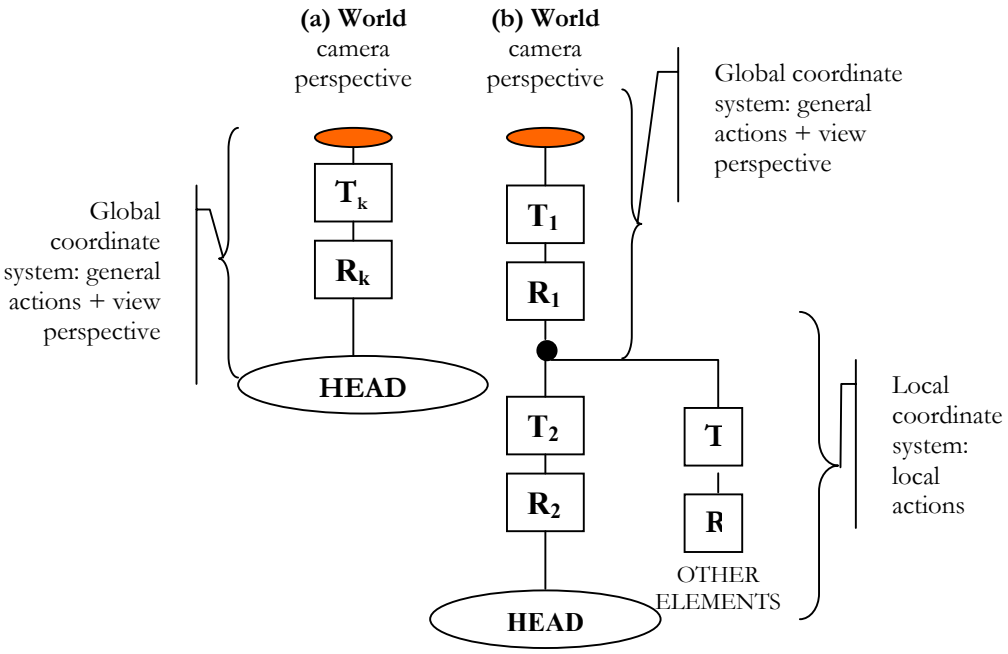


Figure VI-10. (a) Kalman's transform and (b) MPEG-4 transform action

Schematically we could understand Kalman's and MPEG-4 transform action over the head like it is shown in Figure VI-10.

Due to the different nature of the transformation system, predicted translation parameters, t_X, t_Y and t_Z which are represented by the T_k matrix, and rotation parameters α, β and γ which conform the R_k matrix must be correctly interpreted to express the same movement on the MPEG-4 system. In the MPEG-4 scene world, T_1 and R_1 represent different global translation and rotation actions to do over all elements in the scene, to obtain the desired view and emplacement of all the objects. T_2 and R_2 represent translation and rotation actions over the head model with reference to the local head coordinate system. Kalman filter prediction obtains the pose parameters per frame related to the camera model coordinates. We associate Kalman translations to an external action performed over the whole head ($T_2 = T_k$). We associate Kalman rotations to the action of FAPs 48, 49 and 50 because these FAPs express natural head rotation actions, all FAPs are expressed over the local head coordinate system.

More generic information about FAPs and FDPs can be found in Appendix VI-H. Some descriptive tables plus the complete list of FAPs has been included in it.

Table VI-2

FAP DEFINITIONS, GROUP ASSIGNMENTS AND STEP SIZES

#	FAP name	FAP description	units	Uni-or Bidir	Positive motion	Grp	FDP subgrp num	Quant step size	Min/Max I-Frame quantized values	Min/Max P-Frame quantized values
3	open_jaw	Vertical jaw displacement (does not affect mouth opening)	MNS	U	down	2	1	4	+1080	+360
4	lower_t_midlip	Vertical top middle inner lip displacement	MNS	B	down	2	2	2	+-600	+-180
5	raise_b_midlip	Vertical bottom middle inner lip displacement	MNS	B	up	2	3	2	+-1860	+-600
6	stretch_l_cornerlip	Horizontal displacement of left inner lip corner	MW	B	left	2	4	2	+-600	+-180
7	stretch_r_cornerlip	Horizontal displacement of right inner lip corner	MW	B	right	2	5	2	+-600	+-180
8	lower_t_lip_lm	Vertical displacement of midpoint between left corner and middle of top inner lip	MNS	B	down	2	6	2	+-600	+-180
9	lower_t_lip_rm	Vertical displacement of midpoint between right corner and middle of top inner lip	MNS	B	down	2	7	2	+-600	+-180
10	raise_b_lip_lm	Vertical displacement of midpoint between left corner and middle of bottom inner lip	MNS	B	up	2	8	2	+-1860	+-600
11	raise_b_lip_rm	Vertical displacement of midpoint between right corner and middle of bottom inner lip	MNS	B	up	2	9	2	+-1860	+-600
12	raise_l_cornerlip	Vertical displacement of left inner lip corner	MNS	B	up	2	4	2	+-600	+-180
13	raise_r_cornerlip	Vertical displacement of right inner lip corner	MNS	B	up	2	5	2	+-600	+-180

#	FAP name	FAP description	units	Uni-or Bidir	Positive motion	Grp	FDP subgrp num	Quant step size	Min/Max I-Frame quantized values	Min/Max P-Frame quantized values
14	thrust_jaw	Depth displacement of jaw	MNS	U	forward	2	1	1	+600	+180
15	shift_jaw	Side to side displacement of jaw	MW	B	right	2	1	1	+-1080	+-360
16	Push_b_lip	Depth displacement of bottom middle lip	MNS	B	forward	2	3	1	+-1080	+-360
17	Push_t_lip	Depth displacement of top middle lip	MNS	B	forward	2	2	1	+-1080	+-360
18	depress_chin	Upward and compressing movement of the chin (like in sadness)	MNS	B	up	2	10	1	+-420	+-180
19	Close_t_l_eyelid	Vertical displacement of top left eyelid	IRISD	B	down	3	1	1	+-1080	+-600
20	Close_t_r_eyelid	Vertical displacement of top right eyelid	IRISD	B	down	3	2	1	+-1080	+-600
21	Close_b_l_eyelid	Vertical displacement of bottom left eyelid	IRISD	B	up	3	3	1	+-600	+-240
22	Close_b_r_eyelid	Vertical displacement of bottom right eyelid	IRISD	B	up	3	4	1	+-600	+-240
23	yaw_l_eyeball	Horizontal orientation of left eyeball	AU	B	left	3	na	128	+-1200	+-420
24	yaw_r_eyeball	Horizontal orientation of right eyeball	AU	B	left	3	na	128	+-1200	+-420
25	Pitch_l_eyeball	Vertical orientation of left eyeball	AU	B	down	3	na	128	+-900	+-300
26	pitch_r_eyeball	Vertical orientation of right eyeball	AU	B	down	3	na	128	+-900	+-300
31	raise_l_i_eyebrow	Vertical displacement of left inner eyebrow	ENS	B	up	4	1	2	+-900	+-360
32	raise_r_i_eyebrow	Vertical displacement of right inner eyebrow	ENS	B	up	4	2	2	+-900	+-360
33	raise_l_m_eyebrow	Vertical displacement of left middle	ENS	B	up	4	3	2	+-900	+-360

#	FAP name	FAP description	units	Uni-or Bidir	Positive motion	Grp	FDP subgrp num	Quant step size	Min/Max I-Frame quantized values	Min/Max P-Frame quantized values
		eyebrow								
34	raise_r_m_eyebrow	Vertical displacement of right middle eyebrow	ENS	B	up	4	4	2	+-900	+-360
35	raise_l_o_eyebrow	Vertical displacement of left outer eyebrow	ENS	B	up	4	5	2	+-900	+-360
36	raise_r_o_eyebrow	Vertical displacement of right outer eyebrow	ENS	B	up	4	6	2	+-900	+-360
37	squeeze_l_eyebrow	Horizontal displacement of left eyebrow	ES	B	right	4	1	1	+-900	+-300
38	squeeze_r_eyebrow	Horizontal displacement of right eyebrow	ES	B	left	4	2	1	+-900	+-300
48	head_pitch	Head pitch angle from top of spine	AU	B	down	7	na	170	+-1860	+-600
49	head_yaw	Head yaw angle from top of spine	AU	B	left	7	na	170	+-1860	+-600
50	head_roll	Head roll angle from top of spine	AU	B	right	7	na	170	+-1860	+-600
51	lower_t_midlip _o	Vertical top middle outer lip displacement	MNS	B	down	8	1	2	+-600	+-180
52	raise_b_midlip_o	Vertical bottom middle outer lip displacement	MNS	B	up	8	2	2	+-1860	+-600
53	stretch_l_cornerlip _o	Horizontal displacement of left outer lip corner	MW	B	left	8	3	2	+-600	+-180
54	stretch_r_cornerlip _o	Horizontal displacement of right outer lip corner	MW	B	right	8	4	2	+-600	+-180
55	lower_t_lip_lm _o	Vertical displacement of midpoint between left corner and middle of top outer lip	MNS	B	down	8	5	2	+-600	+-180
56	lower_t_lip_rm _o	Vertical displacement of midpoint between right corner and middle of top outer lip	MNS	B	down	8	6	2	+-600	+-180

#	FAP name	FAP description	units	Uni-or Bidir	Positive motion	Grp	FDP subgrp num	Quant step size	Min/Max I-Frame quantized values	Min/Max P-Frame quantized values
57	raise_b_lip_lm_o	Vertical displacement of midpoint between left corner and middle of bottom outer lip	MNS	B	up	8	7	2	+-1860	+-600
58	raise_b_lip_rm_o	Vertical displacement of midpoint between right corner and middle of bottom outer lip	MNS	B	up	8	8	2	+-1860	+-600
59	raise_l_cornerlip_o	Vertical displacement of left outer lip corner	MNS	B	up	8	3	2	+-600	+-180
60	raise_r_cornerlip_o	Vertical displacement of right outer lip corner	MNS	B	up	8	4	2	+-600	+-180
101	tx	Horizontal displacement along the x-axis	MNS	B	left					
102	t_Y	Vertical displacement along the y-axis	MNS	B	up					
103	tz	Depth displacement along the z-axis	MNS	B	forward					

FAPS NAMES MAY CONTAIN LETTERS WITH THE FOLLOWING MEANING: **l = left, r = right, t = top, b = bottom, i = inner, o = outer, m = middle**. THE SUM OF TWO CORRESPONDING TOP AND BOTTOM EYELID FAPS MUST EQUAL 1024 WHEN THE EYELIDS ARE CLOSED. INNER LIPS ARE CLOSED WHEN THE SUM OF TWO CORRESPONDING TOP AND BOTTOM LIP FAPS EQUALS ZERO. FOR EXAMPLE: **(lower_t_midlip + raise_b_midlip)** = 0 WHEN THE LIPS ARE CLOSED. ALL DIRECTIONS ARE DEFINED WITH RESPECT TO THE FACE AND NOT THE IMAGE OF THE FACE.

HEAD POSE	EYES	EYEBROWS	MOUTH

OpenGL implementation

We use OpenGL to render the speaker's 3D head model and simulate the video view acquired from the camera. OpenGL uses its own perspective projection model, as we see it illustrated in Figure VI-11. It is capable of rendering all elements that fall inside the volume defined by the *zNear* plane, the *zFar* plane and the *fovy* angle. The objects are rendered on the *viewport* of the application, or rendering window whose characteristics are determined by *b=bottom*, *t=top*, *l=left* and *r=right*. The *viewport* contains the image representation of the object as if it had been focused on the *zNear* plane. The *fovy* angle, the *zNear* plane, and the *viewport* are highly related. Once two of them are set, the third

one is uniquely determined. We control the final projection settings using OpenGL's call `glFrustum(l,r,b,t,zNear,zFar)` that directly establishes the volume that will be rendered on the *viewport* thanks to the *fovy* angle being automatically deduced.

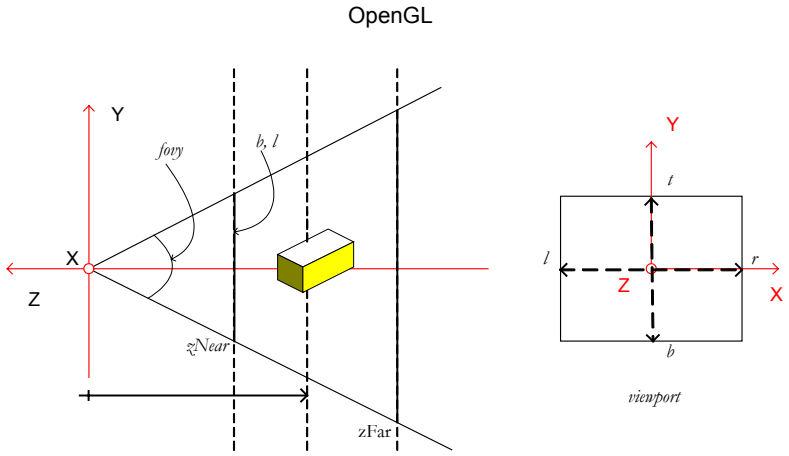


Figure VI-11. Perspective projection model and reference system in the synthetic world generated by OpenGL. The objects are focused on the *zNear* plane and they are rendered on the *viewport*. The *viewport* is determined by *t*=top, *r*=right, *b*=bottom and *l*=left, and takes the size that will be presented on the screen window that must match the size characteristics of the video data.

Since the synthetic rendering must adjust to reality, the synthetic head representation must fit into the video input dimensions regardless of the dimensions of the 3D model itself; otherwise the initialization step, the pose-tracking and the expression analysis would not be feasible. Synthetic worlds are expressed in generic units, here called *sw_u*, that do not represent any specific magnitude by themselves, therefore, before doing any analysis, we need to related those generic units to the real input we get from the camera.

To match the synthetic world with real life, we consider the proposed standard acquisition conditions: a small camera situated in front of the speaker on top of the monitor around 75 cm away from him. To recreate the real world during the synthesis independently of the model characteristics, that is, its size, we rely on the following considerations:

– First, we set an anthropometric generalization, we establish that the distance ES0 is equivalent to 10 cm for all head models

$$ES0 = ES \cdot 1024 = 10\ cm\ .$$

– Then, our real acquisition conditions can be summarized by determining $fovy = 40°$. This choice results in the following parameters if we take into account that the width of the input image is around 384 pel and that we consider a screen resolution of 32 pel/cm:

$$r(sw\_u) = \frac{w(pix)}{2} \cdot \frac{10}{32(pel/cm)} \cdot \frac{ES0(sw\_u)}{10(cm)}$$

$$t(sw\_u) = \frac{h(pix)}{2} \cdot \frac{1}{32(pel/cm)} \cdot \frac{ES0(sw\_u)}{10(cm)}$$

$$l = -r \qquad b = -t$$

$zNear(sw\_u) = 2.74 \cdot r$ value obtained from the $fovy$: $f = \cot(fovy/2) = 2.74$

$zFar(sw\_u) = 20 \cdot ES0$, which sets a value large enough to ensure the complete model visualization.

NOTE: The rendered data is accessed in memory like each video frame (Figure VI-5) but its format is RGB32 instead of BGR32.

VI.2.3 Description of the visual test-bed used and how its real implementation would be

As it is illustrated in Figure VI-1 and already mentioned in the beginning of this section, the experimental settings for our tests consists of two parts: the video window and the rendering window. Let us describe their relevance:

- Video Window: On this window, we plot the video input from the acquisition system. On top of it we draw the following:
  - Red/blue squares: these are the blocks used to perform the block-matching in 2D to track the face features that will be used during the Kalman-based pose-tracking algorithm.
  - Wider blue/yellow rectangles: these are the feature ROIs for eyes, eyebrows and mouth. They are obtained from projecting the 3D ROIs defined over the speaker's head model onto the video image plane.
  - White line/points inside the ROIs: they are the result from drawing the output from then the image processing analysis algorithms used on each feature.
  - Green line/points inside the ROIs: with them, we draw the final appearance of the projection of the feature 3D neutral-motion models after having applied on them the motion parameters

Theoretically, white and green lines should be drawn alike on the ROIs, thus indicating that the complete process: undoing pose and projection plus motion analysis, has worked well.

- Rendering Window: This window shows the synthesis of the 3D head used for the analysis. This model is only animated with the rigid-motion parameters predicted by the Kalman pose-tracker. Exceptionally, an avatar is used during some of the tests for the eye-motion analysis. In this case, the avatar's eyes are also animated.

This experimental framework differs from what it should be expected on a real life application implementation. Indeed, it has only been built for experimental purposes and no optimization has been applied.

A general application would develop the analysis and synthesis required for the coding in background mode. This would imply that there would be no need for a visual implementation related to the analysis or the synthetic results involved during the encoding of motion parameters, unless we would explicitly like to control the analysis procedure results. The only required visual implementation would be related to the receiver part. It would fundamentally include the rendering of the speaker's clone and any virtual elements, such other speakers and common environmental space to share during communications.

VI.3 Head-Pose Tracking Based on an Extended Kalman Filter

This section briefly reviews the algorithm utilized for the rigid-motion tracking in our system. The head tracker is the result of previous work at the Image Group. Some of the information presented comes from S. Valente's Ph.D. thesis (Valente, 1999).

VI.3.1 Theoretical review

To summarize the theoretical bases of Kalman filtering let us consider that the complete procedure is about estimating the state Ψ_t of a certain system at instant *t*. Ψ_t is not directly accessible, but it is the result of several observations $s_t = h(\Psi_t)$. We have also a rough idea of the general system evolution along the time, following $\Psi_{t+1} = a(\Psi_t)$. The uncertainty related to the observation equations and the evolution equation is added through the incorporation of two Gaussian white noises, v_t and w_t, respectively, whose covariance are *R* and *Q*.

(VI-1)
$$\begin{cases} s_t = h(\Psi_t) + v_t \\ \Psi_{t+1} = a(\Psi_t) + w_t \end{cases}$$

The observation function and the dynamic evolution function are linearized around the a-priori estimation $\Psi_{t/t-1}$ and the a-posteriori estimation $\Psi_{t/t}$ respectively, with

(VI-2)
$$\begin{aligned} h(\Psi_t) &\approx h(\Psi_{t/t-1}) + H_t(\Psi_t - \Psi_{t/t-1}) \quad \& \\ a(\Psi_t) &\approx a(\Psi_{t/t}) + A_t(\Psi_t - \Psi_{t/t}) \end{aligned}$$

where $H_t = \frac{\partial h}{\partial \Psi}\Big|_{\Psi_t = \Psi_{t/t-1}}$ and $A_t = \frac{\partial a}{\partial \Psi}\Big|_{\Psi_t = \Psi_{t/t}}$ are the Jacobians of functions $h(\cdot)$ and $a(\cdot)$.

After some computations (Kay, 1993), the a-posteriori estimation of Ψ_t and the covariance matrix of the error associated $P_{t/t}$ are given by the following filter equations:

(VI-3)
$$\begin{cases} K_t = P_{t/t-1} H_t^T \left(R + H_t P_{t/t-1} H_t^T\right)^{-1} \\ \Psi_{t/t} = \Psi_{t/t-1} + K_t\left(s_t - h(\Psi_{t/t-1})\right) \\ P_{t/t} = (I - K_t H_t) P_{t/t-1} \end{cases}$$

and the a-priori estimation, with the error covariance $P_{t+1/t}$ is given by the prediction equations:

(VI-4)
$$\begin{cases} \Psi_{t+1/t} = a\left(\Psi_{t/t}\right) \\ P_{t+1/t} = A_t P_{t/t}\ A_t^T + Q \end{cases}$$

We must recall that there is no guaranty for these estimations to be optimal after the linearization. Equations (VI-2) may not make any sense in a practical system, or be numerically unstable, depending on the applicability of the linearization. In our case, the system will remain stable as long as rigid movements of the head are smooth between consecutive frames.

VI.3.2 Use of the extended Kalman filter in our context

Within the context of our application, the Kalman filter is the central core of the head tracker, and basically, has three tasks:

1. it estimates the 3D location and orientation of the speaker from 2D regions tracked on the video input;
2. it predicts the 2D interest centers of the points chosen from the face to be tracked to help in the block matching procedure on the video; and
3. it ensures that the model will be at the same scale, location and orientation on the synthesized image as on the input video image, even though image acquisition has been made with an uncalibrated camera.

Considering equations (VI-3), we see that the filter produces $\Psi_{t/t}$ by rectifying the predicted state $\Psi_{t/t-1}$ by the correcting term $K_t\left(s_t - h\left(\Psi_{t/t-1}\right)\right)$, which takes into account the difference between observations s_t obtained at instant t and their predictions $h\left(\Psi_{t/t-1}\right)$. Therefore, the action of the Kalman filter can be considered as an iterative process that adjusts the system state $\Psi_{t/t}$ to make it correspond to the observation s_t and to the dynamic evolution model of equations (VI-4) at the same time.

This interpretation, which takes the system as an iterative adjustment, helps to understand how the filter is able to align the synthetic model of the speaker with his head on the real image, by estimating the 3D location and orientation of the real head, and by ensuring both objects to be the same size, regardless of an uncalibrated camera whose focal length is unknown. This is achieved by using the observation model $h(\cdot)$ that corresponds to the geometrical transformations performed by the synthesis (in our case the OpenGL engine) to project the head model on the image plane, and not the equations (undetermined, because there is no calibration) of the camera perspective projection. The filter will align automatically the model and the head on the synthetic reproduction by taking the location and the orientation of the synthetic model inside the

Euclidean space used by OpenGL and $s_t = h(\Psi_t)$, the vector of 2D coordinates of the facial features being tracked on the model synthetic image, as the state vector.

Dynamic evolution model

The 6 parameters that are needed to control the synthesis of the clone are:

$$\psi = (t_X, t_Y, t_Z, \alpha, \beta, \gamma)^T$$

which represent the 3 degrees of freedom of the model translation and the 3 degrees of freedom of its rotation, related to the x-, y- and z-axis of the synthetic world. These are concatenated to their first and second derivatives inside the filter state vector:

$$\Psi = \left(\psi^T, \dot{\psi}^T, \ddot{\psi}^T, \dddot{\psi}^T \right)$$

that follows the system dynamics, which is based on the hypothesis that the system works under constant acceleration:

(VI-5) $$\psi_{t+dt} = \psi_t + \dot{\psi}\, dt + \ddot{\psi}\, dt^2 .$$

Observation model

Although the acquisition camera is not calibrated, we can consider that it makes a perspective projection of the real world and not an orthographic projection. The synthetic module must then mime the perspective projection, whose focal is *F*, taken by the filter. This projection is the observation model utilized for the adaptation process of the feature motion analysis algorithms (IV-2) and that Figure VI-12 recalls.

This requirement is needed because we want to use the predicted pose parameters to extend the usage of the expression analysis algorithms developed to study a frontal view of the face to any other pose observed on the video image.

OpenGL implementation of the model used

The observation model that is used during the analysis must be implemented in OpenGL to enable the synthetic rendering of the 3D head model to be the same size and to have the same pose as the speaker's head on each video frame. As seen in Section VI.2.2, OpenGL handles the projection and the manipulation of the 3D head model in a specific way. In Figure VI-13, we compare the reference system and the components of the observation model proposed for head-pose tracking and algorithmic extension of expression analysis to the reference system and the components of its OpenGL practical implementation.

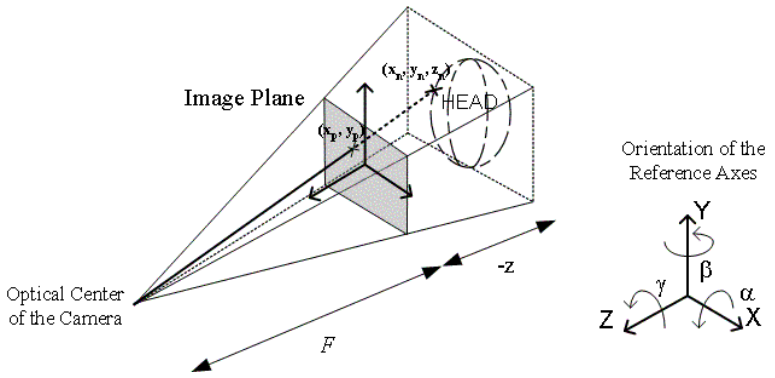


Figure VI-12. Schema of the reference system and camera model used (of focal length *F*) for the adaptation process. It establishes the relationship of a point in the Euclidean space $\mathbf{x}_n = (x_n, y_n, z_n)^T$ and its projected counterpart on the camera image plane $\mathbf{x}_p = (x_p, y_p)^T = \left(\frac{F \cdot x_n}{F - z_n}, \frac{F \cdot y_n}{F - z_n} \right)^T$. The axis orientation is such that the camera only sees the negative part of the Z-axis

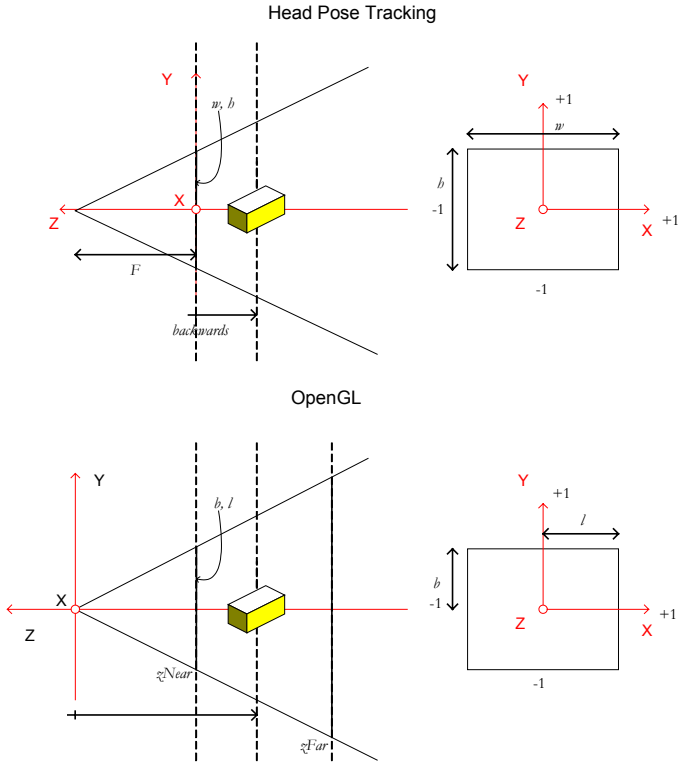


Figure VI-13. Side view of the proposed observation model and its OpenGL practical implementation. In both systems, reference system and components slightly differ but pose and motion description stays the same

After taking into consideration the new references, the final expressions of the 3D→2D relationship are $x_p\big|_{OGL} = \dfrac{-x_p\big|''_{2D}}{l}$ and $y_p\big|_{OGL} = \dfrac{-y_p\big|''_{2D}}{b}$ from the adapted version of system (V-2):

(VI-6) $$\begin{bmatrix} x_p \\ y_p \end{bmatrix}''_{2D} = \frac{zNear}{N}\begin{bmatrix} c_\beta c_\gamma x_n\big|_{OGL} - c_\beta s_\gamma y_n\big|_{OGL} + s_\beta z_n\big|_{OGL} + t_X\big|_{OGL} \\ (s_\alpha s_\beta c_\gamma + c_\alpha s_\gamma)x_n\big|_{OGL} - (s_\alpha s_\beta s_\gamma - c_\alpha c_\gamma)y_n\big|_{OGL} - s_\alpha c_\beta z_n\big|_{OGL} + t_Y\big|_{OGL} \end{bmatrix}$$

$$N = (c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma)x_n\big|_{OGL} + (-c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma)y_n\big|_{OGL} - c_\alpha c_\beta z_n\big|_{OGL} - t_z\big|_{OGL} + zNear + backwards$$

where $zNear = F$, $l = -\frac{w}{2}$, $b = -\frac{h}{2}$ and

$t_X\big|_{OGL} = t_X$, $t_Y\big|_{OGL} = t_Y$ & $t_Z\big|_{OGL} = t_Z + backwards$.

VI.3.3 Influence of the tracking dynamics on the expression analysis

Valente (1999) made a complete and exhaustive analysis about the comportment of the extended Kalman Filter of the head pose tracker utilized. From this analysis, we would like to point out than even when the tracker works fine, the obtained results are slightly noisy and the strongest artifacts appear in the presence of rapid transitions, thus indicating that the prediction model of the Kalman filter, assuming constant acceleration, is not appropriate at those points (see Figure VI-14).

The interference of the noise of the predicted pose parameters in the feature motion interpretation is comparable to the image-processing accuracy. Strong tracking artifacts, which are not common, mask any other effect and frequently lead to erroneous results.

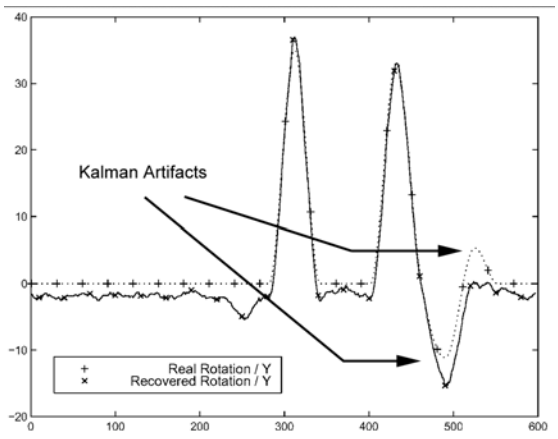


Figure VI-14. Real and recovered Y position of a sample sequence

To better understand the dynamic nature of the inaccuracy introduced by the filter during the tracking, we tried to reproduce the moment at which pose tracking starts. All pose parameters were set to zero and the head was facing the camera in its neutral state. The speaker remained idle for the first seconds of the sequence. The pose parameter predicted during the first 3 seconds are plotted in Figure VI-22. Although no rigid motion should have been noticed, the system noise introduces fluctuations that alter the expected parameter values, which theoretically should stay at zero. The observed fluctuations come from two different origins. In the one hand, we see the sinusoidal deviations caused by the filter itself (most noticeable during the first second); in the other hand, we observe that the noisy data acquired during the tracking is translated into more noise added to the final value (noticeable during the third second).

Fortunately, after studying the magnitude of the error committed, we realize that the fluctuations introduced can be neglected during feature image-processing if we compare their influence to that of the artifacts. We refer the reader to Section 7 in Chapter V where the theoretical evaluation of the pose-tracking inaccuracy influence on the algorithmic extension of the feature processing has been developed. There, the reader will find the bases upon which we have been able to judge the Kalman interference in the complete analysis.

This evaluation has allowed us to support our first assumption regarding using a pose-tracker based on an extended Kalman filter for our coupled pose-expression analysis. If the acquisition is smooth, and the filter dynamics are adapted accordingly, the algorithmic extension of the image processing involved during feature analysis does not suffer much from the Kalman filter dynamic nature.

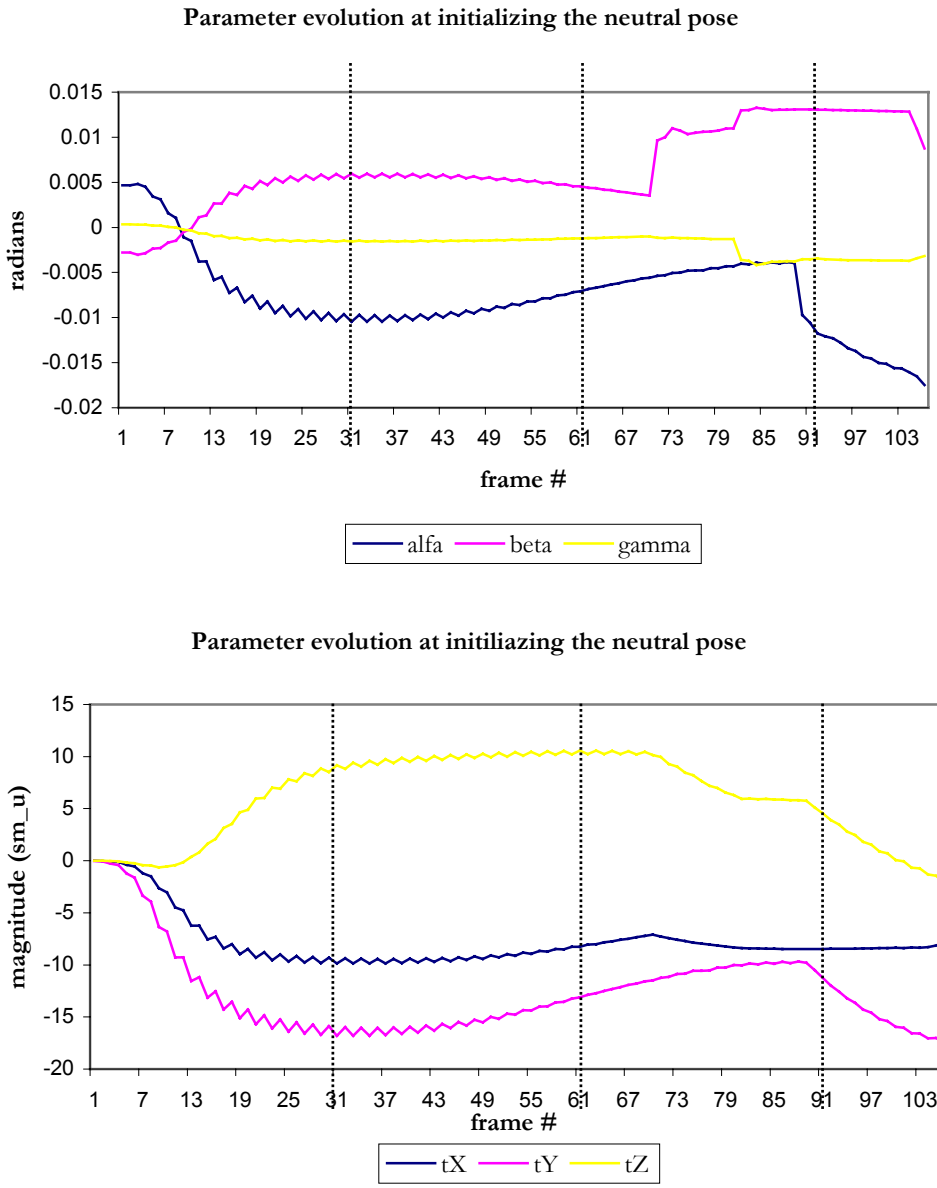


Figure VI-15. These two graphs show the fluctuations that the Kalman filter introduces in the pose values utilized for the head tracking and the expression analysis algorithmic extension. Head model dimensions: WIDTH = 2408.38; HEIGHT = 2871 sm_u & DEPTH = 2847.01 sm_u

VI.4 Evaluating the Motion Template Extension

The practical implementation of algorithms introduces a new source of inaccuracy: the double precision used during computer mathematical operations. Inside our system, errors coming from the mathematical computer manipulations are equivalent to errors coming from the inaccuracy of the data obtained during the image-processing manipulations on the frames. Therefore, they have the form of the expressions treated and studied in Section 7.2 of Chapter V:

$$\widetilde{x}_p = x_p + \varepsilon_x \text{ and } \widetilde{y}_p = y_p + \varepsilon_y .$$

We will recall that the multiplicative errors and the additive errors derived from this imprecision have a different impact on the final value recovered depending on the state of the system: pose parameters, focal length, etc. Under our analysis conditions, the computational inaccuracy is visually translated as a ± 1 pel difference between the original data analyzed and the data obtained after undoing the projection and redoing the projection of these data.

It is difficult to set a quantitative method to enable the evaluation of the accuracy with which the algorithms perform after having been coupled with the pose. The use of real input is the best way to evaluate the real performance of the techniques but does not allow us to control beforehand the *correct outcome* of the analyzed expressions. Moreover, it becomes hard to detect and understand the origin of the inaccuracy; does it come from pose coupling inadequacy, image-processing failure or imprecision during the Kalman pose prediction?

Nevertheless, we have tried to evaluate as concisely as possible how motion template analysis behavior is influenced by the adaptation. First, in Subsection VI.4.1, we discuss how the evolution of the area of analysis alters the processing. Then, in Subsection VI.4.2 we study as quantitatively as possible the limitations of the adaptation in terms of freedom of movement.

Qualitative tests are easier to perform. The visual feedback obtained from the analyzed data can be plotted on the video input. It allows us to verify the performance of the algorithms. The results from the visual evaluation techniques are discussed in Subsection VI.4.3.

VI.4.1 Interference in the image-processing: Deformation of the ROI and introduction of artifacts from other features

The first interference in the deployment of the image-processing analysis comes from the adaptation of the theoretical ROIs to the physical square nature of frames in video memory.

For practical purposes, to make the deformed areas more suitable for image analysis we enclose them in video analysis rectangles $(x_{top}, y_{top}) \rightarrow (x_{bottom}, y_{bottom})$:

$$(x_{top}, y_{top}) = (\min(x) \in ROI, \max(y) \in ROI);$$
$$(x_{bottom}, y_{bottom}) = (\max(x) \in ROI, \min(y) \in ROI).$$

This ensures that the ROI and its feature are completely inside the analyzed area. Unfortunately it also implies the inclusion of some artifacts coming from other facial features next to the one being analyzed. In some cases, like with hair and eyes when analyzing eyebrows, it can be taken into account during the image processing; otherwise they will be possible sources of error that the system will have to control. Figure VI-16 illustrates an example where the eye feature is very much included inside the eyebrow ROI. In this case, the algorithm has resolved positively for the right eye but it is not able to recover the right shape from the left eye, this ROI is not framed well enough.

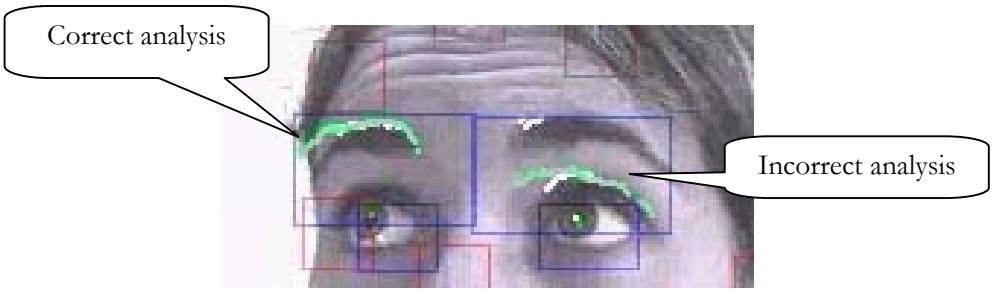


Figure VI-16. The eyebrow-motion analysis algorithm has been able to avoid the influence of the eye feature that is also covered by the eyebrow ROIs when analyzing the right eyebrow. For the analysis of the left eyebrow, the inaccuracy of the ROI determination prevent the algorithm from properly detecting the eyebrow and it detects the eye instead

VI.4.2 Influence of the surface linear approximation

The study of the effect of the surface linear approximation on the performance of the adapted algorithms is not an easy task, because the success of the analysis depends on several factors at the same time: correct pose prediction, right image-processing results, accurate surface approximation, etc. We were capable of studying the algorithmic performance of the template designs and their related image-processing techniques independently of the pose when analyzing faces from a frontal perspective but it is impossible to detach the influence of the algorithmic extension process of these algorithms from their own performance when they work coupled with the pose.

Nevertheless, we have tried to set a quantitative evaluation to estimate how important the influence of the surface design is in obtaining proper results from the coupled analysis. To do so, we have set some experiments on the coupled *eye-state tracking*-algorithm using the realistic 3D head model of the speaker as information source to determine the 3D-ROI.

We have taken into account the data obtained from the analysis of the left and the right eye of an individual whose pose had been pre-established and fixed during the recording of the analyzed sequence. These data are represented in Figure VI-17 and determine the physical relationship between the extracted pupil location and the dimensions of the 3D-ROI on the surface linear approximation. Figure VI-18 depicts the exact 3D-ROI coordinate values taken during the tests.

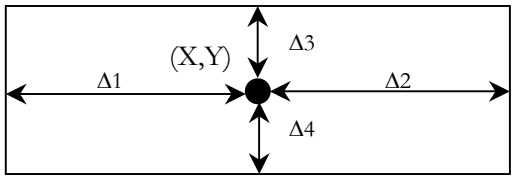


Figure VI-17. Data extracted during the *eye-state tracking*-algorithm study

Table VI-3

RESULTS FOR A FACE IN NEUTRAL POSITION

FAP	EYE	MAG	STATS	X	Y	Δ1	Δ2	Δ3	Δ4
FAPs SET TO 0	LEFT		mean	246.0778	206.9484	126.1229	152.4654	91.98254	47.37912
			max	277.0066	219.5734	155.2335	164.5529	98.72634	59.54971
			min	234.3405	202.0495	113.7071	121.2408	78.65432	38.2181
			stdev	6.174268	5.461572	5.799478	6.271002	5.579391	4.868758
	RIGHT		mean	-330.762	192.4423	171.9816	107.1061	84.35926	69.14623
			max	-320.416	203.7004	182.2539	149.4036	93.25912	83.16978
			min	-372.631	185.7216	130.0194	95.85251	74.51731	51.78346
			stdev	12.79068	4.480565	12.7143	12.8921	4.020785	7.108381

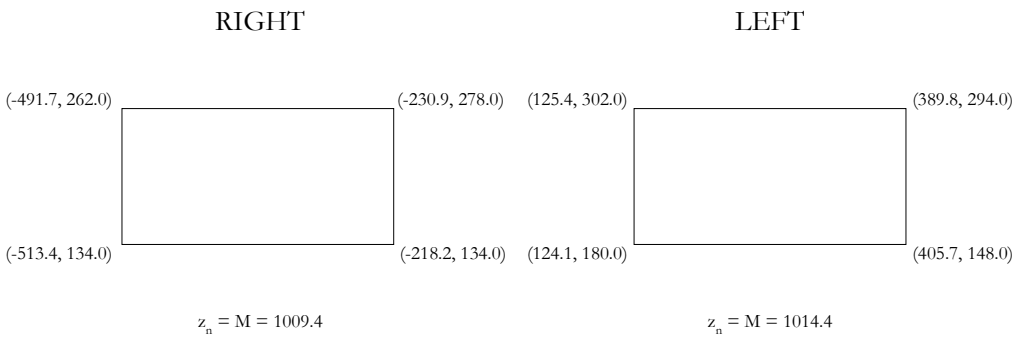


Figure VI-18. Coordinate extracted from the 3D head model of the speaker used to conform the ROIs for the *eye-state tracking*-algorithm adaptation

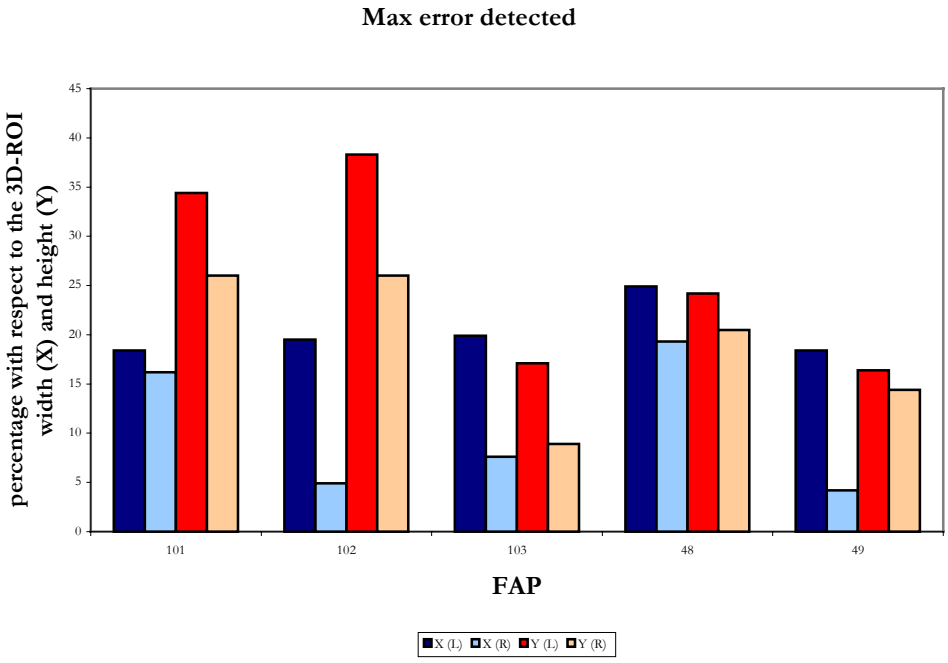
We preset as correct data the results obtained when the technique is applied on the face in its neutral state ($\alpha = 0$, $\beta = 0$, $\gamma = 0$, $t_X = 0$, $t_Y = 0$ and $t_Z = 0$), on Table VI-3. We have compared them against the results obtained from varying the translation and the rotation parameters of the speaker, each parameter at a time, the t_X evolution on Table VI-4, the t_Y evolution on Table VI-5, the t_Z evolution on Table VI-6, the α evolution on Table VI-7, and the β evolution on Table VI-8,. Rotation around the z-axis (γ) over more than 10 degrees is a physical movement difficult to make by human heads. We have not provided data related to this rigid movement because, in this case, the surface approximation is parallel to the image plane, and the γ rotation only implies a simple surface rotation that alters neither the surface projection nor its size on the image.

After studying the different results obtained, we point out that it does not seem to exist any correlation between the performance of the analysis technique and the pose at which the surface is at the moment of the coupling. This implies that the linear surface approximation works fine as long as the feature is completely visible on the image (as it is with the FAP values tested). Interestingly, we observe that the performance is better for the right eye than for the left eye. We refer the reader to Figure VI-19 where we have plotted the maximum error found in the mean of the X, and Y pupil components. This result allows us to think that the selected surface approximation for the right eye was better than for the left one; in equal analysis conditions left analysis performed poorly compared to right analysis and only the surface approximation differed.

The correct design of the surface approximation seems to be important although not critical for the success of the algorithm because the behavior of the methods and its 3D extension can be controlled beforehand. The experiment was carried out on the *eye-state tracking*-algorithm: it could have been made on the eyebrows and mouth as well. The most outstanding difference among the feature algorithmic extension is the degree of complexity and the amount of visual information required for each. This is one of the

reasons why quantitative testing methods may help us to deduce weaknesses and improve algorithms but qualitative methods, where we can observe the dynamic evolution of the implanted analysis, usually are more helpful to set the limitations of the solution proposed. The current study let us consider the benefits from utilizing different surfaces more conveniently adapted to the feature under analysis to study the possibility of not being constrained by the projection of the feature movement on a plane.

From our analysis results, we conclude that the speaker has almost complete freedom of movement regarding the translation along the x-, the y- and the z-axis, and regarding rotations, we have found a limit of around $\pi/4$ rad. These results will be also corroborated with the visual inspection made during the qualitative evaluation.



	Max err X (L)	Max err X (R)	Max err Y (L)	Max err Y (R)	FAP magnitude			
101	18.4	16.2	34.4	26	-400	-2000	2000	400
102	19.5	4.9	38.3	26	-1000	-1000	1500	-1000
103	19.9	7.6	17.1	8.9	4000	-1000	-2000	1000
48	24.9	19.3	24.2	20.5	$\pi/4$	$\pi/10$	$-\pi/10$	$-\pi/10$
49	18.4	4.2	16.4	14.4	$\pi/10$	$\pi/10$	$\pi/8$	$\pi/6$

Figure VI-19. Maximum error in the average X and Y components found during the study. We have also indicated the FAP magnitude at which these values occurred

Table VI-4

RESULTS FOR THE EVOLUTION OF THE t_X PARAMETER

FAP	EYE	MAG	STATS	X	Y	Δ1	Δ2	Δ3	Δ4
101	LEFT	-7000	mean	219.925	229.4832	95.95131	178.1395	79.78937	81.12986
			max	290.1278	244.9332	165.3946	199.254	91.38835	97.76957
			min	199.1345	227.712	74.50269	108.7536	61.39478	65.34629
			stdev	17.1769	3.683016	16.94613	16.92265	7.671807	6.43468
		-4000	mean	294.326	235.3008	169.6913	104.5385	72.93548	77.42578
			max	307.4229	245.1562	183.1198	150.2955	86.06844	87.78757
			min	247.6567	227.7628	123.5877	90.63201	61.9816	64.31565
			stdev	6.871439	4.091718	6.865197	6.858486	4.667995	4.031714
		-2000	mean	253.2494	234.5019	128.6869	145.2287	65.03371	72.66918
			max	262.068	236.729	137.6282	188.5121	84.72427	90.63206
			min	209.8804	227.6834	85.25098	136.1973	61.4227	63.85165
			stdev	14.05842	3.504702	14.0079	13.98988	5.637639	6.034503
		2000	mean	250.4085	250.0373	126.0246	150.2728	48.71068	87.34518
			max	254.6273	253.7811	129.9214	156.6978	54.82479	91.69401
			min	244.4603	244.5479	119.8116	145.2346	44.36698	81.42266
			stdev	4.236639	4.126267	4.119185	4.524514	3.858066	4.419033
		4000	mean	261.9363	244.7908	137.3414	138.0218	54.43484	81.5112
			max	283.4904	253.7811	158.7262	156.6978	65.47995	91.69401
			min	244.4603	236.0342	119.8116	115.6809	44.36698	72.23683
			stdev	10.65808	5.760467	10.50439	11.03075	5.775014	6.237003
		7000	mean	282.5423	205.703	161.7305	116.501	95.80681	46.30951
			max	295.0778	219.9687	179.1674	130.4735	118.764	56.21337
			min	267.7796	185.2888	146.2925	104.3934	79.03406	36.88454
			stdev	7.98134	6.35164	9.062158	7.06197	8.013016	2.878193
	RIGHT	-7000	mean	-389.237	218.5541	115.2761	165.1656	62.94267	90.77478
			max	-327.207	220.9038	176.8382	181.4287	73.6181	94.88335
			min	-405.513	211.1097	97.96298	103.6463	59.63923	83.14508
			stdev	23.27539	3.048323	23.10699	23.11185	3.907144	2.52405
		-4000	mean	-366.597	189.1911	136.3689	143.0759	83.42232	58.30857
			max	-299.285	194.9419	203.3456	168.6394	100.0875	89.59652
			min	-392.065	176.9187	111.1611	75.65071	76.23475	42.95233
			stdev	20.33025	4.675506	20.26118	20.24973	5.815422	9.048432
		-2000	mean	-398.517	221.8596	107.0668	174.7248	62.35204	94.65901
			max	-373.469	229.4976	131.0633	201.8597	81.24355	111.5172
			min	-425.217	192.1428	80.39166	149.5718	45.18404	59.47315
			stdev	15.2118	5.823907	14.22664	15.3417	8.870487	9.49504
		2000	mean	-355.261	215.8529	148.4655	131.0942	54.94575	82.85967
			max	-350.473	219.4728	153.0653	162.1935	64.27537	87.82519
			min	-386.315	209.3808	116.9861	126.0014	50.58001	76.23918
			stdev	7.219102	4.046596	7.136573	7.230788	4.250856	3.899333
		4000	mean	-335.796	230.7062	170.0263	114.0795	47.62328	101.4921
			max	-318.727	237.6596	187.8567	123.3098	54.26826	112.8497
			min	-343.94	219.9961	160.2017	98.95505	37.20647	88.75775
			stdev	7.8026	5.20118	8.294748	6.987775	3.926502	6.61795
		7000	mean	-315.653	206.0826	187.2048	91.24227	78.62906	88.06772
			max	-310.745	211.7503	192.3242	104.1014	98.31733	95.16147
			min	-328.582	185.2996	174.0637	86.32494	67.48839	74.8249
			stdev	4.457614	5.827966	4.351893	4.400164	6.496804	3.942559

Table VI-5

RESULTS FOR THE EVOLUTION OF THE t_Y PARAMETER

FAP	EYE	MAG	STATS	X	Y	Δ1	Δ2	Δ3	Δ4
102	LEFT	-2000	mean	245.197	192.3889	129.8802	155.2778	106.4166	34.75575
			max	251.2462	200.4152	135.674	158.7917	107.4354	42.87484
			min	241.7257	191.6698	124.3283	148.3843	97.82043	31.0551
			stdev	4.005633	2.070018	3.841649	3.924406	2.061546	2.877312
		-1000	mean	278.6177	251.3966	154.2529	123.0762	51.33262	88.59823
			max	297.3795	256.8583	173.3239	132.8201	62.95206	98.33293
			min	269.8527	239.6115	145.6671	105.3338	43.11845	75.90296
			stdev	5.010166	4.415484	4.977368	5.020808	4.65969	4.47853
		-500	mean	254.5407	238.6042	129.9803	144.5159	59.67881	75.47829
			max	286.3853	254.5894	161.6843	156.2842	69.83079	90.69248
			min	243.5393	228.5133	118.8673	114.197	43.51327	64.84169
			stdev	4.784881	7.178113	4.809195	4.765199	7.091796	7.175541
		500	mean	253.6923	233.1824	129.1818	144.6502	66.47396	71.45454
			max	270.9339	240.9092	146.4048	172.9907	73.75242	85.69173
			min	225.146	229.9883	100.7878	127.3801	57.26589	66.16684
			stdev	14.42353	2.776622	14.39835	14.40432	3.718068	4.777644
		1000	mean	277.1136	230.6912	152.7769	121.1847	70.921	68.79102
			max	286.2333	248.272	161.764	155.1884	80.60551	84.4149
			min	242.9277	222.2509	118.5787	112.0121	49.7859	59.43529
			stdev	11.48173	5.519086	11.54503	11.64827	6.510454	5.087421
		1500	mean	249.34	255.0368	125.4133	152.3533	45.54347	93.21579
			max	275.7057	265.1779	151.7478	175.9759	60.44138	106.1323
			min	223.8858	247.8586	99.3429	125.4298	33.99877	83.9826
			stdev	12.70997	3.438981	12.53413	12.6771	3.715004	3.925349
	RIGHT	-2000	mean	-317.934	179.8864	185.6224	97.01862	100.2502	66.46042
			max	-313.858	231.1966	191.0798	119.5458	107.2732	109.9759
			min	-339.708	173.2526	164.6802	92.298	60.87439	47.57035
			stdev	5.890756	5.10677	5.58816	6.421334	3.677889	6.874812
		-1000	mean	-317.506	214.1602	187.1471	95.93238	72.7098	94.75904
			max	-296.948	231.2566	207.0493	119.5458	107.2515	114.3288
			min	-340.564	173.2526	164.6802	74.11838	44.05392	47.57035
			stdev	10.1248	22.47521	10.0392	10.1379	18.93831	19.8266
		-500	mean	-325.89	226.5194	179.095	103.4454	57.38528	102.0831
			max	-296.303	229.3187	208.286	134.0343	81.49886	117.7124
			min	-356.592	211.5763	149.3166	73.24496	40.96669	82.73627
			stdev	16.76642	4.669084	16.2517	16.81326	12.47746	6.339616
		500	mean	-338.535	213.2155	164.9834	114.4178	61.89358	84.96736
			max	-312.448	225.1477	190.3395	157.9517	82.91387	95.88302
			min	-382.218	197.8732	121.7503	87.85109	47.47649	73.60527
			stdev	12.53531	7.916117	12.0788	12.71047	11.51667	4.759317
		1000	mean	-326.942	205.1239	175.8424	102.4322	73.79271	81.4157
			max	-312.448	215.0832	190.3395	147.5064	83.88652	95.88302
			min	-372.105	189.4624	130.7692	87.85109	64.27607	62.13669
			stdev	7.572697	4.173764	7.571381	7.54872	4.491855	4.503268
		1500	mean	-324.148	230.9958	181.6286	102.8568	54.3588	106.0834
			max	-296.303	238.5534	207.5959	115.8477	81.49886	116.5879
			min	-339.541	211.5763	163.8769	73.24496	43.51426	82.73627
			stdev	9.809918	7.197792	9.4497	10.1562	9.809959	6.576524

Table VI-6

RESULTS FOR THE EVOLUTION OF THE t_Z PARAMETER

FAP	EYE	MAG	STATS	X	Y	Δ1	Δ2	Δ3	Δ4
103	LEFT	-4000	mean	251.8976	224.6033	128.2238	145.9847	73.93777	62.39204
			max	265.5043	234.4628	142.9139	171.7537	88.36849	76.24464
			min	226.0895	215.1438	102.5882	132.3494	63.60701	51.14529
			stdev	7.01798	3.348121	6.844331	7.031149	3.695905	3.269797
		-2000	mean	292.0416	228.9754	167.8752	106.3218	77.51339	70.89633
			max	305.8943	240.9154	182.0263	137.9617	97.95706	83.03074
			min	259.8492	204.8105	136.2984	91.92334	60.63557	43.06638
			stdev	8.106223	7.690899	7.988307	8.305355	8.046848	7.389172
		-1000	mean	307.4242	224.3398	183.6125	90.80142	89.40607	73.94604
			max	315.9628	233.7074	191.3221	118.9123	107.9501	86.37689
			min	279.7753	197.6471	158.7889	82.56905	81.00043	41.14707
			mean	307.4242	224.3398	183.6125	90.80142	89.40607	73.94604
		1000	mean	268.3299	203.209	148.5768	130.8488	95.49774	39.61604
			max	278.6559	211.6701	160.1239	140.5394	112.0375	49.47545
			min	261.4205	186.626	139.781	119.5193	86.50047	22.89537
			stdev	4.467924	6.531743	4.722637	4.500349	6.52148	6.480362
		2000	mean	297.9675	220.7067	174.3832	99.84032	87.63512	66.18577
			max	310.2057	221.1745	186.511	135.9278	96.03233	72.71262
			min	261.8723	212.8519	138.5683	87.59434	77.28714	49.11767
			stdev	9.52775	1.277554	9.41467	9.535803	3.669899	4.240933
		4000	mean	298.213	209.7119	176.3485	100.3409	97.63135	56.9641
			max	310.4111	214.118	188.8356	123.0589	116.3171	63.21651
			min	280.1385	183.943	157.7311	88.60678	87.01593	25.26318
			stdev	9.144704	5.413272	8.49165	9.466426	4.734842	8.022456
	RIGHT	-4000	mean	-337.496	206.5312	165.3272	112.9102	68.28268	78.32914
			max	-331.962	215.8015	170.8404	146.8682	70.20623	85.14013
			min	-371.335	205.4975	131.5028	107.3638	55.07801	72.2836
			stdev	8.458757	1.797702	8.388616	8.481607	2.847266	2.273605
		-2000	mean	-318.299	208.2919	184.7955	94.13896	76.22589	89.38888
			max	-294.879	223.8075	208.8978	116.5613	100.0596	114.1783
			min	-340.834	187.9385	162.6172	70.42624	54.87069	69.38057
			stdev	15.36785	8.75137	15.24474	15.34283	12.69049	10.23039
		-1000	mean	-350.781	189.9964	152.1907	127.3411	82.28904	59.72327
			max	-329.025	200.5315	173.5801	150.0279	97.54331	75.88678
			min	-373.815	172.5871	128.9985	104.5797	69.53636	38.70103
			stdev	15.71559	6.827759	15.54882	15.98406	5.810915	9.487913
		1000	mean	-330.302	179.3967	173.4352	109.1396	96.93129	58.61354
			max	-309.798	188.6022	193.538	129.9999	109.9151	78.19129
			min	-351.936	163.1665	151.5676	86.97756	82.05176	36.68745
			stdev	14.75003	6.352245	14.39605	15.07297	5.76923	12.17704
		2000	mean	-324.576	206.6476	178.2616	100.0347	74.64275	84.8744
			max	-307.741	213.812	195.4805	131.5453	84.50104	98.64401
			min	-355.913	203.542	147.5267	83.14601	56.44756	72.65005
			stdev	16.4342	3.386654	16.37584	16.43846	8.583023	7.795089
		4000	mean	-301.886	195.28	200.7844	78.1812	96.1163	89.10457
			max	-287.294	200.1932	215.3022	108.8319	109.8491	102.3383
			min	-332.546	177.1364	170.1451	63.02283	75.88652	59.38843
			stdev	12.44111	4.947965	12.42321	12.45124	7.508342	9.424056

Table VI-7

RESULTS FOR THE EVOLUTION OF THE α PARAMETER

FAP	EYE	MAG	STATS	X	Y	Δ1	Δ2	Δ3	Δ4
48	LEFT	-π/10	mean	237.6456	237.2588	113.0255	161.0427	64.07643	78.24204
			max	244.5658	242.6668	120.048	163.294	68.76157	83.56178
			min	235.9083	232.7572	111.1366	153.7088	56.85999	72.0141
			stdev	3.45874	4.736241	3.504899	3.661036	4.204653	4.967987
		π/10	mean	290.8042	202.0247	170.7461	108.941	101.9002	46.53065
			max	306.7946	227.9634	186.2653	142.9315	123.2939	67.77965
			min	254.87	181.981	131.522	91.61712	76.08641	32.56059
			stdev	7.868979	8.098661	7.509405	8.21735	7.437336	8.774217
		π/8	mean	269.6426	227.3229	145.6001	128.4144	72.29098	64.17278
			max	298.1257	231.9787	177.7314	150.3578	104.4101	70.11921
			min	247.4704	201.7609	123.8748	101.5141	66.50345	50.26774
			stdev	9.228967	4.874041	9.043291	9.109284	4.730112	4.758761
		π/6	mean	256.8351	233.9161	132.3397	141.6431	64.63682	70.84069
			max	274.3336	246.1658	149.8293	166.6546	73.76933	82.32361
			min	232.1618	225.0372	107.394	123.9479	51.88151	61.14149
			stdev	8.356286	5.180578	8.460894	8.475893	5.112651	5.682316
		π/4	mean	311.2207	229.5553	186.834	87.08439	87.44306	81.20692
			max	340.5875	240.1694	216.6998	138.1935	112.9405	95.4503
			min	260.7265	221.3158	135.9684	57.21357	59.52149	69.50309
			stdev	13.43747	4.362342	13.55671	13.60375	10.46032	6.631284
	RIGHT	-π/10	mean	-363.357	222.3144	141.6708	140.099	48.73185	88.82653
			max	-317.133	247.6632	191.9916	148.8206	69.97012	123.6714
			min	-369.264	202.1121	133.4284	101.2348	28.66169	68.31923
			stdev	9.623885	10.02056	9.969819	9.21977	9.813661	10.42203
		π/10	mean	-279.399	197.0182	223.2511	55.69668	110.0154	107.087
			max	-263.253	211.6288	239.4969	104.6323	127.0329	119.9824
			min	-329.202	181.4793	173.4745	42.12516	73.88058	68.18563
			stdev	7.727528	5.02151	7.681516	8.150497	4.348079	8.112438
		π/8	mean	-312.178	189.7209	190.6242	89.14345	94.29517	77.49928
			max	-303.635	201.6457	199.0274	117.2189	98.57433	91.75735
			min	-340.116	184.376	162.9104	79.36226	84.27386	58.10844
			stdev	5.681711	3.480801	5.649284	5.730223	3.528986	5.085105
		π/6	mean	-320.642	197.6126	181.9878	96.47904	83.3035	78.24847
			max	-305.517	202.9413	197.0864	116.1118	94.31101	89.67708
			min	-340.238	188.1387	162.3712	81.11582	75.13216	66.6452
			stdev	6.813143	3.80264	6.821474	6.862787	4.004831	5.494102
		π/4	mean	-293.892	172.898	210.267	76.90247	118.4077	81.92526
			max	-281.756	180.9858	221.7772	94.20871	124.1996	95.13482
			min	-311.212	162.9361	192.8172	63.58425	107.8125	64.35244
			stdev	6.238889	4.899409	5.973387	6.787179	4.228784	6.712917

Table VI-8

RESULTS FOR THE EVOLUTION OF THE β PARAMETER

FAP	EYE	MAG	STATS	X	Y	Δ1	Δ2	Δ3	Δ4
49	LEFT	π/10	mean	294.6991	200.3803	174.8266	105.3581	104.5178	47.36578
			max	303.2531	211.9634	184.4718	112.3666	120.2727	61.35802
			min	289.6597	185.7314	167.5753	94.98094	92.40566	34.027
			stdev	4.605821	6.709029	4.932044	4.566947	6.753799	5.615816
		π/8	mean	229.5822	227.6263	105.7398	168.3991	75.98682	73.07674
			max	241.5237	238.7725	117.0598	177.8033	85.38113	79.7483
			min	219.997	221.362	96.10943	156.8286	61.39461	62.50158
			stdev	6.461792	4.074793	6.309703	6.420592	4.876006	4.121302
		π/6	mean	277.6898	195.5059	159.579	122.8806	104.5822	34.55937
			max	285.905	206.5694	166.7484	133.6844	111.9994	42.90467
			min	266.2865	188.6899	146.1926	114.2221	92.24594	25.33373
			stdev	5.833289	4.324244	6.193993	5.431168	4.745087	3.664375
	RIGHT	π/10	mean	-319.766	175.1497	184.3212	100.1492	103.653	61.90765
			max	-284.575	188.5364	218.275	109.8637	112.7906	97.23403
			min	-328.69	170.8179	175.0575	62.80974	91.79452	52.34923
			stdev	3.707026	5.119057	3.52873	4.285513	4.396777	5.125885
		π/8	mean	-361.467	182.7307	142.0239	138.8905	87.41822	49.16635
			max	-328.504	194.5924	174.4945	144.2858	92.95119	64.094
			min	-365.978	177.1642	137.4031	105.7968	75.55386	43.1749
			stdev	5.089901	4.441246	4.862038	5.398809	4.432221	4.729107
		π/6	mean	-330.703	171.9559	173.879	111.4829	102.8329	51.91394
			max	-324.88	179.8311	179.5646	116.6699	105.8761	61.15397
			min	-335.85	170.4533	168.1382	103.7694	93.83384	47.39302
			stdev	5.077799	3.170758	5.029354	4.928892	3.354504	4.25833

VI.4.3 Qualitative evaluation of the motion template algorithmic extension to 3D

To study the performance of the proposed algorithmic extension, we set several experiments where visual feedback from the analysis was provided.

We implemented the adaptation for eyes and eyebrows like it is explained in Section VI.2. The mouth, whose motion is very complex to analyze, is a separate topic of research and it would require more detailed testing and to the possibility of complementing the analysis with some speech processing.

Evaluating the *eye-state tracking*-algorithm using an avatar

We applied the eye animation parameters (FAPs: 19, 20, 21, 22, 23, 24, 25 & 26) to the head model *Olivier* (see Figure VI-20) to study the degree of naturalness that we could obtain from the *eye-state tracking*-algorithm.

First, we looked at the efficiency of the algorithm when it was only applied without taking into account the pose (VIDEO: frontal.2.avi). Results were encouraging and demonstrate that immediate understanding and replication of eye motion clearly delivers a fine sense of life to the avatar. Then, we performed the same kind of test but having coupled the analysis algorithm with the rigid-motion information provided by the pose predictor based on an extended Kalman filter (VIDEO: eyecoupling.avi).

The naturalness achieved by the coupling is outstanding. Eye and pose motion applied together add to the avatar a natural feeling difficult to obtain with automatic standard facial animation techniques.

Regarding technical issues, the practical implementation of this test-bed allowed us to examine how performing the method becomes when utilizing 3D data that has been extracted from a head model other than the speaker's clone. We conclude that not using

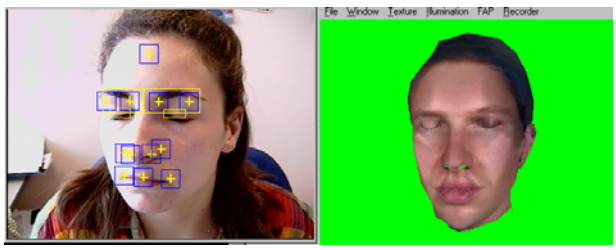


Figure VI-20. Facial animation parameters for the eyes were extracted using the eye-state tracking-algorithm and immediately applied and rendered on the *Olivier* avatar

the speaker's realistic head model was not a hard constraint for the eye analysis algorithm because this latest is simple enough to easily be adapted to any model available. Nevertheless, to use a different 3D-head implies restrictions in the speaker's movements. The pose tracking algorithm could not recover and track back to the neutral head position after the speaker's head had rotated (β parameter) more than ± 20 deg.

The use of a realistic 3D representation of the speaker permits higher freedom of movement; as it could be appreciated during the tests described in the next section.

Evaluating the performance of the eyebrow motion analysis algorithm

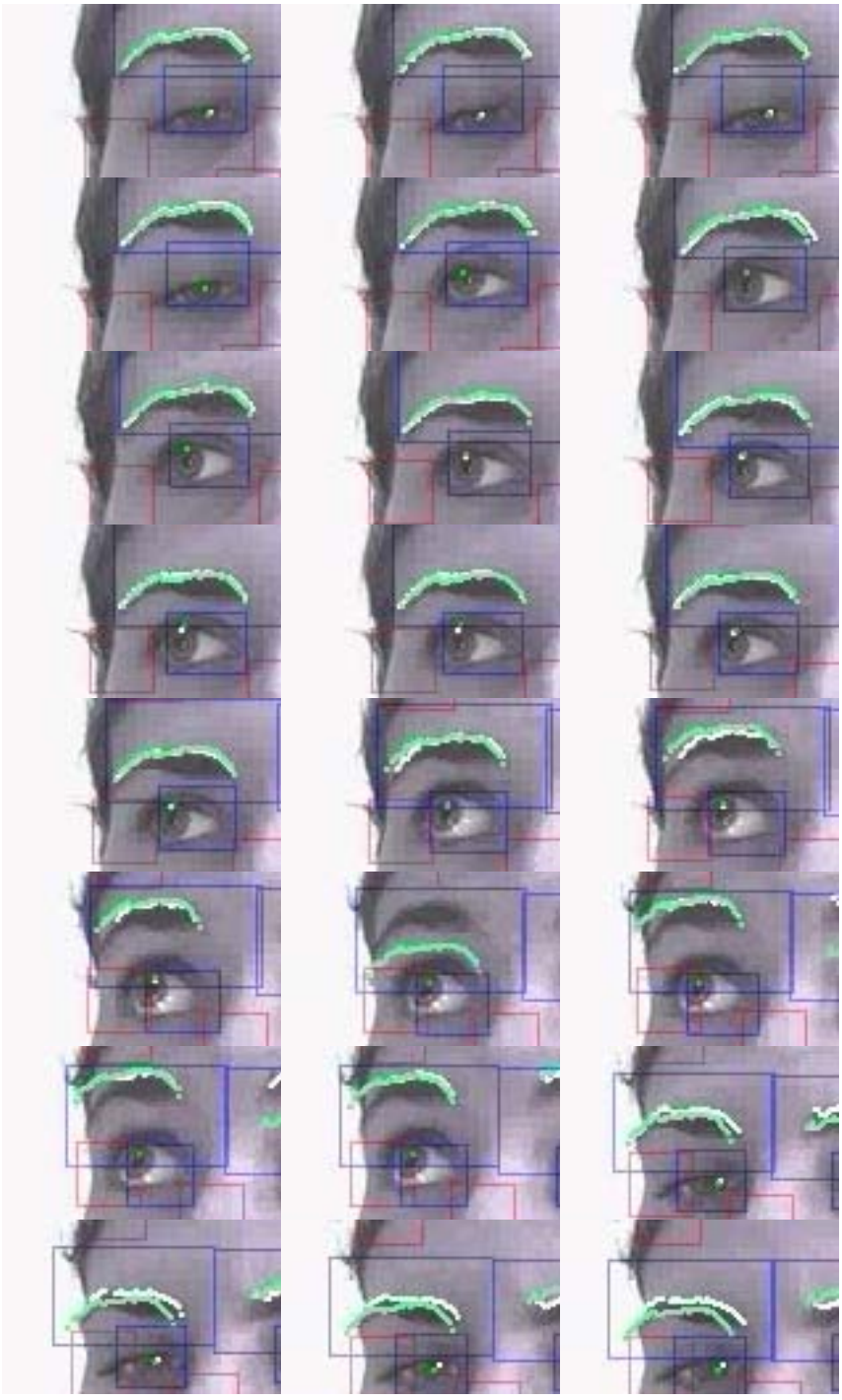
To visually check the performance of the eyebrow analysis algorithm after it had been coupled with pose information, we designed a graphical feedback on the video window of our test platform. The arch obtained from the analysis ("current frame arch") was drawn on each video frame along with the arch ("modeled arch") resulting from applying the motion parameters extracted during the current analysis on the arc extracted while the eyebrow was not moving ("neutral arch"). These two arches were the basis of the quantitative analysis presented in Section 4.3 of Chapter IV, when we exposed the objective tests carried out to evaluate the algorithm in faces analyzed from a frontal perspective. In Figure VI-21, we can see a series of shots extracted from one of the final studied sequences. The "current frame arch" is plotted in white, the "modeled arch" in green. Ideally, if the algorithm works perfectly both arches should be drawn very close to each other.

From the tests we have made, whether it was in the laboratory environment or away, we have been able to conclude that the coupled analysis can extract meaningful motion data as long as the head does not rotate more than $\pm \pi / 4$ deg. The analysis is tolerant to translations.

During these experiments we used a realistic 3D-head model of the speaker. This is the reason why it was possible to recover a neutral position from a wider ranger of rotation than using an avatar.

The algorithm was able to extract coherent motion parameters even when the eyebrow was not completely visible. Although the extracted data are not accurate, they provide meaningful information. They represent the best approximation to the observed movement that we can obtain.

The algorithm has been designed not to draw errors through the analysis of the sequence. If the process does not succeed in analyzing correctly one specific frame, the rest of the sequence does not suffer from this incident. We could observe this behavior when the processing recovered graciously from an incorrect result.



(It continues on the next page)

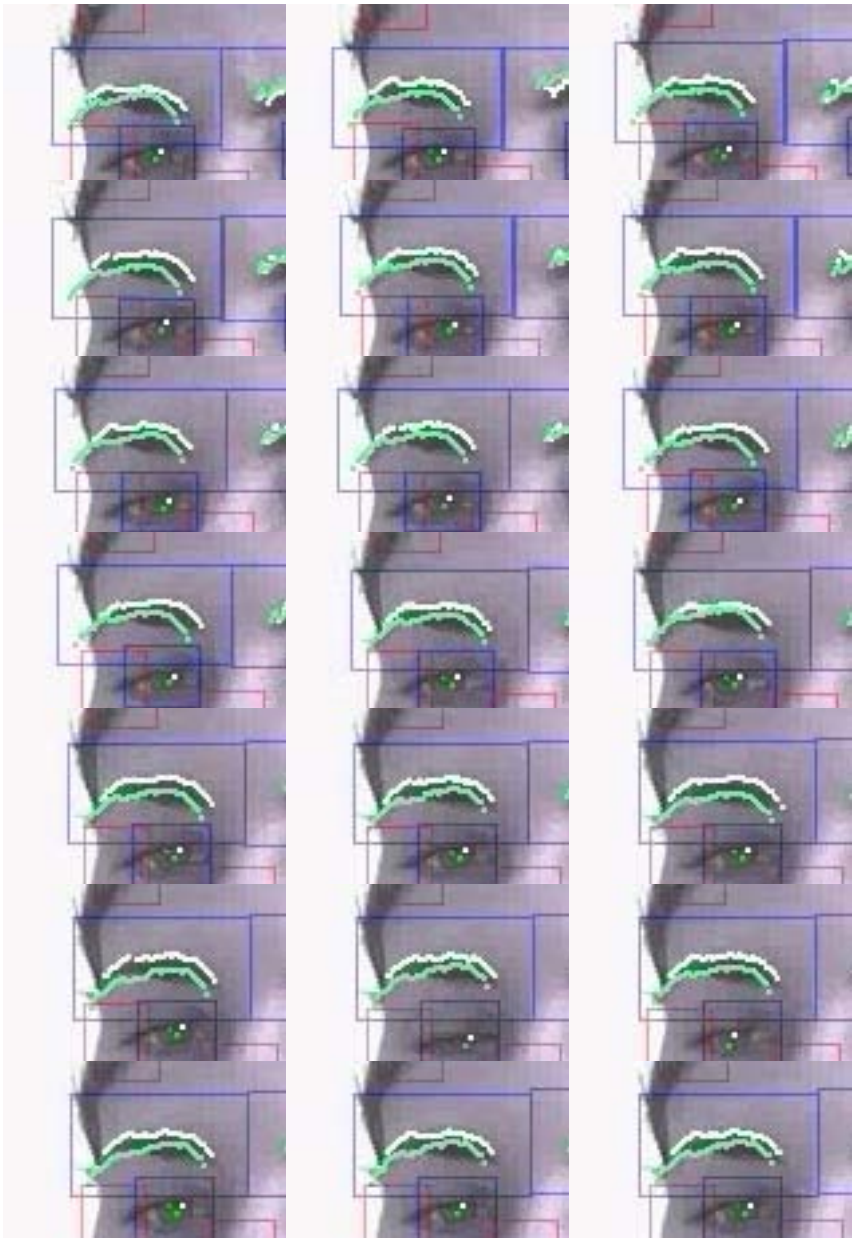


Figure VI-21. Sequence of shots extracted from "eye&eyebrowcoupled.avi". The right eyebrow has been extracted. We observe the evolution of the head rotation at the same time as the eyebrow moves upwards. The white line represents the arch extracted from the image processing analysis, the green line is the results from the projection of the neutral motion model after having applied on it the motion parameters. Ideally, both arches should have the same shape and location. The pupil tracking from the right eye is also plotted. The blue rectangles are the eye and eyebrow ROIs and the red squares are the blocks utilized during the pose tracking

VI.5 Analyzing Real-time Capabilities

The algorithms developed and tested in this thesis aim at providing solutions to deploy virtual teleconferencing systems. When studying possible new telecom applications, it becomes important to evaluate the potential of the system to run real-time.

The algorithmic analytical structure was kept simple and efficient to make real-time capabilities feasible. It has also been developed flexibly to enable the increase in the algorithmic complexity as the computing power available augments.

Two more reasons led us to keep computing complexity as a priority:

1. Kalman filtering for head tracking is a dynamic system highly dependent on the smoothness of the acquired video and the time involved in the feature 2D tracking. Although smoothness can be simulated by recording video at 30 f/s and analyzing the images afterwards at a lower rate, the filter characteristics are set so that its dynamics fit the speed of tracking. Analyzing very slowly would not simulate what would really happen to the filter in real life communications;
2. if very slow analysis algorithms had been implemented, no study about the rendering of faps could have been made because no subjective evaluation of the naturalness of the analysis results could have been made.

The purpose of this section is to detect which are the key parts of the system in terms of computational speed. We want to detect the bottleneck of the analysis-synthesis chain to establish the real-time viability of the proposed solution. We know that the algorithms are not environment-dependent because it has been determined as a premise; here we study if they are practically deployable.

The algorithms have successfully worked on line on the following computers:

– PC Intel Pentium III BiPro @ 700 MHz each, with acquisition card

– Laptop Intel 4 @ 1.6 GHz with FireWire camera

– PC Intel Pentium 4 @ 2.0 GHz with acquisition card

VI.5.1 Time performance evaluation of the algorithms

Let us review the relevance of each of the modules that compose our system. One module contains the video processing involved during the facial analysis, from the pose tracking to the expression analysis algorithmic implementation, plus the video rendering itself. The other module contains the synthetic rendering of the 3D head model after having been animated using the faps obtained from the analysis.

The main processes involved are sequential. One video frame is analyzed, the faps obtained are applied on the model; and then, the model is rendered. No new frame is analyzed until the synthetic rendering is finished; therefore, studying the frame rate evolution of the video becomes a simple way to evaluate the speed performance of each module.

Although the code utilized for the tests has not been optimized, the studies performed helped us to roughly determine the most important points to take into account for practical implementation. The block matching needed for head tracking was implemented to track blocks in parallel. The bi-processor computer utilized for the tests profited from this operation (see Table VI-9). The remaining processes were implemented sequentially.

The main goal of a correct facial video analysis for online applications is to implement algorithms that perform faster that the video acquisition frame rate. At the same time, we need to have a synthetic rendering engine that renders facial animation at least as fast as the video is been acquired, otherwise a slow down effect will appear on the synthesis.

Table VI-9

COMPUTER CHARACTERISTICS FOR THE TESTS

Processor:	PC Intel Pentium III BiPro @ 700 MHz each
Memory:	756 MB RAM
Acquisition Card:	Osprey 100 Video Capture Device
Video Card:	nVidia GeForce2 MX/MX 400
OS:	MS Windows 2000

The first step was to establish the influence of the rendering inside the complete system. The rendering speed of facial animation is inversely proportional to the number of vertices that compose the head model. Several studies about rendering performance of facial animation already exist (Breton, 2002). Our experiments were intended to find what was the model 'size' that would not interfere in the evaluation of the computational performance of the video analysis. We used different versions of the same model, each one being of different size.

Figure VI-22 plots the video frame rate versus the number of vertices of the model rendered. As expected, analyzing this graph, we observe that the larger the number of vertices, the slower the system becomes. When we reduce the number of vertices to increase the performance, we reach a point after which no speed improvement in achieved (~10000 vertices). At this point, we can affirm the following:

- The rendering interference into the final system is almost minimal and therefore it determines the proper conditions to evaluate the system facial analysis rate.
- The head model size is not the optimal option in terms of realistic animation as the visual aspect of the model becomes less pleasant as the number of vertices is reduced. Let us recall that face cloning requires very dense wire frames.

We chose the head model made of 2574 vertices for the facial expression tests. We considered it to be the best trade-off between having the fewest vertices and the best visual appearance.

In our second step, we divided the study of the video analysis block in two. First, we evaluated the influence of the pose-tracking algorithm (Figure VI-23) and then, we investigated the time performance of each feature analysis procedure (Figure VI-24). After examining the graphs, we concluded that the pose-tracking algorithms, and more concretely, the block matching required to track the face features on the video sequence cause the bottleneck in the processing speed. In fact, although parallel programming has been used to speed up the processing, this characteristic is scarcely exploited because our computer has only two processors. Ideally, dedicated implementations for block matching should be used to improve the pose tracking performance.

The influence of the feature analysis algorithms remains very marginal due to the time constraints imposed by the pose tracking. We would like to simply point out that the eye state analysis algorithm is computationally more demanding than the eyebrow motion analysis.

# vertices	f/s
500	2.130
750	2.080
1000	2.130
1500	2.130
2574	2.365
10000	2.365
150000	1.500
262656	1.250

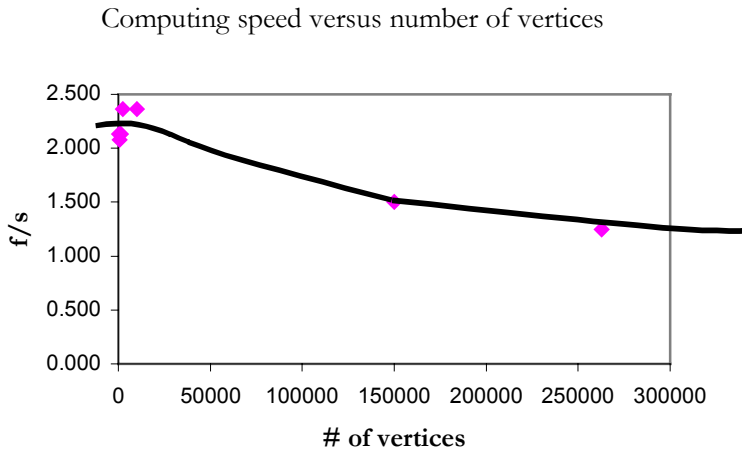


Figure VI-22. Evolution of the system speed versus the complexity of the head model being rendered. Kalman pose tracking was done with 10 features

# tracked points	f/s
5	4.300
6	4.200
7	3.360
8	3.160
9	2.720
10	2.557
13	1.940
17	1.600

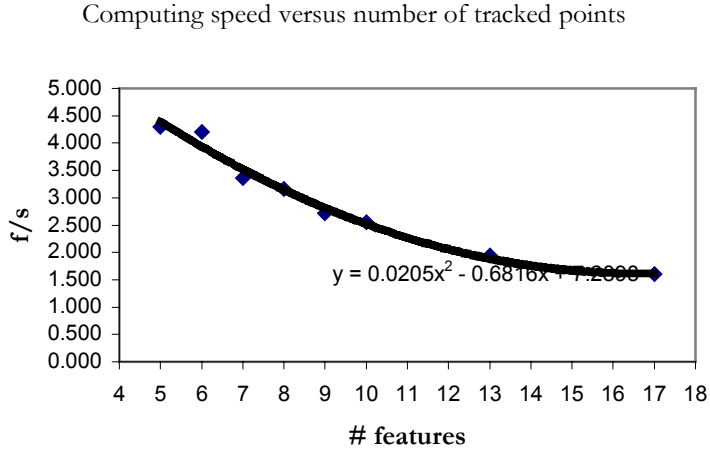


Figure VI-23. Evolution of the system speed versus the number of features utilized during the head tracking. The model used had 2574 vertices. No expression analysis was made

# analyzed features	f/s
1 eyebrow	2.460
2 eyebrows	2.460
2 eyebrows + 1 Eye	2.200
2 eyebrows + 2 Eyes	2.060

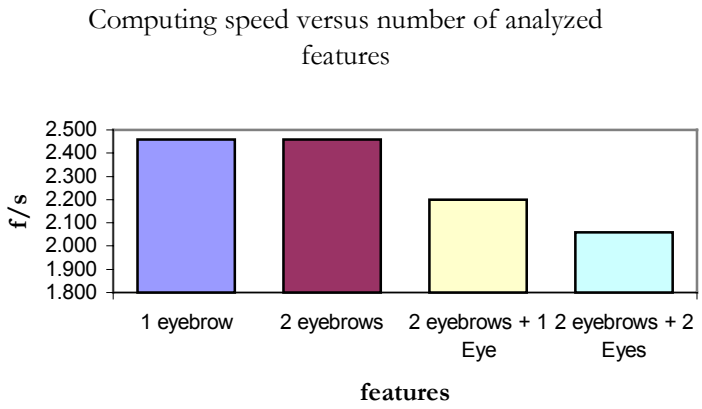


Figure VI-24. Evaluation of the computing speed cost of the expression analysis. Kalman pose tracking was done with 10 features and the model used had 2574 vertices

VI.6 Conclusions

We have deployed an application that has allowed us to evaluate the performance of the coupling between expression analyses and pose tracking. The main purpose of the algorithms developed for the analysis and the methodology used for the coupling is to come up with a global solution that could be as flexible as possible and usable in any circumstances.

Aiming at this goal, we presented our test-bed platform as a demo during ACM Multimedia 2002 (Andrés del Valle & Dugelay, 2002). Despite that the environmental conditions were unknown, the platform, which has been designed to work almost real-time, allowed us to verify that the algorithmic solution proposed could also work in another location rather than the laboratory.

Treating pose and expression separately has many advantages. We have taken the most out of them by designing specific facial feature analysis techniques that aim at extracting the most interesting motion information without building too complicated motion templates. We have proved that expression-pose coupling by extending the definition of motion templates in 3D is feasible and gives very good results. Replicating human facial motion from the analysis of visual data is one of the best ways to generate natural animation because it is a non-invasive method. The use of facial animation in communications that use virtual environments is necessary if we aim at creating a feeling of trust among speakers.

Test that measure the quality of the analysis for applications like ours in a quantitative way are difficult to set. Our tests were designed to give an objective perspective of the performance and the improvement possibilities of the algorithms presented.

From a subjective point of view, the test-bed platform allowed us to evaluate qualitatively our algorithms applied on eyes and eyebrows. The solution proposed has proved to be flexible, robust and it has the potential to be extended to the analysis of other features like the mouth, wrinkles, etc. The main conclusions from our tests were:

- The pose tracker used for the coupling is critical. The stability of the head pose tracking will directly influence the expression analysis results in the presented work frame.
- The theoretical restriction in the head rotation observed during the equation analysis, $\pi/2$, is increased to $\pi/4$ due a to the partial or complete hiding of the feature to be analyzed from the image plane.

Conclusions and Future Work

1 Conclusions from main contributions

Two premises have driven research in videoconferencing. Technically, more efficient ways of coding and transmitting video information are sought; socially, new communication frameworks where visual information increases speaker interaction are investigated. New telecommunications trends consider achieving these goals by using synthetic data. In virtual teleconferencing, which is the framework of our research, speakers are substituted by 3D synthetic models – clones (realistic) or avatars (symbolic). Regular video data is replaced by a limited number of action parameters determining facial motion, thus reducing the bandwidth required for transmission. Furthermore, when synthesizing the models in a common interactive environment a natural communication situation is recreated.

It is quite simple to identify an avatar; it is more difficult to determine when a realistic head representation of a person can be considered as a clone. Cloning someone's face is not simply a matter of using a visually realistic 3D-mesh representation of his head but also the acquired capability of replicating each one of his movements and expressions with detail. Unfortunately, it does not exist a general rule that establishes the difference between a 3D-head realistically animated and a clone. The difference may be a subjective issue and its determination may fall in the domain of psychology and behavior learning more than engineering. Nevertheless, it remains clear that the best way to create the animation for a clone is from the analysis of media obtained from recording the individual in daily life. From all the media available to generate facial animation (speech, magnetic captors, etc.) monocular image/video is the most intensely used because of its availability and its non-invasive nature.

When using facial animation in telecommunications we transmit facial animation parameters from an fap generator and we render them on the receiver. Both sender and receiver must share the same syntax and semantics for the faps in order to set coherent communications. Indeed, the coding techniques utilized during the signal processing must not alter the syntax or the semantics; otherwise it will disturb communications significantly. The loss of data or interference in fap streams might have much worse

effect than current video disruption due to frame dropping. The MPEG-4 standard has established some specific decoding issues to be used for facial animation. The standard specifies common syntax to describe face behavior, thus permitting interoperability among different face animation systems. At this point of the evolution and deployment of applications compliant with MPEG-4 several concerns have appeared: Has the standard given a global solution that all specific face animation systems can adopt? Or, does the syntax restrict the semantics of the possible achievable motion too much? No matter the answer, the existence of all these doubts shows that there is still a long way to go to master face animation and more concretely, the automatic generation of realistic human-like face motion.

The understanding of facial nonverbal behavior, especially from eyes and eyebrows, from monocular images becomes critical to generate natural and coherent facial animation on synthetic head models when using classical teleconferencing input (monocular video). Specifically, facial expression analysis on monocular images has become a major key point to tackle in the following fields:

- Computer Graphics (CG): to create realistic animations;
- Image Processing (IP): for model-based coding;
- Computer Vision (CV): for expression analysis in image reproduction;
- Human-Computer Interaction (HCI): to make machines react to human behavior.

The different approaches taken to perform expression analysis in each field depends on two factors. First, it depends on the amount of detailed motion data needed; for instance, more motion information is needed in CG than in HCI. Second, the methods differ based on the level of understanding required in their applications; in HCI we need to comprehend what kind of action has happened, by recognizing a feeling for example, whereas in CG or IP, we only require replication. In all cases, it becomes crucial to control the influence of the pose on the final expression that appears in the face on the image.

Developing a video analysis scheme where head pose tracking and face feature expression analysis are treated separately permits to design specialized image analysis algorithms adjusted to specific needs, feature characteristics, etc. Algorithms that are universally useable generally lack precision. Indeed, if no previous assumptions are taken then, making suitable the analysis to all cases implies lots of computations and therefore the loss of real-time possibilities. To compensate this restriction, we may generate less precise analysis algorithms (using simple motion models) but keeping in mind the possibility of improving the complexity of the system; as the computational requirements

become less and less restrictive, a flexible analysis scheme will allow us to increase the complexity and to extract more detailed motion data.

This thesis has presented a system that aims at analyzing facial motion and expressions following the aforementioned ideas. It profits as much as possible from intra-feature constraints (like natural eye motion) and inter-feature constraints (specific eyelid motion from eyebrow analysis). The algorithms have first been defined for a frontal position and then, they have been extended to work integrated into a pose-tracking system. The obtained experimental results are very positive and encourage the authors to keep developing the same strategy to analyze other facial features: mouth, wrinkles, ... whose analysis and modeling seems a-priori more complex. The inter-feature information will then be enriched and the obtained motion information more accurate.

The use of natural intra-feature constraints has allowed us to develop a gaze tracking algorithm capable of deducing eyelid motion from the understanding of pupil actions. Muscular-based intra-feature restrictions have set the basis for the mathematical template models that analyze eyebrow and mouth movements.

The correlation that exists among facial features is exploited to set inter-feature constraints that help to complement the analysis of these features. We have developed and tested algorithms that used inter-feature constraints. For instance, we improve eyelid motion extraction by adding eyebrow behavior information to the *eye-state tracking*-algorithm we have designed.

The facial feature motion analysis templates need to be adapted to extend their use from simply analyzing a frontal view of a face to study facial motion from a head showing a different pose on the image. Our approach to perform this algorithmic extension has been to redefine the designed motion templates in 3D space, over a realistic head representation of the speaker that presents its neutral position (facing the camera). Thanks to the rigid-motion data provided by a pose tracking algorithm that uses an extended Kalman filter, we were able to relate the 2D information extracted from each video frame to the 3D description of the motion template and vice versa. The analysis process can be summarized as follows:

- All data from the motion template needed to make the analysis, for instance the feature ROI, are projected from 3D onto the video frame.
- Using image-processing techniques specific to each feature and already tested for a frontal perspective, we extract the data required to interpret the movement.
- Using the pose information, we undo the projection and the pose and we interpret these data on the 3D motion templates redefined on a frontal head-model.

Undoing the projection and the pose is not a simple task. It is impossible to derive 3D information from simple 2D data. This is the reason why it is necessary to model the surfaces upon which the motion templates will be redefined. To keep the analysis simple and the extension from 2D to 3D of the motion templates straightforward, we approximate the feature surface by the plane that best fits the feature.

The use of a pose-tracking algorithm based on Kalman filtering introduces some noise in the extracted data. Nevertheless, its influence in the final results is minimal as long as the tracking is not lost. Our solution is able to analyze facial features and to track the head simultaneously for almost any pose not restricted by the theoretical limitations of the system (rotations greater than $\pm\pi/2$ rad). The evolution of the ROI projection helps to control the performance of the analysis. After having implemented a practical scenario to test our algorithms, we have noticed that the algorithm limitation increases to $\pm\pi/4$ rad because the analysis performance is controlled by the visual representation of the feature. Beyond this range, facial features are not completely present in the image. Our research has proved that this technique is flexible, robust and has the potential to be exploited for the analysis of other facial features.

2 Future work

The proposed facial motion and expression analysis framework has been tested on the most active features; it can also be easily extended to analyze any other part on the face, for example, wrinkles and furrows. A future challenge will appear at the moment when more facial features start being coupled with the proposed Kalman-based pose tracking system. As more features are susceptible of moving, less fixed facial tracking-points will be available for the head tracking algorithm. At that point, studies about the robustness of the pose tracking versus the number of features and the freedom of movement for the speaker will have to be done. Complementary solutions might be used. We will cite, for example, the insertion of complete visual feedback from the clone, that is, getting not only visual feedback from the rigid-motion but also from the facial expression as a complement for the Kalman-based pose-tracker. We might even search for a possible head tracker substitute that relying on the same geometrical characteristics could also work in the present environment.

Motion template adaptation to 3D space has been done with the approximation of each feature surface by a plane. This was a simple, straightfoward and robust solution but it introduced some inaccuracies in the results. To reduce the imprecision that this approximation creates, templates could be redefined on surfaces that shape better the area that will be analyzed. Regardless of the surface used, we should keep in mind that the utilization of monocular video is by itself a restriction. Features might be partially or completely occluded at some point of the analysis. To compensate for this lack of visual information, we could reinforce symmetry constraints to generate motion data of the missing features; we could use the analyzed parameters obtained from their symmetric counterparts.

The system presented in this thesis is the integration of several modules that can work independently from each other. We can profit from this fact by reutilizing the analysis modules separately in different contexts and applications. For instance, the proposed *eye-state tracking*-algorithm could be used on high-speed video recordings of eye sequences from medical patients in the search for correlated patterns of brain activity.

Although the technique herein described has been addressed as a solution to analyze facial expressions to obtain animation parameters for synthetic facial motion, it can also be extended to other scientific fields where the knowledge of the instant actions of the person in front of the camera is desired, for instance, in Human Computer Interaction analysis (Andrés del Valle & Dugelay, 2003). People can understand facial action even when faces are under very bad lighting or in the presence of disturbing objects over them. This is basically due to the fact that humans are able to automatically

reduce the complexity of the analysis into different parts and to do this analysis progressively. First, we examine the conditions under which the face is and we decide if further understanding is possible; then, we locate the head and get its rigid motion (its pose) and finally, we pay attention to the different details of the face that are interesting to us because they contain expression information. When humans are not able to perform an exhaustive analysis (lighting is very bad, or a significant part of the face is occluded), they make up for the missing information (generally assuming standard human behavior) or they simply accept that they cannot understand the face motion they are observing. The presented framework is designed to perform facial motion and expression analysis on monocular images trying to replicate this natural and intuitive human behavior.

The interest in facial motion understanding is increasing. New research in this field is in its way. The new European Network of Excellence (NoE) Similar (2003) has joined the effort of several institutions that aim at developing tools like the virtual teleconferencing system aimed by our research. Among other activities, this NoE will be developing applications based on techniques similar to the one presented here, will also do research to improve current algorithms and will set the bases for a global European network in multimodal human-computer interaction.

3 Publications derived from this research

Book chapters

Techniques for Face Motion & Expression Analysis on Monocular Images
Ana C. Andrés del Valle and Jean-Luc Dugelay
To appear in N. Sarris & M. G. Strintzis (Eds.) "3D Modeling and Animation: Synthesis and Analysis Techniques for the Human Body"
Idea Group Publishers.

Journals

Efficient Ocular Expression Analysis for Synthetic Reproduction
A. C. Andrés del Valle and Jean-Luc Dugelay
Submitted to
IEEE Transactions on Multimedia.

Analysis and Reproduction of Facial Expressions for Realistic Clones
Stéphane Valente, Ana C. Andrés del Valle and Jean-Luc Dugelay
The Journal of VLSI Signal Processing
August 2001 Vol. 29 Issue:1/2 pp:41-49.

Conferences

Making Machines Understand Facial Motion Like Humans Do
A. C. Andrés del Valle & J.-L. Dugelay
Human Computer Interaction International, HCI International 2003
June 21st-23rd, 2003 Crete - Greece

Eyebrow Movement Analysis over Real-time Video Sequences for Synthetic Representation
A. C. Andrés del Valle & J.-L. Dugelay
AMDO 2002
November 21st-23rd, 2002 Palma de Mallorca - Spain

Facial Expression Analysis Robust to 3D Head Pose Motion
A. C. Andrés del Valle & J.-L. Dugelay
ICME 2002
August 16th-29th, 2002 Lausanne - Switzerland

Eye State Tracking for Face Cloning
A. C. Andrés del Valle & J.-L. Dugelay
ICIP 2001
October 7th - 10th, 2001 Thessaloniki- Greece

Analysis-Synthesis Cooperation for MPEG-4 Realistic Clone Animation
J.-L. Dugelay & A. C. Andrés del Valle
Euroimage ICAV3D 2001
May 30th - June 1st, 2001 Mykonos - Greece

Acquisition et Animation de Clones Réalistes pour les Télécommunications
A. C. Andrés del Valle & J.-L. Dugelay & E. Garcia & S. Valente
Compression et Représentation des Signaux Audiovisuels, CORESA 2000
October 19, 2000 Poitiers - France

Tutorials

Human movement. Face recognition and animation
«Facial Animation. Analysis and Synthesis Methods to Replicate Human Communications»
A. C. Andrés del Valle & R. Mas & F. J. Perales
Second International Workshop on Articulated Motion and Deformable Objects, AMDO 2002
November 2002 Palma de Mallorca - Spain

Technical Demos

Online Face Analysis: Coupling Head Pose-Tracking with Face Expression Analysis
A. C. Andrés del Valle & J.-L. Dugelay
ACM Multimedia
December 2002 Juan-Les-Pins - France

Technical Reports

Pose Coupling with Eye Movement Tracking
A. C. Andrés del Valle & J.-L. Dugelay
February 2002 Eurecom - France
RR-01-08

A Video Conference System under MPEG-4; Overview of Face Animation in MPEG-4; and Study of the Compliance Level of Eurecom's Face Animation-Teleconferencing System
A. C. Andrés del Valle & J.-L. Dugelay & D. Pelé
March 2001 Eurecom/France Telecom - France
RR-2001-053; FT/BD/DIH/HDM/11/DP

Appendices

Appendix I-A

Camera Calibration

During camera calibration we must link the real world reference frame to the image reference frame in order to find the relationship between the coordinates of the points in 3D space and the coordinates of the same points in the image. To do so, we introduce the camera reference frame because there is no direct relation between the previously mentioned reference frames. Then, we can find an equation linking the camera reference frame with the image reference frame (LinkI), and another equation linking the world reference frame with the camera reference frame (LinkE). Identifying LinkI and LinkE is equivalent to finding the camera's characteristics, also known as the camera's extrinsic and intrinsic parameters. We generally define these parameters as follows (Trucco & Verri, 1998):

Extrinsic parameters are the parameters that define the location and orientation of the camera reference frame with respect to a known world reference frame.

Intrinsic parameters are the parameters necessary to link the pixel coordinates of an image point with the corresponding point in the camera reference frame.

There exist many calibration techniques that have been reported in the past two decades. The developed methods can be roughly classified into two groups: photogrammetic calibration and self-calibration.

Photogrammetic calibration: this type of calibration is performed by observing an object whose geometry in 3D-space is known with very good precision. The calibration object usually consists of two or three planes orthogonal to each other. Sometimes, a plane undergoing a precisely known translation is also used. This type of approach requires an elaborate setup, but can be done very efficiently.

Self-calibration: this method does not use a calibration object. By moving a camera in a static scene, the rigidity of the scene already provides two constraints of the camera's parameters from one camera displacement by using image information alone. Three images taken by the same camera with fixed intrinsic parameters are sufficient to recover both intrinsic and extrinsic parameters. This approach is very flexible, however, it is not as accurate as the photogrammetic one.

Appendix I-B

Illumination Models

There are two major categories of reflected light:

(i) Diffuse Reradiation (scattering): this occurs when the incident light penetrates the surface and is reflected equally in all directions. The light interacts strongly with the surface, so its color is affected by the surface color. This kind of illumination behavior predominates on unpolished surfaces.

(ii) Specular Reflection: light does not penetrate the object, but it is instead directly reflected from its outer surface. It makes the object look shinny and it has the same color as the light source. Mirrors are totally specular.

The total light reflected in a certain direction is the sum of the diffuse and the specular components in that direction.

The intensity of the pixels that we get from the image of the face is the result of the light from the recorded scene (i.e. the face) scattered towards the camera lens. The nature of the reflection phenomenon requires the knowledge of some vector magnitudes (Figure App-1):

- the normal $\vec{n}$ to the surface at the point p being studied;
- the vector $\vec{v}$ from p to the camera; and
- the vector $\vec{s}$ from p to the light source.

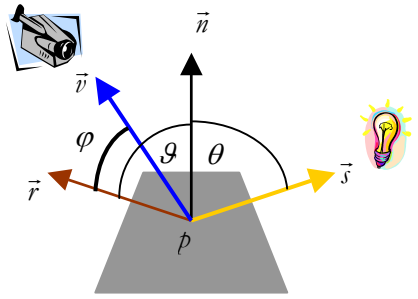


Figure App-1. The reflected light that reaches the camera lens depends on the direction of the normal to the surface ($\vec{n}$) the vector from the studied point to the light source ($\vec{s}$) and the vector from the point to the camera lens ($\vec{v}$). $\vartheta = \varphi$ for perfectly specular reflections.

A fairly extended approach to appreciate the result of lighting on faces in to analyze illumination by trying to synthetically reproduce it on the realistic 3D-model of the user's head. Phong's reflection model is the 3D shading model most heavily used to assign shades to each individual pixel of the synthetic face. It is characterized by simplifying second order reflections introducing an ambient reflection term that

f

simulates the sparse (diffuse) reflection coming from sources whose light has been so dispersed that it is very difficult to determine its origin. In a more or less simplified way we can understand the final intensity of each of the pixels as:

(0-1)
$$I = I_a r_a + \sum_{l \in L} f_{att}^{l} I_s^{l} \left\lfloor r_d \left(\vec{s}_l \bullet \vec{n}\right) + r_s \left(\vec{r}_l \bullet \vec{v}\right)^f \right\rfloor,$$

where each addition term represents the intensity contribution of the each light source (*l*) being reflected (*L*) by the surface:

$I_a r_a$: is the scalar product of the intensity of the ambient light and the ambient reflection coefficient for the surface; it is a single value independent of other light sources.

f_{att}^{l} : is the light attenuation coefficient. Energy from a point light source reaching a piece of a surface falls off as the inverse square of the distance the light has traveled (*dl*).

(0-2)
$$I_{s,att} = \frac{I_s}{d_l^2}.$$

Real world light sources are not points. Generally, the attenuation is then approximated by:

(0-3)
$$f_{att} \cong \min\left(\frac{1}{c_1 + c_2 d_l + c_3 d_l^2}, 1\right),$$

where c_1, c_2 and c_3 and some pre-established model values.

I_s^l : is the intensity magnitude of the light (*l*) of source *s*.

$r_d\left(\vec{s}_l \bullet \vec{n}\right)$: represents the contribution of the diffuse reradiation. $I_s^l r_d\left(\vec{s}_l \bullet \vec{n}\right)$, also known as *Lambert's Law*, states that if a surface is turned away from a light by some angle θ, the area subtended by the light is $\cos(\theta)$ less than before, which implies that the brightness of the light source decreases by the same amount. $\cos(\theta)$ is $\vec{s}_l \bullet \vec{n}$ (normalized). r_d is the diffuse reflection coefficient for the surface.

$r_s\left(\vec{r}_l \bullet \vec{v}\right)^f$: represents the contribution due to the specular reflection. *f* is the parameter that controls the shininess of the surface. Larger *f* values represent shinier surfaces, and will lead to smaller specular highlights. This is a simplified expression that tries to model the fact that the amount of light goes down as the angle φ between $\vec{r}_l$ and $\vec{v}$ goes up. The actual expression of φ is very complex.

Appendix I-C

Morphological Mathematics: the Watershed Transformation

Here, we will discuss the most extensively used algorithm for segmentation that is based on mathematical morphology: *watershed*, so to get an idea of the strength of the math tools that are proposed.

The watershed transformation

Principle

Any grey tone image can be considered as a topographic surface. If we flood this surface from its minima and, if we prevent the merging of the waters coming from different sources, we partition the image into two different sets: the catchment basins and the watershed lines.

(i) If we apply this transformation to the image *gradient*, the catchment basins should theoretically correspond to the homogeneous grey level regions of this image.

However, in practice, this transform produces an important over-segmentation due to noise or local irregularities in the gradient image.

Marker-controlled watershed

A major enhancement of the watershed transformation consists in flooding the topographic surface from a previously defined set of markers. Doing so, we prevent any over-segmentation.

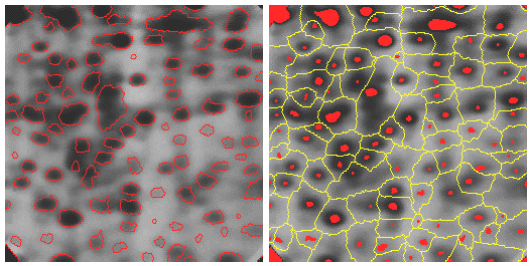


Figure App-2. Markers of the blobs and of the background and marker-controlled watershed of the gradient image.

The segmentation paradigm

Segmenting an image by the watershed transformation is therefore a two-step process:

h

1) Finding the markers and the segmentation criterion (the criterion or function which will be used to split the regions - it is most often the contrast or gradient, but not necessarily).

2) Performing a marker-controlled watershed with these two elements.

The difficulty of the technique lies on how to determine the image characteristic that will permit an automatic marking process. In the case illustrated in Figure App-3, where coffee beans are detected, the criterion used to mark the correct area is the distance function to the initial image.

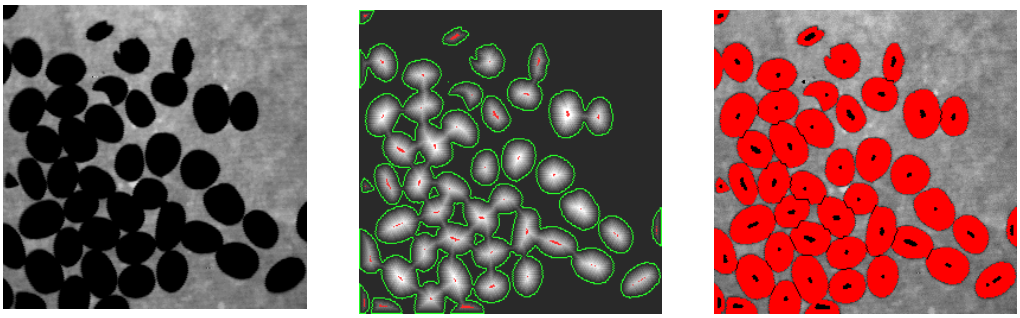


Figure App-3. To count the coffee beans the watershed marking decision follows a criterion based on the distance function to the initial image.

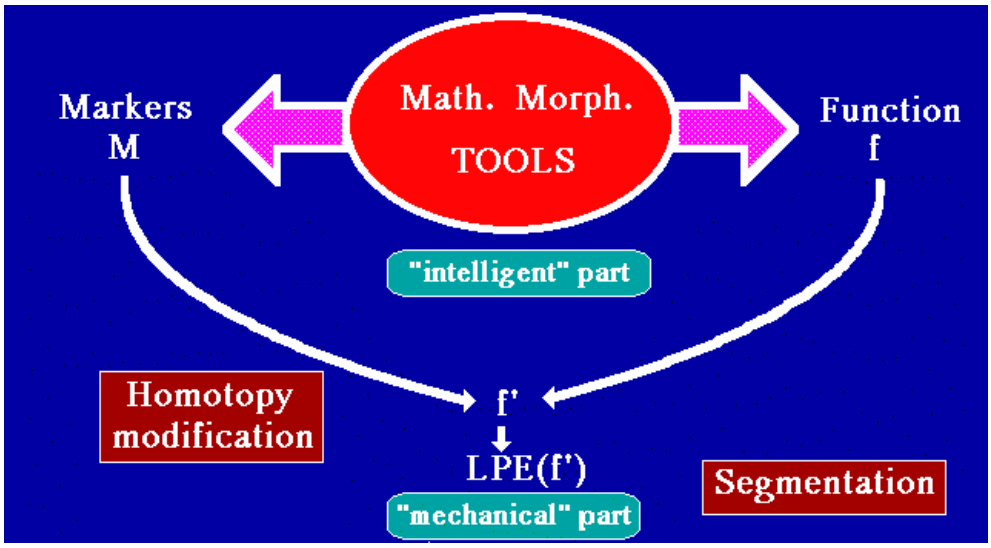


Figure App-4. This graph is a conceptual description of the complete segmentation process. The mathematical morphology tool provide the basis for the processing but there must exist and 'intelligent' decision part to evaluate the coherence and lead the analysis

Hierarchical segmentation

The watershed transformation can also be used to define a hierarchy among the catchment basins. Starting from the initial watershed transformation of the gradient image, a mosaic image can be defined, and then its associated gradient.

From this image, a new criterion function is built (based on the relative heights of the walls separating the initial catchment basins). The watershed transformation applied to this image provides a higher level of hierarchy in the segmented image (thus suppressing much of the over-segmentation).

Many other techniques and tools can be used to define a hierarchy on an image. Most of them are based on a flooding process.

Feature-based face motion analysis can profit from mathematical morphology tools. Like we see in (Ravyse, Sahli, Reinders, & Cornelis, 2000), where the authors do their eye gesture analysis with a mathematical morphology scale-space approach, forming spatio-temporal curves out of scale measurements statistics.

Figure App-5. Top: Initial image (left) and initial watershed of the gradient (right). Bottom: Mosaic image (left) and first level of hierarchy (right).

Appendix I-D

Estimators

Linear

Let us call $\vec{\lambda}$ the vector of parameters obtained from the image analysis and $\vec{\mu}$ the vector of FA parameters for the synthesis observed by $\vec{\lambda}$. The usual way to construct the linear estimator L, which best satisfies $\vec{\mu} = L \cdot \vec{\lambda}$ on the training database, is to find a solution in the least square sense. We verify that this linear estimator is given by

(0-4) $$L = \mathrm{M}\Lambda^T(\Lambda\Lambda^T)^{-1}$$

where $\mathrm{M} = [\vec{\mu}_1 | \ldots | \vec{\mu}_d]$ and $\Lambda = [\vec{\lambda}_1 | \ldots | \vec{\lambda}_d]$ are the matrices obtained by concatenating all $\vec{\mu}$ and $\vec{\lambda}$ vectors from the training set.

*Neural networks**

Neural networks are algorithms inspired on the processing structures of the brain. They allow computers to learn a task from examples. Neural networks are typically organized in layers. Layers are made up of a number of interconnected "nodes" which contain an "activation function", see Figure App-6.

Most artificial neural networks, or ANNs, contain some form of 'learning rule' that modifies the weights of the connections according to the input patterns that it is presented with.

The most extensively used rule is the *delta rule*. It is often utilized by the most common class of ANNs called 'backpropagational neural networks' (BPNNs). Backpropagation is an abbreviation for the backwards propagation of error.

With the delta rule, as with other types of backpropagation, 'learning' is a supervised process that occurs with each cycle or 'epoch' (i.e. each time the network is presented with a new input pattern) through a forward activation flow of outputs, and the backwards error propagation of weight adjustments. More simply, when a neural network is initially presented with a pattern it makes a random 'guess' as to what it might be. It then sees how far its answer was from the actual one and makes an appropriate adjustment to its connection weights. More graphically, the process looks like Figure App-7.

---

* Information partially taken from the GMSlab – university of University of Illinois Urbana - Champaign images are copyrighted.

Backpropagation performs a gradient descent within the solution's vector space towards a 'global minimum' along the steepest vector of the error surface. The global minimum is that theoretical solution with the lowest possible error. The error surface itself is a hyperparaboloid but is seldom 'smooth' as is depicted in Figure App-8. Indeed, in most problems, the solution space is quite irregular with numerous 'pits' and 'hills', which may cause the network to settle down in a 'local minimum', which is not the best overall solution.

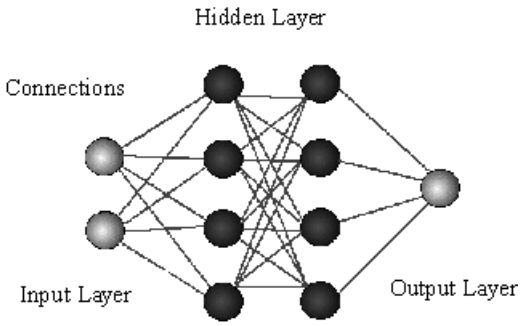


Figure App-6. Patterns are presented to the network via the 'input layer', which communicates to one or more 'hidden layers' where the actual processing is done via a system of weighted 'connections'. The hidden layers then link to an 'output layer' where the answer is output as shown in the graphic below.

Since the nature of the error space cannot be known *a priori,* neural network analysis often requires a large number of individual runs to determine the best solution. Most learning rules have built-in mathematical terms to assist in this process, which control the 'speed' (Beta-coefficient) and the 'momentum' of the learning. The speed of learning is actually the rate of convergence between the current solution and the global minimum. Momentum helps the network to overcome obstacles (local minima) in the error surface and settle down at or near the global minimum.

Once a neural network is 'trained' to a satisfactory level it may be used as an analytical tool on other data. To do this, the user no longer specifies any training runs and instead allows the network to work in forward propagation mode only. New inputs are presented to the input pattern where they filter into and are processed by the middle layers as though training were taking place, however, at this point the output is retained and no back propagation occurs. The output of a forward propagation run is the predicted model for the data, which can then be used for further analysis and interpretation.

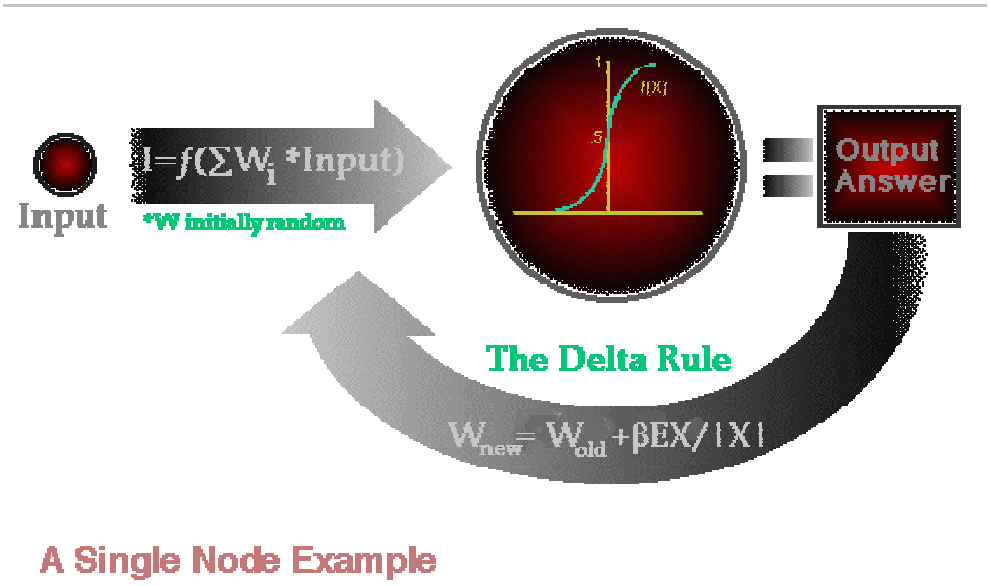


Figure App-7. Note that within each hidden layer node is a sigmoidal activation function that polarizes network activity and helps stabilize it.

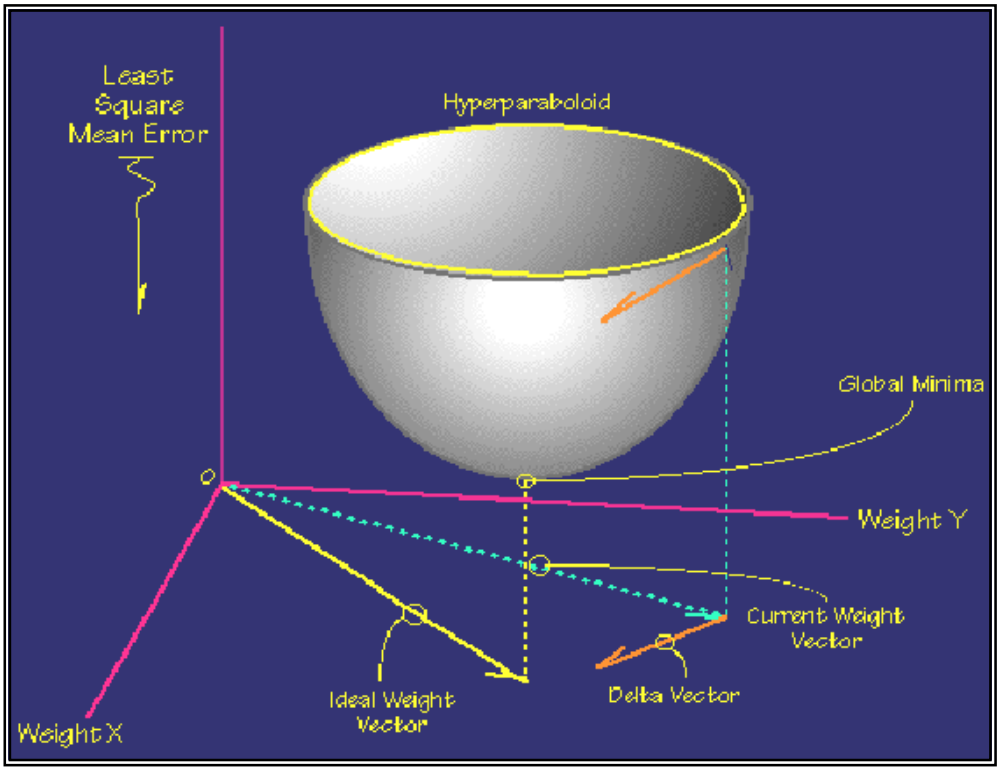


Figure App-8. Graphical interpretation of the search for a minimum

Appendix I-E

Fuzzy Logic

It is important to point out the distinction between fuzzy systems and probability. Both operate over the same numeric range, and have similar values: 0.0 representing False (or not membership), and 1.0 representing True (or membership). However we must differentiate that probability establishes the chances for a statement to be true whereas fuzzy logic describes the degree of concreteness of the statement itself. For instance, the sentence: "There is an 80% chance that Pablo is tall" corresponds in fuzzy terminology to "Pablo's degree of membership within the set of tall people is 0.80". The semantic difference is significant: the first view supposes that Pablo is either tall or not tall, and that we only have 80% chance of knowing which set he is in. Fuzzy terminology supposes that Pablo is *more or less* tall, or some term corresponding to the value of 0.80.

Fuzzy systems try to gather mathematical tools to represent natural language, where the concepts of Truth and False are too extreme and intermediate or more *vague* interpretations are needed. Let us state some formal definitions:

1. Let X be a set of objects, with elements noted as x. Thus X={x}.
2. A fuzzy set A in X is characterized by a membership function mA(x) which maps each point in X onto the real interval [0.0, 1.0]. As mA(x) approaches 1.0, the "grade of membership" of x in A increases.
3. A is EMPTY iff for all x: mA(x)=0.0.
4. A=B iff for all x: mA(x)=mB(x) [or, mA=mB].
5. mA'=1-mA.
6. A is CONTAINED in B iff mA<=mB.
7. C=A UNION B, where: mC(x)=MAX(mA(x), mB(x)).
8. C=A INTERSECTION B where: mC(x)=MIN(mA(x), mB(x)).

Besides the basic operations amongst sets, fuzzy systems permit the definition of "hedges", or modifiers of fuzzy values. These operations are provided in an effort to maintain close ties to natural language, and to allow for the generation of fuzzy statements through mathematical calculations. As such, the initial definition of hedges and operations upon them is quite a subjective process and may vary from one application to another. Hedges mathematically model the concepts of *very*, *somewhat*, *sort of*, and so on. For instance, m"very"A(x) = $mA(x)^2$.

Appendix I-F

Hidden Markov Models

The problem arises when wanting to probabilistically model a specific problem and we only count on the outputs to do so. To take advantage of HMM we can do in two ways:

- **From outputs to states**

We want to determine the set of internal states that most likely gave rise to a particular sequence of outputs. The Viterbi algorithm is the method used for solving this problem. It is clear that for any sequence of outputs and states the probabilistic weighting can be calculated with no much difficulty. But for a given long sequence of outputs, there is an immense number of possible sequences of states to choose in order to find the most probable. The Viterbi algorithm helps to ease this search.

If we consider the possibilities for the first n states, we retain not just the set of states with the highest weight, but also the set of states with the highest weight for all other possibilities for the state at time n in addition to the one in the set of states with overall highest weight.

To obtain the set of states with overall highest weight for the n+1 state, and also the set of states with the highest weight for any possible state at time n+1, we only need to consider possibilities involving the sets of states from time 1 to time n that we previously retained.

- **From outputs to model**

This is the most complicated of the problems. We assume that the model has one or more variable parameters in its description, and we are looking for the values of those parameters that would make an observed sequence of outputs the most likely.

Two major methods are used. One, the *segmental K-means* method, obtains an initial approximation to the model, and involves assuming that a particular set of states accompanies the known outputs. The other, the *Baum-Welch estimation* algorithm, is used to obtain the best fit of the model to the output sequence considering all possible sequences of states that could have produced the known output.

s

Appendix IV-G

ORIGINAL DATA FROM APPLYING THE PUPIL-SEARCH ALGORITHM ON VIDEO SEQUENCE "NEON"

numframes	In .avi	diff X	diff Y	diff X	diff Y		x:3%	y:3%	TOTAL		x:5%	y:5%	TOTAL		x:10%	y:10%	TOTAL
6	7.944724	580.8538	7.769349	26.99548	9.353121	7.93	0	0	0	13.22	0	0	0	26.44	0	1	0
7	9.268844	580.8538	7.769349	26.99354	9.353909	5.16	0	0	0	8.6	0	0	0	17.2	0	1	0
8	10.59296	580.8538	7.769349	9.89694	0.747989		0	1	0		1	1	1		1	1	1
9	11.91709	580.8538	7.769349	9.899069	0.749304		0	1	0		1	1	1		1	1	1
10	13.24121	580.8538	7.769349	18.45029	9.301452		0	0	0		0	0	0		1	1	1
11	14.56533	580.8538	7.769349	9.914283	0.676617		0	1	0		1	1	1		1	1	1
12	15.88945	580.8538	7.769349	1.364747	0.646249		1	1	1		1	1	1		1	1	1
13	17.21357	580.8538	7.769349	9.93641	0.509561		0	1	0		1	1	1		1	1	1
14	18.53769	580.8538	7.769349	1.376779	0.45848		1	1	1		1	1	1		1	1	1
15	19.86181	580.8538	7.769349	1.371219	0.253243		1	1	1		1	1	1		1	1	1
16	21.18593	580.8538	7.769349	1.346299	0.214834		1	1	1		1	1	1		1	1	1
17	22.51005	580.8538	7.769349	1.297352	-0.002378		1	1	1		1	1	1		1	1	1
18	23.83417	580.8538	7.769349	1.250614	-0.008007		1	1	1		1	1	1		1	1	1
19	25.15829	580.8538	7.769349	1.15459	-0.202894		1	1	1		1	1	1		1	1	1
20	26.48241	580.8538	7.769349	1.095796	-0.175404		1	1	1		1	1	1		1	1	1
21	27.80653	580.8538	7.769349	0.963745	-0.342418		1	1	1		1	1	1		1	1	1
22	29.13065	580.8538	7.769349	0.907263	-0.290717		1	1	1		1	1	1		1	1	1
23	30.45477	580.8538	7.769349	0.757352	-0.433737		1	1	1		1	1	1		1	1	1
24	31.77889	580.8538	7.769349	17.82008	-0.374734		0	1	0		0	1	0		1	1	1
25	33.10302	580.8538	7.769349	0.56489	-0.49172		1	1	1		1	1	1		1	1	1
26	34.42714	580.8538	7.769349	0.544626	-0.418457		1	1	1		1	1	1		1	1	1
27	35.75126	580.8538	7.769349	0.403194	-0.528412		1	1	1		1	1	1		1	1	1
28	37.07538	580.8538	7.769349	17.50088	-0.460029		0	1	0		0	1	0		1	1	1
29	38.3995	580.8538	7.769349	0.276409	-0.551787		1	1	1		1	1	1		1	1	1
30	39.72362	580.8538	7.769349	0.295355	-0.476455		1	1	1		1	1	1		1	1	1
31	41.04774	580.8538	7.769349	0.18091	-0.566743		1	1	1		1	1	1		1	1	1

t

numframes	In .avi	diff X	diff Y	diff X	diff Y	x:3%	y:3%	TOTAL	x:5%	y:5%	TOTAL	x:10%	y:10%	TOTAL
32	42.37186	580.8538	7.769349	0.213086	-0.493017	1	1	1	1	1	1	1	1	1
33	43.69598	580.8538	7.769349	17.20021	-0.583279	0	1	0	0	1	0	1	1	1
34	45.0201	580.8538	7.769349	0.151602	-0.50462	1	1	1	1	1	1	1	1	1
35	46.34422	580.8538	7.769349	0.058549	-0.581337	1	1	1	1	1	1	1	1	1
36	47.66834	580.8538	7.769349	0.105501	-0.512542	1	1	1	1	1	1	1	1	1
37	48.99246	580.8538	7.769349	0.000773	-9.131303	1	0	0	1	0	0	1	1	1
38	50.31658	580.8538	7.769349	0.05233	-9.0652	1	0	0	1	0	0	1	1	1
39	51.6407	580.8538	7.769349	17.07796	-0.590248	0	1	0	0	1	0	1	1	1
40	52.96482	580.8538	7.769349	-0.072344	-0.631207	1	1	1	1	1	1	1	1	1
41	54.28894	580.8538	7.769349	-0.009389	-0.557873	1	1	1	1	1	1	1	1	1
42	55.61307	580.8538	7.769349	0.039908	-0.501299	1	1	1	1	1	1	1	1	1
43	56.93719	580.8538	7.769349	-0.032016	-0.558617	1	1	1	1	1	1	1	1	1
44	58.26131	580.8538	7.769349	-0.091558	-0.601145	1	1	1	1	1	1	1	1	1
45	59.58543	580.8538	7.769349	-0.032189	-0.537477	1	1	1	1	1	1	1	1	1
46	60.90955	580.8538	7.769349	-0.091096	-0.577378	1	1	1	1	1	1	1	1	1
47	62.23367	580.8538	7.769349	-0.053181	-9.06526	1	0	0	1	0	0	1	1	1
48	63.55779	580.8538	7.769349	-0.096484	-0.577702	1	1	1	1	1	1	1	1	1
49	64.88191	580.8538	7.769349	-0.065513	-9.100716	1	0	0	1	0	0	1	1	1
50	66.20603	580.8538	7.769349	0.008376	7.995366	1	0	0	1	1	1	1	1	1
51	67.53015	580.8538	7.769349	-17.12191	16.51727	0	0	0	0	0	0	1	1	1
52	68.85427	580.8538	7.769349	-0.103768	16.43192	1	0	0	1	0	0	1	1	1
53	70.17839	580.8538	7.769349	8.472639	7.910507	0	0	0	1	1	1	1	1	1
54	71.50251	580.8538	7.769349	-0.142118	-0.662208	1	1	1	1	1	1	1	1	1
55	72.82663	580.8538	7.769349	-8.657852	-17.68946	0	0	0	1	0	0	1	0	0
56	74.15075	580.8538	7.769349	-0.159478	-9.189696	1	0	0	1	0	0	1	1	1
57	75.47487	580.8538	7.769349	-0.104706	-9.139874	1	0	0	1	0	0	1	1	1
58	76.79899	580.8538	7.769349	-0.159162	-9.152743	1	0	0	1	0	0	1	1	1
59	78.12312	580.8538	7.769349	-0.105999	-9.107118	1	0	0	1	0	0	1	1	1
60	79.44724	580.8538	7.769349	-8.697029	-9.107441	0	0	0	1	0	0	1	1	1

u

numframes	In .avi	diff X	diff Y	diff X	diff Y	x:3%	y:3%	TOTAL	x:5%	y:5%	TOTAL	x:10%	y:10%	TOTAL
61	80.77136	580.8538	7.769349	-0.107778	-9.089707	1	0	0	1	0	0	1	1	1
62	82.09548	580.8538	7.769349	-0.157711	-9.090764	1	0	0	1	0	0	1	1	1
63	83.4196	580.8538	7.769349	-0.106733	-9.052756	1	0	0	1	0	0	1	1	1
64	84.74372	580.8538	7.769349	-0.16917	-17.60003	1	0	0	1	0	0	1	0	0
65	86.06784	580.8538	7.769349	8.42004	-17.57276	0	0	0	1	0	0	1	0	0
66	87.39196	580.8538	7.769349	8.375445	-17.56245	0	0	0	1	0	0	1	0	0
67	88.71608	580.8538	7.769349	-0.113642	-17.52731	1	0	0	1	0	0	1	0	0
68	90.0402	580.8538	7.769349	8.467529	-17.50931	0	0	0	1	0	0	1	0	0
69	91.36432	580.8538	7.769349	8.422244	-17.49685	0	0	0	1	0	0	1	0	0
70	92.68844	580.8538	7.769349	8.464897	-17.47578	0	0	0	1	0	0	1	0	0
71	94.01256	580.8538	7.769349	8.423947	-17.45875	0	0	0	1	0	0	1	0	0
72	95.33668	580.8538	7.769349	-8.602453	-8.826186	0	0	0	1	0	0	1	1	1
73	96.6608	580.8538	7.769349	-0.103589	-8.774258	1	0	0	1	0	0	1	1	1
74	97.98492	580.8538	7.769349	-0.066613	-8.735598	1	0	0	1	0	0	1	1	1
75	99.30905	580.8538	7.769349	-0.100208	-8.716642	1	0	0	1	0	0	1	1	1
76	100.6332	580.8538	7.769349	-0.049464	-0.16515	1	1	1	1	1	1	1	1	1
77	101.9573	580.8538	7.769349	-0.079302	-0.144817	1	1	1	1	1	1	1	1	1
78	103.2814	580.8538	7.769349	-0.040911	-0.13715	1	1	1	1	1	1	1	1	1
79	104.6055	580.8538	7.769349	-0.066581	-0.115328	1	1	1	1	1	1	1	1	1
80	105.9296	580.8538	7.769349	8.521303	8.42757	0	0	0	1	1	1	1	1	1
81	107.2538	580.8538	7.769349	-8.592139	-0.074017	0	1	0	1	1	1	1	1	1
82	108.5779	580.8538	7.769349	17.07282	8.46052	0	0	0	0	1	0	1	1	1
83	109.902	580.8538	7.769349	17.05394	8.482987	0	0	0	0	1	0	1	1	1
84	111.2261	580.8538	7.769349	17.09056	8.483199	0	0	0	0	1	0	1	1	1
85	112.5503	580.8538	7.769349	-8.550976	-0.027007	0	1	0	1	1	1	1	1	1
86	113.8744	580.8538	7.769349	0.026186	-0.042547	1	1	1	1	1	1	1	1	1
87	115.1985	580.8538	7.769349	0.017149	-0.019574	1	1	1	1	1	1	1	1	1
88	116.5226	580.8538	7.769349	0.049864	-0.024502	1	1	1	1	1	1	1	1	1
89	117.8467	580.8538	7.769349	0.044737	-0.002125	1	1	1	1	1	1	1	1	1

v

numframes	In .avi	diff X	diff Y	diff X	diff Y	x:3%	y:3%	TOTAL	x:5%	y:5%	TOTAL	x:10%	y:10%	TOTAL
90	119.1709	580.8538	7.769349	0.07131	0.000606	1	1	1	1	1	1	1	1	1
91	120.495	580.8538	7.769349	0.047308	-8.492592	1	0	0	1	1	1	1	1	1
92	121.8191	580.8538	7.769349	17.14879	-8.48527	0	0	0	0	1	0	1	1	1
93	123.1432	580.8538	7.769349	8.590847	-16.98876	0	0	0	1	0	0	1	1	1
94	124.4673	580.8538	7.769349	0.068028	-16.96834	1	0	0	1	0	0	1	1	1
95	125.7915	580.8538	7.769349	8.612312	-16.95305	0	0	0	1	0	0	1	1	1
96	127.1156	580.8538	7.769349	0.091717	-16.95224	1	0	0	1	0	0	1	1	1
97	128.4397	580.8538	7.769349	0.097676	-16.93734	1	0	0	1	0	0	1	1	1
98	129.7638	580.8538	7.769349	-8.425978	-16.93721	0	0	0	1	0	0	1	1	1
99	131.0879	580.8538	7.769349	-16.95013	0.18123	0	1	0	0	1	0	1	1	1
100	132.4121	580.8538	7.769349	0.153584	-8.381637	1	0	0	1	1	1	1	1	1
101	133.7362	580.8538	7.769349	0.157182	-16.92227	1	0	0	1	0	0	1	1	1
102	135.0603	580.8538	7.769349	-8.36447	-8.369552	0	0	0	1	1	1	1	1	1
103	136.3844	580.8538	7.769349	-8.351529	-8.358606	0	0	0	1	1	1	1	1	1
104	137.7085	580.8538	7.769349	-8.327366	0.175941	0	1	0	1	1	1	1	1	1
105	139.0327	580.8538	7.769349	-8.312631	0.185898	0	1	0	1	1	1	1	1	1
106	140.3568	580.8538	7.769349	-8.298081	0.176322	0	1	0	1	1	1	1	1	1
107	141.6809	580.8538	7.769349	-8.281573	0.185158	0	1	0	1	1	1	1	1	1
108	143.005	580.8538	7.769349	-8.269392	0.175642	0	1	0	1	1	1	1	1	1
109	144.3291	580.8538	7.769349	-8.251397	0.183363	0	1	0	1	1	1	1	1	1
110	145.6533	580.8538	7.769349	-8.241552	0.174015	0	1	0	1	1	1	1	1	1
111	146.9774	580.8538	7.769349	-8.222343	0.180651	0	1	0	1	1	1	1	1	1
112	148.3015	580.8538	7.769349	-8.222357	-8.384017	0	0	0	1	1	1	1	1	1
113	149.6256	580.8538	7.769349	-8.202056	-8.378717	0	0	0	1	1	1	1	1	1
114	150.9497	580.8538	7.769349	-8.196902	-8.387584	0	0	0	1	1	1	1	1	1
115	152.2739	580.8538	7.769349	-8.175914	-8.383314	0	0	0	1	1	1	1	1	1
116	153.598	580.8538	7.769349	0.404696	0.165632	1	1	1	1	1	1	1	1	1
117	154.9221	580.8538	7.769349	0.412251	-16.98525	1	0	0	1	0	0	1	1	1
118	156.2462	580.8538	7.769349	9.007273	-17.03255	0	0	0	1	0	0	1	1	1

w

numframes	In .avi	diff X	diff Y	diff X	diff Y	x:3%	y:3%	TOTAL	x:5%	y:5%	TOTAL	x:10%	y:10%	TOTAL
119	157.5704	580.8538	7.769349	0.472147	-8.502335	1	0	0	1	1	1	1	1	1
120	158.8945	580.8538	7.769349	0.470089	-8.510205	1	0	0	1	1	1	1	1	1
121	160.2186	580.8538	7.769349	0.492836	-8.508234	1	0	0	1	1	1	1	1	1
122	161.5427	580.8538	7.769349	0.494135	-8.53465	1	0	0	1	1	1	1	1	1
123	162.8668	580.8538	7.769349	0.51667	-8.533657	1	0	0	1	1	1	1	1	1
124	164.191	580.8538	7.769349	0.511363	-8.540454	1	0	0	1	1	1	1	1	1
125	165.5151	580.8538	7.769349	0.541817	0.008498	1	1	1	1	1	1	1	1	1
126	166.8392	580.8538	7.769349	26.17548	-0.00442	0	1	0	0	1	0	1	1	1
127	168.1633	580.8538	7.769349	0.556821	0.001962	1	1	1	1	1	1	1	1	1
128	169.4874	580.8538	7.769349	9.096003	-0.006019	0	1	0	1	1	1	1	1	1
129	170.8116	580.8538	7.769349	9.117227	-0.016341	0	1	0	1	1	1	1	1	1
130	172.1357	580.8538	7.769349	9.108405	-0.030574	0	1	0	1	1	1	1	1	1
131	173.4598	580.8538	7.769349	9.137757	8.517809	0	0	0	1	1	1	1	1	1
132	174.7839	580.8538	7.769349	17.67106	8.521328	0	0	0	0	1	0	1	1	1
133	176.108	580.8538	7.769349	17.68618	-0.040948	0	1	0	0	1	0	1	1	1
134	177.4322	580.8538	7.769349	17.67531	-0.025998	0	1	0	0	1	0	1	1	1
135	178.7563	580.8538	7.769349	17.70326	8.521506	0	0	0	0	1	0	1	1	1
136	180.0804	580.8538	7.769349	17.68774	8.528748	0	0	0	0	1	0	1	1	1
137	181.4045	580.8538	7.769349	-16.49977	-0.031573	0	1	0	0	1	0	1	1	1
138	182.7286	580.8538	7.769349	-25.04896	8.501121	0	0	0	0	1	0	1	1	1
139	184.0528	580.8538	7.769349	-25.05275	-8.631655	0	0	0	0	0	0	1	1	1
140	185.3769	580.8538	7.769349	-25.06428	-8.633222	0	0	0	0	0	0	1	1	1
141	186.701	580.8538	7.769349	-25.03987	-0.07582	0	1	0	0	1	0	1	1	1
142	188.0251	580.8538	7.769349	-25.06158	-8.637549	0	0	0	0	0	0	1	1	1
143	189.3492	580.8538	7.769349	-25.04815	-8.640892	0	0	0	0	0	0	1	1	1
144	190.6734	580.8538	7.769349	-25.06006	-8.641502	0	0	0	0	0	0	1	1	1
145	191.9975	580.8538	7.769349	-25.04767	-8.644875	0	0	0	0	0	0	1	1	1
146	193.3216	580.8538	7.769349	-25.03758	-8.648195	0	0	0	0	0	0	1	1	1
147	194.6457	580.8538	7.769349	-25.05125	-8.648023	0	0	0	0	0	0	1	1	1

x

numframes	In .avi	diff X	diff Y	diff X	diff Y	x:3%	y:3%	TOTAL	x:5%	y:5%	TOTAL	x:10%	y:10%	TOTAL
148	195.9698	580.8538	7.769349	-25.04198	-8.651263	0	0	0	0	0	0	1	1	1
149	197.294	580.8538	7.769349	-25.05512	-8.650762	0	0	0	0	0	0	1	1	1
150	198.6181	580.8538	7.769349	-25.04668	-8.653902	0	0	0	0	0	0	1	1	1
151	199.9422	580.8538	7.769349	-16.49064	8.47391	0	0	0	0	1	0	1	1	1
152	201.2663	580.8538	7.769349	9.145615	-8.612335	0	0	0	1	0	0	1	1	1
153	202.5905	580.8538	7.769349	-7.963655	-8.595923	0	0	0	1	1	1	1	1	1
154	203.9146	580.8538	7.769349	-7.946771	-0.048751	0	1	0	1	1	1	1	1	1
155	205.2387	580.8538	7.769349	0.593055	-0.061077	1	1	1	1	1	1	1	1	1
156	206.5628	580.8538	7.769349	0.599316	-0.063829	1	1	1	1	1	1	1	1	1
157	207.8869	580.8538	7.769349	0.58774	-0.062601	1	1	1	1	1	1	1	1	1
158	209.2111	580.8538	7.769349	0.593146	-0.065184	1	1	1	1	1	1	1	1	1
159	210.5352	580.8538	7.769349	0.582126	-0.063828	1	1	1	1	1	1	1	1	1
160	211.8593	580.8538	7.769349	0.586706	-0.066229	1	1	1	1	1	1	1	1	1
161	213.1834	580.8538	7.769349	17.68224	8.479043	0	0	0	0	1	0	1	1	1
162	214.5075	580.8538	7.769349	17.68623	8.476827	0	0	0	0	1	0	1	1	1
163	215.8317	580.8538	7.769349	17.67599	8.478247	0	0	0	0	1	0	1	1	1
164	217.1558	580.8538	7.769349	17.66725	8.479384	0	0	0	0	1	0	1	1	1
165	218.4799	580.8538	7.769349	-7.982246	-0.063146	0	1	0	1	1	1	1	1	1
166	219.804	580.8538	7.769349	-8.000247	-8.601828	0	0	0	1	0	0	1	1	1
167	221.1281	580.8538	7.769349	-7.997444	-8.584895	0	0	0	1	1	1	1	1	1
168	222.4523	580.8538	7.769349	-7.976953	8.539295	0	0	0	1	1	1	1	1	1
169	223.7764	580.8538	7.769349	-7.995042	0.001532	0	1	0	1	1	1	1	1	1
170	225.1005	580.8538	7.769349	-42.19537	0.043608	0	1	0	0	1	0	0	1	0
171	226.4246	580.8538	7.769349	-16.54947	0.014593	0	1	0	0	1	0	1	1	1
172	227.7487	580.8538	7.769349	-50.75384	0.066371	0	1	0	0	1	0	0	1	0
173	229.0729	580.8538	7.769349	-33.64513	8.589286	0	0	0	0	1	0	0	1	0
174	230.397	580.8538	7.769349	-50.73951	8.59603	0	0	0	0	1	0	0	1	0
175	231.7211	580.8538	7.769349	-50.74551	8.597287	0	0	0	0	1	0	0	1	0
176	233.0452	580.8538	7.769349	-50.74631	8.596576	0	0	0	0	1	0	0	1	0

y

numframes	In .avi	diff X	diff Y	diff X	diff Y	x:3%	y:3%	TOTAL	x:5%	y:5%	TOTAL	x:10%	y:10%	TOTAL
177	234.3693	580.8538	7.769349	-50.75176	8.597762	0	0	0	0	1	0	0	1	0
178	235.6935	580.8538	7.769349	-16.56178	8.572626	0	0	0	0	1	0	1	1	1
179	237.0176	580.8538	7.769349	-25.11765	8.587214	0	0	0	0	1	0	1	1	1
180	238.3417	580.8538	7.769349	-16.56871	8.573358	0	0	0	0	1	0	1	1	1
181	239.6658	580.8538	7.769349	-16.5734	8.57447	0	0	0	0	1	0	1	1	1
182	240.9899	580.8538	7.769349	-16.58816	0.008712	0	1	0	0	1	0	1	1	1
183	242.3141	580.8538	7.769349	-8.04537	-0.012001	0	1	0	1	1	1	1	1	1
184	243.6382	580.8538	7.769349	0.512497	8.513486	1	0	0	1	1	1	1	1	1
185	244.9623	580.8538	7.769349	0.489313	-8.600847	1	0	0	1	0	0	1	1	1
186	246.2864	580.8538	7.769349	-8.060048	-8.607119	0	0	0	1	0	0	1	1	1
187	247.6106	580.8538	7.769349	-8.053343	-0.057674	0	1	0	1	1	1	1	1	1
188	248.9347	580.8538	7.769349	17.59369	8.481877	0	0	0	0	1	0	1	1	1
189	250.2588	580.8538	7.769349	0.487561	-0.059995	1	1	1	1	1	1	1	1	1
190	251.5829	580.8538	7.769349	17.58741	8.491928	0	0	0	0	1	0	1	1	1
191	252.907	580.8538	7.769349	0.491883	8.498488	1	0	0	1	1	1	1	1	1
192	254.2312	580.8538	7.769349	17.5816	8.492819	0	0	0	0	1	0	1	1	1
193	255.5553	580.8538	7.769349	0.486695	8.499248	1	0	0	1	1	1	1	1	1
194	256.8794	580.8538	7.769349	17.57611	8.493684	0	0	0	0	1	0	1	1	1
195	258.2035	580.8538	7.769349	0.472074	-0.0575	1	1	1	1	1	1	1	1	1
196	259.5276	580.8538	7.769349	0.468568	-0.057157	1	1	1	1	1	1	1	1	1
197	260.8518	580.8538	7.769349	17.56996	8.485812	0	0	0	0	1	0	1	1	1
198	262.1759	580.8538	7.769349	0.493392	25.59763	1	0	0	1	0	0	1	0	0
199	263.5	580.8538	7.769349	9.028903	17.04666	0	0	0	1	0	0	1	1	1
200	264.8241	580.8538	7.769349	9.02523	17.04698	0	0	0	1	0	0	1	1	1
201	266.1482	580.8538	7.769349	0.468861	8.502001	1	0	0	1	1	1	1	1	1
202	267.4724	580.8538	7.769349	26.08435	-8.601614	0	0	0	0	0	0	1	1	1
203	268.7965	580.8538	7.769349	26.08426	-8.601407	0	0	0	0	0	0	1	1	1
204	270.1206	580.8538	7.769349	26.08052	-8.600858	0	0	0	0	0	0	1	1	1
205	271.4447	580.8538	7.769349	8.988447	-8.595421	0	0	0	1	1	1	1	1	1

z

numframes	In .avi	diff X	diff Y	diff X	diff Y	x:3%	y:3%	TOTAL	x:5%	y:5%	TOTAL	x:10%	y:10%	TOTAL
206	272.7688	580.8538	7.769349	26.08663	-0.052465	0	1	0	0	1	0	1	1	1
207	274.093	580.8538	7.769349	26.08709	-0.052397	0	1	0	0	1	0	1	1	1
208	275.4171	580.8538	7.769349	26.08345	-0.051888	0	1	0	0	1	0	1	1	1
209	276.7412	580.8538	7.769349	26.08416	-0.05188	0	1	0	0	1	0	1	1	1
210	278.0653	580.8538	7.769349	26.08061	-0.051375	0	1	0	0	1	0	1	1	1
211	279.3894	580.8538	7.769349	26.08153	-0.05142	0	1	0	0	1	0	1	1	1
212	280.7136	580.8538	7.769349	34.62415	-0.0533	0	1	0	0	1	0	0	1	0
213	282.0377	580.8538	7.769349	34.62528	-0.053391	0	1	0	0	1	0	0	1	0
214	283.3618	580.8538	7.769349	34.6219	-0.052903	0	1	0	0	1	0	0	1	0
215	284.6859	580.8538	7.769349	34.6232	-0.053035	0	1	0	0	1	0	0	1	0
216	286.0101	580.8538	7.769349	34.61995	-0.052566	0	1	0	0	1	0	0	1	0
217	287.3342	580.8538	7.769349	26.07539	-0.050378	0	1	0	0	1	0	1	1	1
218	288.6583	580.8538	7.769349	34.6183	-0.052289	0	1	0	0	1	0	0	1	0
219	289.9824	580.8538	7.769349	34.61988	-0.052484	0	1	0	0	1	0	0	1	0
220	291.3065	580.8538	7.769349	34.61693	-0.052068	0	1	0	0	1	0	0	1	0
221	292.6307	580.8538	7.769349	34.61861	-0.052286	0	1	0	0	1	0	0	1	0
222	293.9548	580.8538	7.769349	34.61582	-0.051899	0	1	0	0	1	0	0	1	0
223	295.2789	580.8538	7.769349	34.61759	-0.052134	0	1	0	0	1	0	0	1	0
224	296.603	580.8538	7.769349	26.06906	-0.049456	0	1	0	0	1	0	1	1	1
225	297.9271	580.8538	7.769349	26.07086	-0.049698	0	1	0	0	1	0	1	1	1
226	299.2513	580.8538	7.769349	26.06844	-0.049385	0	1	0	0	1	0	1	1	1
227	300.5754	580.8538	7.769349	34.61971	-0.063047	0	1	0	0	1	0	0	1	0
228	301.8995	580.8538	7.769349	34.61744	-0.062766	0	1	0	0	1	0	0	1	0
229	303.2236	580.8538	7.769349	26.06992	-0.049605	0	1	0	0	1	0	1	1	1
230	304.5477	580.8538	7.769349	8.999339	17.0405	0	0	0	1	0	0	1	1	1
231	305.8719	580.8538	7.769349	0.451765	17.05374	1	0	0	1	0	0	1	1	1
232	307.196	580.8538	7.769349	0.440415	8.496956	1	0	0	1	1	1	1	1	1
233	308.5201	580.8538	7.769349	26.08926	17.03684	0	0	0	0	0	0	1	1	1
234	309.8442	580.8538	7.769349	26.08752	17.03699	0	0	0	0	0	0	1	1	1

aa

numframes	In .avi	diff X	diff Y	diff X	diff Y	x:3%	y:3%	TOTAL	x:5%	y:5%	TOTAL	x:10%	y:10%	TOTAL
235	311.1683	580.8538	7.769349	26.08941	17.03679	0	0	0	0	0	0	1	1	1
236	312.4925	580.8538	7.769349	26.07807	8.48896	0	0	0	0	1	0	1	1	1
237	313.8166	580.8538	7.769349	26.07993	8.488753	0	0	0	0	1	0	1	1	1
238	315.1407	580.8538	7.769349	17.5327	8.491327	0	0	0	0	1	0	1	1	1
239	316.4648	580.8538	7.769349	26.08035	8.488666	0	0	0	0	1	0	1	1	1
240	317.7889	580.8538	7.769349	26.0791	8.48875	0	0	0	0	1	0	1	1	1
241	319.1131	580.8538	7.769349	26.08088	8.488565	0	0	0	0	1	0	1	1	1
242	320.4372	580.8538	7.769349	17.53395	8.491093	0	0	0	0	1	0	1	1	1
243	321.7613	580.8538	7.769349	17.53565	8.490928	0	0	0	0	1	0	1	1	1
244	323.0854	580.8538	7.769349	17.53473	8.490964	0	0	0	0	1	0	1	1	1
245	324.4095	580.8538	7.769349	17.53637	8.490813	0	0	0	0	1	0	1	1	1
246	325.7337	580.8538	7.769349	17.53559	8.490829	0	0	0	0	1	0	1	1	1
247	327.0578	580.8538	7.769349	17.53716	8.490692	0	0	0	0	1	0	1	1	1
248	328.3819	580.8538	7.769349	26.08596	8.477257	0	0	0	0	1	0	1	1	1
249	329.706	580.8538	7.769349	17.53802	8.49057	0	0	0	0	1	0	1	1	1
250	331.0302	580.8538	7.769349	17.5375	8.490554	0	0	0	0	1	0	1	1	1
251	332.3543	580.8538	7.769349	8.987105	-0.065448	0	1	0	1	1	1	1	1	1
252	333.6784	580.8538	7.769349	0.427625	-8.590322	1	0	0	1	1	1	1	1	1
253	335.0025	580.8538	7.769349	-127.7931	-8.428086	0	0	0	0	1	0	0	1	0
254	336.3266	580.8538	7.769349	-127.7697	8.667259	0	0	0	0	0	0	0	1	0
255	337.6508	580.8538	7.769349	-136.3617	17.19642	0	0	0	0	0	0	0	1	0
256	338.9749	580.8538	7.769349	-119.1463	26.41742	0	0	0	0	0	0	0	0	0
257	340.299	580.8538	7.769349	-161.5978	17.68793	0	0	0	0	0	0	0	0	0
258	341.6231	580.8538	7.769349	-161.2661	17.60661	0	0	0	0	0	0	0	0	0
259	342.9472	580.8538	7.769349	-143.8852	17.57316	0	0	0	0	0	0	0	0	0
260	344.2714	580.8538	7.769349	-143.5571	17.49795	0	0	0	0	0	0	0	0	0
261	345.5955	580.8538	7.769349	-151.9327	17.51187	0	0	0	0	0	0	0	0	0
262	346.9196	580.8538	7.769349	-143.0371	17.51645	0	0	0	0	0	0	0	0	0
263	348.2437	580.8538	7.769349	-151.4629	17.52276	0	0	0	0	0	0	0	0	0

bb

numframes	In .avi	diff X	diff Y	diff X	diff Y	x:3%	y:3%	TOTAL	x:5%	y:5%	TOTAL	x:10%	y:10%	TOTAL
264	349.5678	580.8538	7.769349	-133.9715	17.53655	0	0	0	0	0	0	0	0	0
265	350.892	580.8538	7.769349	-133.8431	17.53454	0	0	0	0	0	0	0	0	0
266	352.2161	580.8538	7.769349	-133.5379	17.53465	0	0	0	0	0	0	0	0	0
267	353.5402	580.8538	7.769349	-133.4436	17.53456	0	0	0	0	0	0	0	0	0
268	354.8643	580.8538	7.769349	-124.5394	17.53207	0	0	0	0	0	0	0	0	0
269	356.1884	580.8538	7.769349	-124.4744	17.53477	0	0	0	0	0	0	0	0	0
270	357.5126	580.8538	7.769349	-115.5706	17.53447	0	0	0	0	0	0	0	0	0
271	358.8367	580.8538	7.769349	-124.1495	17.53786	0	0	0	0	0	0	0	0	0
272	360.1608	580.8538	7.769349	-123.8754	17.53926	0	0	0	0	0	0	0	0	0
273	361.4849	580.8538	7.769349	-132.4955	8.931981	0	0	0	0	0	0	0	1	0
274	362.809	580.8538	7.769349	-123.635	0.285588	0	1	0	0	1	0	0	1	0
275	364.1332	580.8538	7.769349	-115.1908	-0.015457	0	1	0	0	1	0	0	1	0
276	365.4573	580.8538	7.769349	-123.9223	-8.79955	0	0	0	0	0	0	0	1	0
277	366.7814	580.8538	7.769349	22.25137	-0.386063	0	1	0	0	1	0	1	1	1
278	368.1055	580.8538	7.769349	13.50582	-0.339875	0	1	0	0	1	0	1	1	1
279	369.4296	580.8538	7.769349	13.15164	-0.343419	0	1	0	1	1	1	1	1	1
280	370.7538	580.8538	7.769349	12.98153	-8.934028	0	0	0	1	0	0	1	1	1
281	372.0779	580.8538	7.769349	4.082125	-0.325146	1	1	1	1	1	1	1	1	1
282	373.402	580.8538	7.769349	12.49977	-8.947698	0	0	0	1	0	0	1	1	1
283	374.7261	580.8538	7.769349	12.23748	-0.340875	0	1	0	1	1	1	1	1	1
284	376.0503	580.8538	7.769349	12.04792	-8.93273	0	0	0	1	0	0	1	1	1
285	377.3744	580.8538	7.769349	11.83526	8.256057	0	0	0	1	1	1	1	1	1
286	378.6985	580.8538	7.769349	11.65365	8.262588	0	0	0	1	1	1	1	1	1
287	380.0226	580.8538	7.769349	2.859515	8.280946	1	0	0	1	1	1	1	1	1
288	381.3467	580.8538	7.769349	11.25763	8.268182	0	0	0	1	1	1	1	1	1
289	382.6709	580.8538	7.769349	11.07878	8.268089	0	0	0	1	1	1	1	1	1
290	383.995	580.8538	7.769349	2.299585	-0.28765	1	1	1	1	1	1	1	1	1
291	385.3191	580.8538	7.769349	2.137808	-8.858054	1	0	0	1	0	0	1	1	1
292	386.6432	580.8538	7.769349	1.964841	-0.262146	1	1	1	1	1	1	1	1	1

CC

numframes	In .avi	diff X	diff Y	diff X	diff Y	x:3%	y:3%	TOTAL	x:5%	y:5%	TOTAL	x:10%	y:10%	TOTAL
293	387.9673	580.8538	7.769349	-15.29624	-0.209072	0	1	0	0	1	0	1	1	1
294	389.2915	580.8538	7.769349	-126.8281	-0.065531	0	1	0	0	1	0	0	1	0
295	390.6156	580.8538	7.769349	-126.9215	-8.621293	0	0	0	0	0	0	0	1	0
296	391.9397	580.8538	7.769349	-135.6115	-0.046768	0	1	0	0	1	0	0	1	0
297	393.2638	580.8538	7.769349	-127.1238	-8.614577	0	0	0	0	0	0	0	1	0
298	394.5879	580.8538	7.769349	-127.2655	-8.608673	0	0	0	0	0	0	0	1	0
299	395.9121	580.8538	7.769349	-127.2952	-0.043034	0	1	0	0	1	0	0	1	0
300	397.2362	580.8538	7.769349	-136.0034	-8.587826	0	0	0	0	1	0	0	1	0
301	398.5603	580.8538	7.769349	-127.4624	-0.0412	0	1	0	0	1	0	0	1	0
302	399.8844	580.8538	7.769349	-127.5976	-0.038402	0	1	0	0	1	0	0	1	0
303	401.2085	580.8538	7.769349	-136.1789	-8.584395	0	0	0	0	1	0	0	1	0
304	402.5327	580.8538	7.769349	-127.761	-8.585731	0	0	0	0	1	0	0	1	0
305	403.8568	580.8538	7.769349	-161.8938	25.63328	0	0	0	0	0	0	0	0	0
306	405.1809	580.8538	7.769349	-119.3426	-8.598141	0	0	0	0	1	0	0	1	0
307	406.505	580.8538	7.769349	-119.3195	-0.054443	0	1	0	0	1	0	0	1	0
308	407.8291	580.8538	7.769349	-128.0009	-8.589967	0	0	0	0	1	0	0	1	0
309	409.1533	580.8538	7.769349	-136.5152	-0.041781	0	1	0	0	1	0	0	1	0
310	410.4774	580.8538	7.769349	-128.1007	-8.588458	0	0	0	0	1	0	0	1	0
311	411.8015	580.8538	7.769349	-136.6055	-0.042215	0	1	0	0	1	0	0	1	0
312	413.1256	580.8538	7.769349	-119.6487	-8.592953	0	0	0	0	1	0	0	1	0
313	414.4497	580.8538	7.769349	-119.6185	-8.595416	0	0	0	0	1	0	0	1	0
314	415.7739	580.8538	7.769349	-119.7232	-8.592379	0	0	0	0	1	0	0	1	0
315	417.098	580.8538	7.769349	-136.7496	-0.043838	0	1	0	0	1	0	0	1	0
316	418.4221	580.8538	7.769349	-111.2499	-8.597188	0	0	0	0	1	0	0	1	0
317	419.7462	580.8538	7.769349	-111.2084	-8.599738	0	0	0	0	1	0	0	1	0
318	421.0704	580.8538	7.769349	-136.893	-0.044626	0	1	0	0	1	0	0	1	0
319	422.3945	580.8538	7.769349	-162.4308	17.04862	0	0	0	0	0	0	0	1	0
320	423.7186	580.8538	7.769349	-128.41	-8.588309	0	0	0	0	1	0	0	1	0
321	425.0427	580.8538	7.769349	-119.8285	-8.595597	0	0	0	0	1	0	0	1	0

dd

numframes	In .avi	diff X	diff Y	diff X	diff Y	x:3%	y:3%	TOTAL	x:5%	y:5%	TOTAL	x:10%	y:10%	TOTAL
322	426.3668	580.8538	7.769349	-136.9608	-0.047673	0	1	0	0	1	0	0	1	0
323	427.691	580.8538	7.769349	-111.3222	-8.601054	0	0	0	0	0	0	0	1	0
324	429.0151	580.8538	7.769349	-119.9276	-8.595268	0	0	0	0	1	0	0	1	0
325	430.3392	580.8538	7.769349	-128.3977	-0.055376	0	1	0	0	1	0	0	1	0
326	431.6633	580.8538	7.769349	-128.4731	-8.591921	0	0	0	0	1	0	0	1	0
327	432.9874	580.8538	7.769349	-136.9429	-0.051788	0	1	0	0	1	0	0	1	0
328	434.3116	580.8538	7.769349	-111.4135	-8.602451	0	0	0	0	0	0	0	1	0
329	435.6357	580.8538	7.769349	-136.9477	-0.053196	0	1	0	0	1	0	0	1	0
330	436.9598	580.8538	7.769349	-128.4771	-8.595003	0	0	0	0	1	0	0	1	0
331	438.2839	580.8538	7.769349	-128.4251	-8.59658	0	0	0	0	1	0	0	1	0
332	439.608	580.8538	7.769349	-128.4697	-8.596624	0	0	0	0	1	0	0	1	0
333	440.9322	580.8538	7.769349	-119.8848	-8.602685	0	0	0	0	0	0	0	1	0
334	442.2563	580.8538	7.769349	-128.4571	-8.598254	0	0	0	0	1	0	0	1	0
335	443.5804	580.8538	7.769349	-128.3947	-0.05272	0	1	0	0	1	0	0	1	0
336	444.9045	580.8538	7.769349	-136.9729	-8.59505	0	0	0	0	1	0	0	1	0
337	446.2286	580.8538	7.769349	-145.4467	-0.043801	0	1	0	0	1	0	0	1	0
338	447.5528	580.8538	7.769349	-145.4732	-0.044926	0	1	0	0	1	0	0	1	0
339	448.8769	580.8538	7.769349	-145.4277	-0.044782	0	1	0	0	1	0	0	1	0
340	450.201	580.8538	7.769349	-128.3766	-0.06828	0	1	0	0	1	0	0	1	0
341	451.5251	580.8538	7.769349	-8.822302	-8.763329	0	0	0	1	0	0	1	1	1
342	452.8492	580.8538	7.769349	8.232637	-8.821801	0	0	0	1	0	0	1	1	1
343	454.1734	580.8538	7.769349	8.288907	-8.831719	0	0	0	1	0	0	1	1	1
344	455.4975	580.8538	7.769349	-0.2636	-8.838862	1	0	0	1	0	0	1	1	1
345	456.8216	580.8538	7.769349	8.325248	-8.842667	0	0	0	1	0	0	1	1	1
346	458.1457	580.8538	7.769349	8.313902	-8.843921	0	0	0	1	0	0	1	1	1
347	459.4698	580.8538	7.769349	-0.160494	-0.299466	1	1	1	1	1	1	1	1	1
348	460.794	580.8538	7.769349	-0.179108	-8.841651	1	0	0	1	0	0	1	1	1
349	462.1181	580.8538	7.769349	-0.120585	-0.300065	1	1	1	1	1	1	1	1	1
350	463.4422	580.8538	7.769349	-0.134488	-8.842788	1	0	0	1	0	0	1	1	1

ee

numframes	In .avi	diff X	diff Y	diff X	diff Y	x:3%	y:3%	TOTAL	x:5%	y:5%	TOTAL	x:10%	y:10%	TOTAL
351	464.7663	580.8538	7.769349	-0.092001	-8.842515	1	0	0	1	0	0	1	1	1
352	466.0905	580.8538	7.769349	-0.076348	-0.301664	1	1	1	1	1	1	1	1	1
353	467.4146	580.8538	7.769349	-0.050002	-8.843367	1	0	0	1	0	0	1	1	1
354	468.7387	580.8538	7.769349	-0.043025	-8.84455	1	0	0	1	0	0	1	1	1
355	470.0628	580.8538	7.769349	-0.007604	-8.844072	1	0	0	1	0	0	1	1	1
356	471.3869	580.8538	7.769349	0.01565	-0.301841	1	1	1	1	1	1	1	1	1
357	472.7111	580.8538	7.769349	0.060346	8.243105	1	0	0	1	1	1	1	1	1
358	474.0352	580.8538	7.769349	0.048607	-8.845672	1	0	0	1	0	0	1	1	1
359	475.3593	580.8538	7.769349	17.18396	8.23593	0	0	0	0	1	0	1	1	1
360	476.6834	580.8538	7.769349	0.106332	-0.301443	1	1	1	1	1	1	1	1	1
361	478.0075	580.8538	7.769349	-8.394268	8.270289	0	0	0	1	1	1	1	1	1
362	479.3317	580.8538	7.769349	-16.92932	-0.260455	0	1	0	0	1	0	1	1	1
363	480.6558	580.8538	7.769349	-16.89617	8.294289	0	0	0	0	1	0	1	1	1
364	481.9799	580.8538	7.769349	-16.87553	8.294201	0	0	0	0	1	0	1	1	1
365	483.304	580.8538	7.769349	-8.327982	-0.254244	0	1	0	1	1	1	1	1	1
366	484.6281	580.8538	7.769349	0.240779	-0.273921	1	1	1	1	1	1	1	1	1
367	485.9523	580.8538	7.769349	0.255827	-0.272845	1	1	1	1	1	1	1	1	1
368	487.2764	580.8538	7.769349	0.280433	-0.272986	1	1	1	1	1	1	1	1	1
369	488.6005	580.8538	7.769349	0.292438	-0.271982	1	1	1	1	1	1	1	1	1
370	489.9246	580.8538	7.769349	-8.213023	8.279338	0	0	0	1	1	1	1	1	1
371	491.2487	580.8538	7.769349	-16.76375	-0.246756	0	1	0	0	1	0	1	1	1
372	492.5729	580.8538	7.769349	-8.190106	-0.265821	0	1	0	1	1	1	1	1	1
373	493.897	580.8538	7.769349	-16.73161	-0.245821	0	1	0	0	1	0	1	1	1
374	495.2211	580.8538	7.769349	-16.70496	-0.2456	0	1	0	0	1	0	1	1	1
375	496.5452	580.8538	7.769349	-8.153111	-0.263949	0	1	0	1	1	1	1	1	1
376	497.8693	580.8538	7.769349	0.419589	-0.268864	1	1	1	1	1	1	1	1	1
377	499.1935	580.8538	7.769349	-8.127732	-0.249189	0	1	0	1	1	1	1	1	1
378	500.5176	580.8538	7.769349	-8.100025	-0.248835	0	1	0	1	1	1	1	1	1
379	501.8417	580.8538	7.769349	0.448075	-0.267249	1	1	1	1	1	1	1	1	1

ff

numframes	In .avi	diff X	diff Y	diff X	diff Y		x:3%	y:3%	TOTAL		x:5%	y:5%	TOTAL		x:10%	y:10%	TOTAL
380	503.1658	580.8538	7.769349	-16.60334	8.292523		0	0	0		0	1	0		1	1	1
381	504.4899	580.8538	7.769349	-8.056988	8.27384		0	0	0		1	1	1		1	1	1
382	505.8141	580.8538	7.769349	-16.57918	8.293948		0	0	0		0	1	0		1	1	1
383	507.1382	580.8538	7.769349	-16.58399	8.294244		0	0	0		0	1	0		1	1	1
384	508.4623	580.8538	7.769349	-8.023051	-0.259349		0	1	0		1	1	1		1	1	1
385	509.7864	580.8538	7.769349	-25.13335	-0.208728		0	1	0		0	1	0		1	1	1
386	511.1106	580.8538	7.769349	-16.55341	-0.239641		0	1	0		0	1	0		1	1	1
387	512.4347	580.8538	7.769349	-16.56142	-0.239449		0	1	0		0	1	0		1	1	1
388	513.7588	580.8538	7.769349	-16.53581	-0.238786		0	1	0		0	1	0		1	1	1
389	515.0829	580.8538	7.769349	-7.994622	-0.257379		0	1	0		1	1	1		1	1	1
390	516.407	580.8538	7.769349	-7.972759	-0.243552		0	1	0		1	1	1		1	1	1
391	517.7312	580.8538	7.769349	-16.51837	8.311601		0	0	0		0	1	0		1	1	1
392	519.0553	580.8538	7.769349	-25.03852	8.305777		0	0	0		0	1	0		1	1	1
393	520.3794	580.8538	7.769349	0.580073	-0.26149		1	1	1		1	1	1		1	1	1
394	521.7035	580.8538	7.769349	-16.49547	-0.236583		0	1	0		0	1	0		1	1	1
395	523.0276	580.8538	7.769349	-7.960178	-0.242275		0	1	0		1	1	1		1	1	1
396	524.3518	580.8538	7.769349	-33.56572	8.313686		0	0	0		0	1	0		0	1	0
397	525.6759	580.8538	7.769349	-33.57852	8.313261		0	0	0		0	1	0		0	1	0
398	527	580.8538	7.769349	-16.4783	-0.235446		0	1	0		0	1	0		1	1	1
									16.83673 %				34.94898 %				73.46939 %
						StateDiagram			83.67%				87.69%				89.69%

Diffx_1: X_L-X_R
Diffy_1: Y_L-Y_R
Diffx: Diffx_1/WIDTH
Diffy: Diffy_1/HEIGHT

0: DIFFERENCE FAILS TO BE < %
1: DIFFERENCE SUCCEDS TO BE < %
TOTAL: X% ⊕ Y%

gg

Appendix IV-H

NEON GRAPH DATA

		3%						5%						10%					
frame #	in .avi	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data
6	7.944724	3	243.4045	236.2352	from previous results			3	243.4045	236.2352	from previous results			2	243.4045	236.2352			
7	9.268844	3	243.4045	236.2352	from previous results			3	243.4045	236.2352	from previous results			2	243.4045	236.2352			
8	10.59296	2	243.4045	241.3952				1	256.6245	244.8352				1	243.4045	236.2352			
9	11.91709	2	243.4045	241.3952				1	256.6245	244.8352				1	243.4045	236.2352			
10	13.24121	3	243.4045	241.3952	from previous results			2	251.3345	241.3952				1	269.8445	236.2352			
11	14.56533	2	243.4045	246.5552				1	256.6245	244.8352				1	243.4045	236.2352			
12	15.88945	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	236.2352			
13	17.21357	2	243.4045	246.5552				1	256.6245	244.8352				1	243.4045	236.2352			
14	18.53769	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	236.2352			
15	19.86181	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
16	21.18593	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
17	22.51005	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
18	23.83417	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
19	25.15829	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
20	26.48241	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
21	27.80653	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
22	29.13065	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
23	30.45477	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
24	31.77889	3	243.4045	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
25	33.10302	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
26	34.42714	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
27	35.75126	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
28	37.07538	3	243.4045	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
29	38.3995	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
30	39.72362	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			

hh

		3%						5%						10%					
frame #	in .avi	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data
31	41.04774	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
32	42.37186	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
33	43.69598	3	243.4045	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
34	45.0201	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
35	46.34422	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
36	47.66834	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
37	48.99246	2	243.4045	241.3952				2	243.4045	241.3952				1	243.4045	236.2352			
38	50.31658	2	243.4045	241.3952				2	243.4045	241.3952				1	243.4045	236.2352			
39	51.6407	3	243.4045	241.3952	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
40	52.96482	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
41	54.28894	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
42	55.61307	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
43	56.93719	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
44	58.26131	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
45	59.58543	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
46	60.90955	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	253.4352			
47	62.23367	2	243.4045	241.3952				2	243.4045	241.3952				1	243.4045	236.2352			
48	63.55779	1	235.4745	246.5552				1	230.1845	244.8352				1	243.4045	253.4352			
49	64.88191	2	219.6145	231.0752				2	219.6145	231.0752				1	216.9645	236.2352			
50	66.20603	2	195.8245	225.9152				1	190.5245	227.6352				1	190.5245	236.2352			
51	67.53015	3	174.9424	227.9933	192.0643	211.476	closed 2	2	179.9645	220.7552				1	164.0845	236.2352			
52	68.85427	3	174.9604	227.9994	175.0642	211.5675	closed 2	2	172.0345	220.7552				1	164.0845	236.2352			
53	70.17839	2	172.0345	225.9152				1	177.3045	227.6352				1	164.0845	236.2352			
54	71.50251	1	164.1045	225.9152				1	164.0845	227.6352				1	164.0845	236.2352			
55	72.82663	3	164.1045	225.9152	from previous results			3	164.0845	227.6352	from previous results			2	164.0845	236.2352			
56	74.15075	2	164.1045	241.3952				2	164.1045	241.3952				1	164.0845	236.2352			
57	75.47487	2	164.1045	241.3952				2	164.1045	241.3952				1	164.0845	236.2352			
58	76.79899	2	164.1045	241.3952				2	164.1045	241.3952				1	164.0845	236.2352			

		3%						5%						10%					
frame #	in .avi	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data
59	78.12312	2	164.1045	241.3952				2	164.1045	241.3952				1	164.0845	236.2352			
60	79.44724	2	164.1045	241.3952				2	164.1045	241.3952				1	164.0845	236.2352			
61	80.77136	2	156.1745	241.3952				2	156.1745	241.3952				1	164.0845	236.2352			
62	82.09548	2	156.1745	241.3952				2	156.1745	241.3952				1	164.0845	236.2352			
63	83.4196	2	156.1745	241.3952				2	156.1745	241.3952				1	164.0845	236.2352			
64	84.74372	3	156.1745	241.3952	from previous results			3	156.1745	241.3952	from previous results			2	164.0845	253.4352			
65	86.06784	3	156.1745	241.3952	from previous results			3	156.1745	241.3952	from previous results			2	164.0845	253.4352			
66	87.39196	3	156.1745	241.3952	from previous results			3	156.1745	241.3952	from previous results			2	164.0845	253.4352			
67	88.71608	3	156.1745	241.3952	from previous results			3	156.1745	241.3952	from previous results			2	164.0845	253.4352			
68	90.0402	3	156.1745	241.3952	from previous results			3	156.1745	241.3952	from previous results			2	164.0845	253.4352			
69	91.36432	3	156.1745	241.3952	from previous results			3	156.1745	241.3952	from previous results			2	164.0845	253.4352			
70	92.68844	3	156.1745	241.3952	from previous results			3	156.1745	241.3952	from previous results			2	164.0845	253.4352			
71	94.01256	3	156.1745	241.3952	from previous results			3	156.1745	241.3952	from previous results			2	164.0845	253.4352			
72	95.33668	2	187.8945	241.3952				2	187.8945	241.3952				1	190.5245	236.2352			
73	96.6608	2	219.6145	241.3952				2	219.6145	241.3952				1	216.9645	236.2352			
74	97.98492	2	235.4745	241.3952				2	235.4745	241.3952				1	243.4045	236.2352			
75	99.30905	2	235.4745	241.3952				2	235.4745	241.3952				1	243.4045	236.2352			
76	100.6332	1	235.4745	246.5552				1	230.1845	244.8352				1	243.4045	236.2352			
77	101.9573	1	235.4745	246.5552				1	230.1845	244.8352				1	243.4045	236.2352			
78	103.2814	1	235.4745	246.5552				1	230.1845	244.8352				1	243.4045	236.2352			
79	104.6055	1	235.4745	246.5552				1	230.1845	244.8352				1	243.4045	236.2352			
80	105.9296	2	235.4745	246.5552				1	243.4045	253.4352				1	243.4045	253.4352			
81	107.2538	2	235.4745	246.5552				1	230.1845	244.8352				1	243.4045	236.2352			
82	108.5779	3	235.4745	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
83	109.902	3	235.4745	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
84	111.2261	3	235.4745	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
85	112.5503	2	235.4745	246.5552				1	230.1845	244.8352				1	243.4045	236.2352			
86	113.8744	1	235.4745	246.5552				1	230.1845	244.8352				1	243.4045	236.2352			

jj

		3%						5%						10%					
frame #	in .avi	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data
87	115.1985	1	235.4745	246.5552				1	230.1845	244.8352				1	243.4045	236.2352			
88	116.5226	1	235.4745	246.5552				1	230.1845	244.8352				1	243.4045	236.2352			
89	117.8467	1	235.4745	246.5552				1	230.1845	244.8352				1	243.4045	236.2352			
90	119.1709	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	236.2352			
91	120.495	2	267.1945	241.3952				1	269.8445	236.2352				1	269.8445	236.2352			
92	121.8191	3	267.1945	241.3952	from previous results			2	290.9845	241.3952				1	296.2845	236.2352			
93	123.1432	3	267.1945	241.3952	from previous results			2	306.8445	236.2352				1	322.7245	219.0352			
94	124.4673	3	319.5022	219.0323	319.4341	236.0007	closed 2	2	322.7045	225.9152				1	322.7245	219.0352			
95	125.7915	3	328.0169	219.0237	319.4046	235.9768	closed 2	2	322.7045	225.9152				1	322.7245	219.0352			
96	127.1156	3	328.0471	219.0359	327.9554	235.9881	closed 2	2	330.6345	225.9152				1	322.7245	219.0352			
97	128.4397	3	328.0241	219.03	327.9265	235.9674	closed 2	2	330.6345	225.9152				1	322.7245	219.0352			
98	129.7638	3	328.0544	219.0415	336.4803	235.9787	closed 2	2	330.6345	225.9152				1	322.7245	219.0352			
99	131.0879	3	328.0425	227.5803	344.9926	227.3991	closed 2	2	338.5645	225.9152				1	322.7245	219.0352			
100	132.4121	2	338.5645	220.7552				1	335.9445	219.0352				1	349.1645	219.0352			
101	133.7362	3	336.5916	219.0444	336.4344	235.9667	closed 2	2	338.5645	225.9152				1	349.1645	219.0352			
102	135.0603	2	338.5645	220.7552				1	335.9445	219.0352				1	349.1645	219.0352			
103	136.3844	2	338.5645	220.7552				1	335.9445	219.0352				1	349.1645	219.0352			
104	137.7085	2	338.5645	225.9152				1	335.9445	227.6352				1	349.1645	219.0352			
105	139.0327	2	338.5645	225.9152				1	335.9445	227.6352				1	349.1645	219.0352			
106	140.3568	2	338.5645	225.9152				1	335.9445	227.6352				1	349.1645	219.0352			
107	141.6809	2	338.5645	225.9152				1	335.9445	227.6352				1	349.1645	219.0352			
108	143.005	2	338.5645	225.9152				1	335.9445	227.6352				1	349.1645	219.0352			
109	144.3291	2	338.5645	225.9152				1	335.9445	227.6352				1	349.1645	219.0352			
110	145.6533	2	338.5645	225.9152				1	335.9445	227.6352				1	349.1645	219.0352			
111	146.9774	2	338.5645	225.9152				1	335.9445	227.6352				1	349.1645	219.0352			
112	148.3015	2	338.5645	231.0752				1	335.9445	227.6352				1	349.1645	236.2352			
113	149.6256	2	338.5645	231.0752				1	335.9445	227.6352				1	349.1645	236.2352			
114	150.9497	2	338.5645	231.0752				1	335.9445	227.6352				1	349.1645	236.2352			

kk

		3%						5%						10%					
frame #	in .avi	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data
115	152.2739	2	338.5645	231.0752				1	335.9445	227.6352				1	349.1645	236.2352			
116	153.598	1	338.5645	225.9152				1	335.9445	227.6352				1	349.1645	236.2352			
117	154.9221	3	302.5894	219.1167	302.1771	236.102	closed 2	2	298.9145	225.9152				1	296.2845	219.0352			
118	156.2462	3	268.415	219.1276	259.4077	236.1602	closed 2	2	267.1945	225.9152				1	269.8445	219.0352			
119	157.5704	2	251.3345	231.0752				1	256.6245	227.6352				1	243.4045	236.2352			
120	158.8945	2	251.3345	231.0752				1	256.6245	227.6352				1	243.4045	236.2352			
121	160.2186	2	251.3345	231.0752				1	256.6245	227.6352				1	243.4045	236.2352			
122	161.5427	2	243.4045	241.3952				1	243.4045	236.2352				1	243.4045	236.2352			
123	162.8668	2	243.4045	241.3952				1	243.4045	236.2352				1	243.4045	236.2352			
124	164.191	2	243.4045	241.3952				1	243.4045	236.2352				1	243.4045	236.2352			
125	165.5151	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	236.2352			
126	166.8392	3	243.4045	246.5552	from previous results			2	259.2645	246.5552				1	269.8445	236.2352			
127	168.1633	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	236.2352			
128	169.4874	2	243.4045	246.5552				1	256.6245	244.8352				1	243.4045	236.2352			
129	170.8116	2	243.4045	251.7152				1	256.6245	253.4352				1	243.4045	253.4352			
130	172.1357	2	243.4045	262.0352				1	256.6245	262.0352				1	243.4045	253.4352			
131	173.4598	2	243.4045	267.1952				1	256.6245	270.6352				1	243.4045	270.6352			
132	174.7839	3	243.4045	267.1952	from previous results			2	259.2645	267.1952				1	269.8445	270.6352			
133	176.108	3	243.4045	267.1952	from previous results			2	251.3345	262.0352				1	269.8445	253.4352			
134	177.4322	3	243.4045	267.1952	from previous results			2	251.3345	246.5552				1	269.8445	236.2352			
135	178.7563	3	243.4045	267.1952	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
136	180.0804	3	243.4045	267.1952	from previous results			2	259.2645	246.5552				1	269.8445	253.4352			
137	181.4045	3	243.4045	267.1952	from previous results			2	251.3345	272.3552				1	243.4045	270.6352			
138	182.7286	3	243.4045	267.1952	from previous results			2	243.4045	292.9952				1	243.4045	287.8352			
139	184.0528	3	243.4045	267.1952	from previous results			2	243.4045	308.4752				1	243.4045	305.0352			
140	185.3769	3	243.4045	267.1952	from previous results			2	243.4045	308.4752				1	243.4045	305.0352			
141	186.701	3	243.4045	267.1952	from previous results			2	243.4045	303.3152				1	243.4045	305.0352			
142	188.0251	3	243.4045	267.1952	from previous results			2	243.4045	308.4752				1	243.4045	305.0352			

II

		3%						5%						10%					
frame #	in .avi	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data
143	189.3492	3	243.4045	267.1952	from previous results			2	243.4045	308.4752				1	243.4045	305.0352			
144	190.6734	3	243.4045	267.1952	from previous results			2	243.4045	308.4752				1	243.4045	305.0352			
145	191.9975	3	243.4045	267.1952	from previous results			2	243.4045	308.4752				1	243.4045	305.0352			
146	193.3216	3	243.4045	267.1952	from previous results			2	243.4045	308.4752				1	243.4045	305.0352			
147	194.6457	3	243.4045	267.1952	from previous results			2	243.4045	308.4752				1	243.4045	305.0352			
148	195.9698	3	243.4045	267.1952	from previous results			2	243.4045	308.4752				1	243.4045	305.0352			
149	197.294	3	243.4045	267.1952	from previous results			2	243.4045	308.4752				1	243.4045	305.0352			
150	198.6181	3	243.4045	267.1952	from previous results			2	243.4045	308.4752				1	243.4045	305.0352			
151	199.9422	3	243.4045	267.1952	from previous results			2	251.3345	292.9952				1	243.4045	287.8352			
152	201.2663	2	259.2645	246.5552				2	259.2645	246.5552				1	269.8445	236.2352			
153	202.5905	2	243.4045	241.3952				1	243.4045	236.2352				1	243.4045	236.2352			
154	203.9146	2	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	236.2352			
155	205.2387	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	236.2352			
156	206.5628	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	236.2352			
157	207.8869	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	236.2352			
158	209.2111	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	236.2352			
159	210.5352	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	236.2352			
160	211.8593	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	236.2352			
161	213.1834	3	243.4045	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
162	214.5075	3	243.4045	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
163	215.8317	3	243.4045	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
164	217.1558	3	243.4045	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
165	218.4799	2	235.4745	246.5552				1	230.1845	244.8352				1	243.4045	236.2352			
166	219.804	2	235.4745	231.0752				2	235.4745	231.0752				1	243.4045	236.2352			
167	221.1281	2	235.4745	215.5952				1	230.1845	210.4352				1	243.4045	219.0352			
168	222.4523	2	235.4745	200.1152				1	230.1845	201.8352				1	243.4045	201.8352			
169	223.7764	2	235.4745	184.6352				1	230.1845	184.6352				1	243.4045	184.6352			
170	225.1005	3	225.8503	184.9751	268.0457	184.9315	closed 2	3	225.8503	184.9751	268.0457	184.9315	closed 2	2	243.4045	184.6352			

mm

		3%						5%						10%					
frame #	in .avi	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data
171	226.4246	3	225.8344	176.4263	242.3838	176.4117	closed 2	2	235.4745	174.3152				1	216.9645	184.6352			
172	227.7487	3	225.8366	176.4258	276.5905	176.3594	closed 2	3	225.8366	176.4258	276.5905	176.3594	closed 2	2	243.4045	184.6352			
173	229.0729	3	217.2833	176.429	250.9284	167.8397	closed 2	3	217.2833	176.429	250.9284	167.8397	closed 2	2	243.4045	167.4352			
174	230.397	3	200.191	176.436	250.9305	167.84	closed 2	3	200.191	176.436	250.9305	167.84	closed 2	2	216.9645	167.4352			
175	231.7211	3	200.1855	176.4357	250.931	167.8384	closed 2	3	200.1855	176.4357	250.931	167.8384	closed 2	2	216.9645	167.4352			
176	233.0452	3	200.1867	176.4351	250.933	167.8385	closed 2	3	200.1867	176.4351	250.933	167.8385	closed 2	2	216.9645	167.4352			
177	234.3693	3	200.1817	176.4349	250.9334	167.8371	closed 2	3	200.1817	176.4349	250.9334	167.8371	closed 2	2	216.9645	167.4352			
178	235.6935	3	225.8229	176.4231	242.3847	167.8505	closed 2	2	235.4745	174.3152				1	216.9645	184.6352			
179	237.0176	3	225.8179	176.4231	250.9356	167.8358	closed 2	2	235.4745	174.3152				1	216.9645	184.6352			
180	238.3417	3	225.8181	176.4224	242.3869	167.849	closed 2	2	235.4745	174.3152				1	216.9645	184.6352			
181	239.6658	3	225.8136	176.4224	242.387	167.848	closed 2	2	235.4745	174.3152				1	216.9645	184.6352			
182	240.9899	3	234.379	193.5139	250.9671	193.5052	closed 2	2	243.4045	194.9552				1	243.4045	201.8352			
183	242.3141	2	243.4045	210.4352				1	243.4045	210.4352				1	243.4045	219.0352			
184	243.6382	2	243.4045	225.9152				1	243.4045	227.6352				1	243.4045	236.2352			
185	244.9623	2	243.4045	231.0752				2	243.4045	231.0752				1	243.4045	236.2352			
186	246.2864	2	235.4745	241.3952				2	235.4745	241.3952				1	243.4045	236.2352			
187	247.6106	2	235.4745	246.5552				1	230.1845	244.8352				1	243.4045	236.2352			
188	248.9347	3	235.4745	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
189	250.2588	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	236.2352			
190	251.5829	3	243.4045	246.5552	from previous results			2	251.3345	241.3952				1	269.8445	236.2352			
191	252.907	2	243.4045	241.3952				1	243.4045	244.8352				1	243.4045	236.2352			
192	254.2312	3	243.4045	241.3952	from previous results			2	251.3345	241.3952				1	269.8445	236.2352			
193	255.5553	2	243.4045	241.3952				1	243.4045	244.8352				1	243.4045	236.2352			
194	256.8794	3	243.4045	241.3952	from previous results			2	251.3345	241.3952				1	269.8445	236.2352			
195	258.2035	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	236.2352			
196	259.5276	1	243.4045	246.5552				1	243.4045	244.8352				1	243.4045	236.2352			
197	260.8518	3	243.4045	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
198	262.1759	3	243.4045	246.5552	from previous results			3	251.3345	246.5552	from previous results			2	243.4045	253.4352			

nn

		3%						5%						10%					
frame #	in .avi	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data
199	263.5	3	243.4045	246.5552	from previous results			2	243.4045	246.5552				1	243.4045	253.4352			
200	264.8241	3	243.4045	246.5552	from previous results			2	243.4045	246.5552				1	243.4045	253.4352			
201	266.1482	2	243.4045	241.3952				1	243.4045	244.8352				1	243.4045	236.2352			
202	267.4724	3	243.4045	241.3952	from previous results			2	259.2645	231.0752				1	269.8445	236.2352			
203	268.7965	3	243.4045	241.3952	from previous results			2	259.2645	231.0752				1	269.8445	236.2352			
204	270.1206	3	243.4045	241.3952	from previous results			2	259.2645	231.0752				1	269.8445	236.2352			
205	271.4447	2	243.4045	231.0752				1	256.6245	227.6352				1	243.4045	236.2352			
206	272.7688	3	243.4045	231.0752	from previous results			2	259.2645	236.2352				1	269.8445	236.2352			
207	274.093	3	243.4045	231.0752	from previous results			2	259.2645	236.2352				1	269.8445	236.2352			
208	275.4171	3	243.4045	231.0752	from previous results			2	259.2645	236.2352				1	269.8445	236.2352			
209	276.7412	3	243.4045	231.0752	from previous results			2	259.2645	236.2352				1	269.8445	236.2352			
210	278.0653	3	243.4045	231.0752	from previous results			2	259.2645	236.2352				1	269.8445	236.2352			
211	279.3894	3	243.4045	231.0752	from previous results			2	259.2645	236.2352				1	269.8445	236.2352			
212	280.7136	3	243.4045	231.0752	from previous results			3	259.2645	236.2352	from previous results			2	269.8445	236.2352			
213	282.0377	3	243.4045	231.0752	from previous results			3	259.2645	236.2352	from previous results			2	269.8445	236.2352			
214	283.3618	3	243.4045	231.0752	from previous results			3	259.2645	236.2352	from previous results			2	269.8445	236.2352			
215	284.6859	3	243.4045	231.0752	from previous results			3	259.2645	236.2352	from previous results			2	269.8445	236.2352			
216	286.0101	3	243.4045	231.0752	from previous results			3	259.2645	236.2352	from previous results			2	269.8445	236.2352			
217	287.3342	3	243.4045	231.0752	from previous results			2	259.2645	236.2352				1	269.8445	236.2352			
218	288.6583	3	243.4045	231.0752	from previous results			3	259.2645	236.2352	from previous results			2	269.8445	236.2352			
219	289.9824	3	243.4045	231.0752	from previous results			3	259.2645	236.2352	from previous results			2	269.8445	236.2352			
220	291.3065	3	243.4045	231.0752	from previous results			3	259.2645	236.2352	from previous results			2	269.8445	236.2352			
221	292.6307	3	243.4045	231.0752	from previous results			3	259.2645	236.2352	from previous results			2	269.8445	236.2352			
222	293.9548	3	243.4045	231.0752	from previous results			3	259.2645	236.2352	from previous results			2	269.8445	236.2352			
223	295.2789	3	243.4045	231.0752	from previous results			3	259.2645	236.2352	from previous results			2	269.8445	236.2352			
224	296.603	3	243.4045	231.0752	from previous results			2	259.2645	236.2352				1	269.8445	236.2352			
225	297.9271	3	243.4045	231.0752	from previous results			2	259.2645	236.2352				1	269.8445	236.2352			
226	299.2513	3	243.4045	231.0752	from previous results			2	259.2645	236.2352				1	269.8445	236.2352			

oo

		3%						5%						10%					
frame #	in .avi	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data
227	300.5754	3	243.4045	231.0752	from previous results			3	259.2645	236.2352	from previous results			2	243.4045	236.2352			
228	301.8995	3	243.4045	231.0752	from previous results			3	259.2645	236.2352	from previous results			2	243.4045	236.2352			
229	303.2236	3	243.4045	231.0752	from previous results			2	259.2645	236.2352				1	269.8445	236.2352			
230	304.5477	3	243.4045	231.0752	from previous results			2	235.4745	246.5552				1	243.4045	253.4352			
231	305.8719	3	243.4045	231.0752	from previous results			2	243.4045	246.5552				1	243.4045	253.4352			
232	307.196	2	243.4045	246.5552				1	243.4045	253.4352				1	243.4045	253.4352			
233	308.5201	3	243.4045	246.5552	from previous results			2	259.2645	251.7152				1	269.8445	253.4352			
234	309.8442	3	243.4045	246.5552	from previous results			2	259.2645	251.7152				1	269.8445	253.4352			
235	311.1683	3	243.4045	246.5552	from previous results			2	259.2645	251.7152				1	269.8445	253.4352			
236	312.4925	3	243.4045	246.5552	from previous results			2	259.2645	246.5552				1	269.8445	253.4352			
237	313.8166	3	243.4045	246.5552	from previous results			2	259.2645	246.5552				1	269.8445	253.4352			
238	315.1407	3	243.4045	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
239	316.4648	3	243.4045	246.5552	from previous results			2	259.2645	246.5552				1	269.8445	253.4352			
240	317.7889	3	243.4045	246.5552	from previous results			2	259.2645	246.5552				1	269.8445	253.4352			
241	319.1131	3	243.4045	246.5552	from previous results			2	259.2645	246.5552				1	269.8445	253.4352			
242	320.4372	3	243.4045	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
243	321.7613	3	243.4045	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
244	323.0854	3	243.4045	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
245	324.4095	3	243.4045	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
246	325.7337	3	243.4045	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
247	327.0578	3	243.4045	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
248	328.3819	3	243.4045	246.5552	from previous results			2	243.4045	246.5552				1	269.8445	253.4352			
249	329.706	3	243.4045	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
250	331.0302	3	243.4045	246.5552	from previous results			2	251.3345	246.5552				1	269.8445	253.4352			
251	332.3543	2	235.4745	246.5552				1	243.4045	244.8352				1	243.4045	236.2352			
252	333.6784	2	243.4045	231.0752				1	243.4045	227.6352				1	243.4045	236.2352			
253	335.0025	3	191.6022	202.0782	319.3953	210.5063	closed 2	3	191.6022	202.0782	319.3953	210.5063	closed 2	3	191.6022	202.0782	319.3953	210.5063	closed 2
254	336.3266	3	183.0663	210.6293	310.836	201.962	closed 2	3	183.0663	210.6293	310.836	201.962	closed 2	3	183.0663	210.6293	310.836	201.962	closed 2

pp

		3%						5%						10%					
frame #	in .avi	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data
255	337.6508	3	174.5242	221.7013	310.8859	204.5049	closed 1	3	174.5242	221.7013	310.8859	204.5049	closed 1	3	174.5242	221.7013	310.8859	204.5049	closed 1
256	338.9749	3	243.4045	231.0752	from previous results			3	243.4045	227.6352	from previous results			3	243.4045	236.2352	from previous results		
257	340.299	3	157.3067	245.9583	318.9045	228.2704	closed 1	3	157.3067	245.9583	318.9045	228.2704	closed 1	3	157.3067	245.9583	318.9045	228.2704	closed 1
258	341.6231	3	157.5478	247.0822	318.8139	229.4756	closed 1	3	157.5478	247.0822	318.8139	229.4756	closed 1	3	157.5478	247.0822	318.8139	229.4756	closed 1
259	342.9472	3	157.7233	246.9117	301.6085	229.3386	closed 1	3	157.7233	246.9117	301.6085	229.3386	closed 1	3	157.7233	246.9117	301.6085	229.3386	closed 1
260	344.2714	3	157.965	246.5215	301.5221	229.0235	closed 1	3	157.965	246.5215	301.5221	229.0235	closed 1	3	157.965	246.5215	301.5221	229.0235	closed 1
261	345.5955	3	158.0689	246.7009	310.0016	229.1891	closed 1	3	158.0689	246.7009	310.0016	229.1891	closed 1	3	158.0689	246.7009	310.0016	229.1891	closed 1
262	346.9196	3	158.2473	246.8386	301.2844	229.3221	closed 1	3	158.2473	246.8386	301.2844	229.3221	closed 1	3	158.2473	246.8386	301.2844	229.3221	closed 1
263	348.2437	3	158.3313	246.9367	309.7942	229.4139	closed 1	3	158.3313	246.9367	309.7942	229.4139	closed 1	3	158.3313	246.9367	309.7942	229.4139	closed 1
264	349.5678	3	158.4909	247.1263	292.4624	229.5897	closed 1	3	158.4909	247.1263	292.4624	229.5897	closed 1	3	158.4909	247.1263	292.4624	229.5897	closed 1
265	350.892	3	158.5632	247.1414	292.4062	229.6068	closed 1	3	158.5632	247.1414	292.4062	229.6068	closed 1	3	158.5632	247.1414	292.4062	229.6068	closed 1
266	352.2161	3	158.7274	247.2237	292.2653	229.6891	closed 1	3	158.7274	247.2237	292.2653	229.6891	closed 1	3	158.7274	247.2237	292.2653	229.6891	closed 1
267	353.5402	3	158.7809	247.2414	292.2245	229.7069	closed 1	3	158.7809	247.2414	292.2245	229.7069	closed 1	3	158.7809	247.2414	292.2245	229.7069	
268	354.8643	3	243.4045	231.0752	from previous results			3	243.4045	227.6352	from previous results			3	243.4045	236.2352	from previous results		
269	356.1884	3	243.4045	231.0752	from previous results			3	243.4045	227.6352	from previous results			3	243.4045	236.2352	from previous results		
270	357.5126	3	243.4045	231.0752	from previous results			3	243.4045	227.6352	from previous results			3	243.4045	236.2352	from previous results		
271	358.8367	3	243.4045	231.0752	from previous results			3	243.4045	227.6352	from previous results			3	243.4045	236.2352	from previous results		
272	360.1608	3	243.4045	231.0752	from previous results			3	243.4045	227.6352	from previous results			3	243.4045	236.2352	from previous results		closed 1
273	361.4849	3	159.2922	238.8904	291.7877	229.9585	closed 1	3	159.2922	238.8904	291.7877	229.9585	closed 1	3	159.2922	238.8904	291.7877	229.9585	closed 2
274	362.809	3	185.2798	221.6871	308.9148	221.4015	closed 2	3	185.2798	221.6871	308.9148	221.4015	closed 2	3	185.2798	221.6871	308.9148	221.4015	closed 2
275	364.1332	3	194.0022	219.0949	309.193	219.1103	closed 2	3	194.0022	219.0949	309.193	219.1103	closed 2	3	194.0022	219.0949	309.193	219.1103	closed 2
276	365.4573	3	185.4567	211.2715	309.379	220.071	closed 2	3	185.4567	211.2715	309.379	220.071	closed 2	3	185.4567	211.2715	309.379	220.071	closed 2
277	366.7814	3	245.5481	202.4252	223.2967	202.8112	closed 2	2	235.4745	200.1152				1	243.4045	201.8352			
278	368.1055	2	235.4745	220.7552				2	235.4745	220.7552				1	243.4045	219.0352			
279	369.4296	2	235.4745	225.9152				1	243.4045	227.6352				1	243.4045	236.2352			
280	370.7538	2	235.4745	231.0752				2	235.4745	231.0752				1	243.4045	236.2352			
281	372.0779	1	235.4745	236.2352				1	230.1845	236.2352				1	243.4045	236.2352			
282	373.402	2	235.4745	231.0752				2	235.4745	231.0752				1	243.4045	236.2352			

qq

		3%						5%						10%					
frame #	in .avi	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data
283	374.7261	2	235.4745	236.2352				1	243.4045	236.2352				1	243.4045	236.2352			
284	376.0503	2	235.4745	231.0752				2	235.4745	231.0752				1	243.4045	236.2352			
285	377.3744	2	235.4745	241.3952				1	243.4045	244.8352				1	243.4045	253.4352			
286	378.6985	2	235.4745	241.3952				1	243.4045	244.8352				1	243.4045	253.4352			
287	380.0226	2	243.4045	241.3952				1	243.4045	244.8352				1	243.4045	253.4352			
288	381.3467	2	235.4745	241.3952				1	243.4045	244.8352				1	243.4045	253.4352			
289	382.6709	2	235.4745	241.3952				1	243.4045	244.8352				1	243.4045	253.4352			
290	383.995	1	243.4045	236.2352				1	243.4045	236.2352				1	243.4045	236.2352			
291	385.3191	2	243.4045	231.0752				2	243.4045	231.0752				1	243.4045	236.2352			
292	386.6432	1	243.4045	220.7552				1	243.4045	219.0352				1	243.4045	219.0352			
293	387.9673	2	251.3345	200.1152				2	251.3345	200.1152				1	243.4045	201.8352			
294	389.2915	3	183.7668	210.6053	310.595	210.6708	closed 2	3	183.7668	210.6053	310.595	210.6708	closed 2	3	183.7668	210.6053	310.595	210.6708	closed 2
295	390.6156	3	192.2716	202.0281	319.1931	210.6494	closed 2	3	192.2716	202.0281	319.1931	210.6494	closed 2	3	192.2716	202.0281	319.1931	210.6494	closed 2
296	391.9397	3	183.6412	210.5701	319.2527	210.6169	closed 1	3	183.6412	210.5701	319.2527	210.6169	closed 1	3	183.6412	210.5701	319.2527	210.6169	closed 1
297	393.2638	3	192.1524	202.0004	319.2762	210.615	closed 2	3	192.1524	202.0004	319.2762	210.615	closed 2	3	192.1524	202.0004	319.2762	210.615	closed 2
298	394.5879	3	192.0693	201.9752	319.3348	210.5838	closed 2	3	192.0693	201.9752	319.3348	210.5838	closed 2	3	192.0693	201.9752	319.3348	210.5838	closed 2
299	395.9121	3	192.0438	201.9765	319.339	202.0196	closed 2	3	192.0438	201.9765	319.339	202.0196	closed 2	3	192.0438	201.9765	319.339	202.0196	closed 2
300	397.2362	3	191.9624	201.9528	327.9658	210.5406	closed 1	3	191.9624	201.9528	327.9658	210.5406	closed 1	3	191.9624	201.9528	327.9658	210.5406	closed 1
301	398.5603	3	191.9453	201.9564	319.4077	201.9976	closed 2	3	191.9453	201.9564	319.4077	201.9976	closed 2	3	191.9453	201.9564	319.4077	201.9976	closed 2
302	399.8844	3	191.866	201.9342	319.4635	201.9726	closed 2	3	191.866	201.9342	319.4635	201.9726	closed 2	3	191.866	201.9342	319.4635	201.9726	closed 2
303	401.2085	3	191.8566	201.9397	328.0355	210.5241	closed 1	3	191.8566	201.9397	328.0355	210.5241	closed 1	3	191.8566	201.9397	328.0355	210.5241	closed 1
304	402.5327	3	200.3259	201.9128	328.0869	210.4986	closed 2	3	200.3259	201.9128	328.0869	210.4986	closed 2	3	200.3259	201.9128	328.0869	210.4986	closed 2
305	403.8568	3	157.6313	227.5983	319.5251	201.965	closed 1	3	157.6313	227.5983	319.5251	201.965	closed 1	3	157.6313	227.5983	319.5251	201.965	closed 1
306	405.1809	3	200.2467	201.9011	319.5893	210.4993	closed 2	3	200.2467	201.9011	319.5893	210.4993	closed 2	3	200.2467	201.9011	319.5893	210.4993	closed 2
307	406.505	3	191.7065	201.9152	311.026	201.9696	closed 2	3	191.7065	201.9152	311.026	201.9696	closed 2	3	191.7065	201.9152	311.026	201.9696	closed 2
308	407.8291	3	191.6354	201.8981	319.6362	210.4881	closed 2	3	191.6354	201.8981	319.6362	210.4881	closed 2	3	191.6354	201.8981	319.6362	210.4881	closed 2
309	409.1533	3	183.1149	210.4574	319.6301	210.4992	closed 1	3	183.1149	210.4574	319.6301	210.4992	closed 1	3	183.1149	210.4574	319.6301	210.4992	closed 1
310	410.4774	3	191.5765	201.8917	319.6772	210.4801	closed 2	3	191.5765	201.8917	319.6772	210.4801	closed 2	3	191.5765	201.8917	319.6772	210.4801	closed 2

rr

		3%						5%						10%					
frame #	in .avi	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data
311	411.8015	3	183.0623	210.4497	319.6678	210.4919	closed 1	3	183.0623	210.4497	319.6678	210.4919	closed 1	3	183.0623	210.4497	319.6678	210.4919	closed 1
312	413.1256	3	200.0638	201.8821	319.7125	210.4751	closed 2	3	200.0638	201.8821	319.7125	210.4751	closed 2	3	200.0638	201.8821	319.7125	210.4751	closed 2
313	414.4497	3	200.082	201.8919	319.7005	210.4873	closed 2	3	200.082	201.8919	319.7005	210.4873	closed 2	3	200.082	201.8919	319.7005	210.4873	closed 2
314	415.7739	3	200.0194	201.8802	319.7426	210.4726	closed 2	3	200.0194	201.8802	319.7426	210.4726	closed 2	3	200.0194	201.8802	319.7426	210.4726	closed 2
315	417.098	3	182.9786	210.4411	319.7283	210.4849	closed 1	3	182.9786	210.4411	319.7283	210.4849	closed 1	3	182.9786	210.4411	319.7283	210.4849	closed 1
316	418.4221	3	208.5179	201.8751	319.7678	210.4723	closed 2	3	208.5179	201.8751	319.7678	210.4723	closed 2	3	208.5179	201.8751	319.7678	210.4723	closed 2
317	419.7462	3	208.5432	201.8848	319.7516	210.4846	closed 2	3	208.5432	201.8848	319.7516	210.4846	closed 2	3	208.5432	201.8848	319.7516	210.4846	closed 2
318	421.0704	3	182.8953	210.4293	319.7883	210.4739	closed 1	3	182.8953	210.4293	319.7883	210.4739	closed 1	3	182.8953	210.4293	319.7883	210.4739	closed 1
319	422.3945	3	157.3397	227.5345	319.7705	210.4859	closed 1	3	157.3397	227.5345	319.7705	210.4859	closed 1	3	157.3397	227.5345	319.7705	210.4859	closed 1
320	423.7186	3	191.3947	201.8888	319.8047	210.4771	closed 2	3	191.3947	201.8888	319.8047	210.4771	closed 2	3	191.3947	201.8888	319.8047	210.4771	closed 2
321	425.0427	3	199.9572	201.8931	319.7856	210.4887	closed 2	3	199.9572	201.8931	319.7856	210.4887	closed 2	3	199.9572	201.8931	319.7856	210.4887	closed 2
322	426.3668	3	182.8563	210.4339	319.817	210.4816	closed 1	3	182.8563	210.4339	319.817	210.4816	closed 1	3	182.8563	210.4339	319.817	210.4816	closed 1
323	427.691	3	208.4749	201.8916	319.7971	210.4926	closed 2	3	208.4749	201.8916	319.7971	210.4926	closed 2	3	208.4749	201.8916	319.7971	210.4926	closed 2
324	429.0151	3	199.8982	201.8919	319.8258	210.4872	closed 2	3	199.8982	201.8919	319.8258	210.4872	closed 2	3	199.8982	201.8919	319.8258	210.4872	closed 2
325	430.3392	3	191.4075	210.4421	319.8052	210.4975	closed 2	3	191.4075	210.4421	319.8052	210.4975	closed 2	3	191.4075	210.4421	319.8052	210.4975	closed 2
326	431.6633	3	191.3582	201.9017	319.8313	210.4936	closed 2	3	191.3582	201.9017	319.8313	210.4936	closed 2	3	191.3582	201.9017	319.8313	210.4936	closed 2
327	432.9874	3	182.8674	210.4514	319.8103	210.5031	closed 1	3	182.8674	210.4514	319.8103	210.5031	closed 1	3	182.8674	210.4514	319.8103	210.5031	closed 1
328	434.3116	3	208.4203	201.8981	319.8338	210.5006	closed 2	3	208.4203	201.8981	319.8338	210.5006	closed 2	3	208.4203	201.8981	319.8338	210.5006	closed 2
329	435.6357	3	182.8649	210.4561	319.8126	210.5093	closed 1	3	182.8649	210.4561	319.8126	210.5093	closed 1	3	182.8649	210.4561	319.8126	210.5093	closed 1
330	436.9598	3	191.3564	201.913	319.8335	210.508	closed 2	3	191.3564	201.913	319.8335	210.508	closed 2	3	191.3564	201.913	319.8335	210.508	closed 2
331	438.2839	3	191.3872	201.9193	319.8123	210.5159	closed 2	3	191.3872	201.9193	319.8123	210.5159	closed 2	3	191.3872	201.9193	319.8123	210.5159	closed 2
332	439.608	3	191.3611	201.919	319.8308	210.5156	closed 2	3	191.3611	201.919	319.8308	210.5156	closed 2	3	191.3611	201.919	319.8308	210.5156	closed 2
333	440.9322	3	199.925	201.92	319.8098	210.5227	closed 2	3	199.925	201.92	319.8098	210.5227	closed 2	3	199.925	201.92	319.8098	210.5227	closed 2
334	442.2563	3	191.3688	201.9251	319.8259	210.5233	closed 2	3	191.3688	201.9251	319.8259	210.5233	closed 2	3	191.3688	201.9251	319.8259	210.5233	closed 2
335	443.5804	3	191.3986	201.9301	319.7932	201.9828	closed 2	3	191.3986	201.9301	319.7932	201.9828	closed 2	3	191.3986	201.9301	319.7932	201.9828	closed 2
336	444.9045	3	182.8462	201.9359	319.8191	210.531	closed 1	3	182.8462	201.9359	319.8191	210.531	closed 1	3	182.8462	201.9359	319.8191	210.531	closed 1
337	446.2286	3	174.3403	201.9455	319.7869	201.9893	closed 1	3	174.3403	201.9455	319.7869	201.9893	closed 1	3	174.3403	201.9455	319.7869	201.9893	closed 1
338	447.5528	3	174.3253	201.9469	319.7985	201.9919	closed 1	3	174.3253	201.9469	319.7985	201.9919	closed 1	3	174.3253	201.9469	319.7985	201.9919	closed 1

SS

		3%						5%						10%					
frame #	in .avi	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data
339	448.8769	3	174.3514	201.9508	319.779	201.9956	closed 1	3	174.3514	201.9508	319.779	201.9956	closed 1	3	174.3514	201.9508	319.779	201.9956	closed 1
340	450.201	3	182.8731	201.9476	311.2497	202.0159	closed 2	3	182.8731	201.9476	311.2497	202.0159	closed 2	3	182.8731	201.9476	311.2497	202.0159	closed 2
341	451.5251	2	243.4045	200.1152				2	243.4045	200.1152				1	243.4045	201.8352			
342	452.8492	2	235.4745	215.5952				2	235.4745	215.5952				1	243.4045	219.0352			
343	454.1734	2	235.4745	225.9152				2	235.4745	225.9152				1	243.4045	219.0352			
344	455.4975	2	235.4745	231.0752				2	235.4745	231.0752				1	243.4045	219.0352			
345	456.8216	2	235.4745	231.0752				2	235.4745	231.0752				1	243.4045	219.0352			
346	458.1457	2	235.4745	231.0752				2	235.4745	231.0752				1	243.4045	219.0352			
347	459.4698	1	235.4745	236.2352				1	230.1845	236.2352				1	243.4045	236.2352			
348	460.794	2	235.4745	231.0752				2	235.4745	231.0752				1	243.4045	219.0352			
349	462.1181	1	235.4745	236.2352				1	230.1845	236.2352				1	243.4045	236.2352			
350	463.4422	2	235.4745	231.0752				2	235.4745	231.0752				1	243.4045	219.0352			
351	464.7663	2	235.4745	231.0752				2	235.4745	231.0752				1	243.4045	219.0352			
352	466.0905	1	235.4745	236.2352				1	230.1845	236.2352				1	243.4045	236.2352			
353	467.4146	2	235.4745	231.0752				2	235.4745	231.0752				1	243.4045	219.0352			
354	468.7387	2	235.4745	231.0752				2	235.4745	231.0752				1	243.4045	219.0352			
355	470.0628	2	235.4745	231.0752				2	235.4745	231.0752				1	243.4045	219.0352			
356	471.3869	1	235.4745	236.2352				1	230.1845	236.2352				1	243.4045	236.2352			
357	472.7111	2	235.4745	241.3952				1	230.1845	244.8352				1	243.4045	236.2352			
358	474.0352	2	235.4745	231.0752				2	235.4745	231.0752				1	243.4045	219.0352			
359	475.3593	3	235.4745	231.0752	from previous results			2	243.4045	241.3952				1	243.4045	236.2352			
360	476.6834	1	235.4745	236.2352				1	230.1845	236.2352				1	243.4045	236.2352			
361	478.0075	2	219.6145	205.2752				1	216.9645	210.4352				1	216.9645	219.0352			
362	479.3317	3	208.6962	193.4288	225.6255	193.6893	closed 2	2	219.6145	194.9552				1	216.9645	201.8352			
363	480.6558	3	208.7064	193.4276	225.6026	185.1333	closed 2	2	219.6145	189.7952				1	216.9645	201.8352			
364	481.9799	3	208.7169	193.4297	225.5924	185.1355	closed 2	2	219.6145	189.7952				1	216.9645	201.8352			
365	483.304	2	219.6145	184.6352				1	216.9645	184.6352				1	216.9645	184.6352			
366	484.6281	1	219.6145	184.6352				1	216.9645	184.6352				1	216.9645	184.6352			

tt

		3%						5%						10%					
frame #	in .avi	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data
367	485.9523	1	219.6145	184.6352				1	216.9645	184.6352				1	216.9645	184.6352			
368	487.2764	1	219.6145	184.6352				1	216.9645	184.6352				1	216.9645	184.6352			
369	488.6005	1	219.6145	184.6352				1	216.9645	184.6352				1	216.9645	184.6352			
370	489.9246	2	211.6845	189.7952				1	203.7445	193.2352				1	216.9645	201.8352			
371	491.2487	3	208.766	184.8831	225.5298	185.1299	closed 2	2	219.6145	184.6352				1	216.9645	184.6352			
372	492.5729	2	211.6845	184.6352				1	203.7445	184.6352				1	216.9645	184.6352			
373	493.897	3	208.7819	184.8827	225.5135	185.1285	closed 2	2	219.6145	184.6352				1	216.9645	184.6352			
374	495.2211	3	208.7953	184.8835	225.5002	185.1291	closed 2	2	219.6145	184.6352				1	216.9645	184.6352			
375	496.5452	2	211.6845	184.6352				1	203.7445	184.6352				1	216.9645	184.6352			
376	497.8693	1	219.6145	184.6352				1	216.9645	184.6352				1	216.9645	184.6352			
377	499.1935	2	219.6145	184.6352				1	216.9645	184.6352				1	216.9645	184.6352			
378	500.5176	2	219.6145	184.6352				1	216.9645	184.6352				1	216.9645	184.6352			
379	501.8417	1	219.6145	184.6352				1	216.9645	184.6352				1	216.9645	184.6352			
380	503.1658	3	200.3039	193.4352	216.9072	185.1426	closed 2	2	211.6845	189.7952				1	190.5245	201.8352			
381	504.4899	2	203.7545	189.7952				1	203.7445	193.2352				1	190.5245	201.8352			
382	505.8141	3	200.3153	193.4347	216.8945	185.1408	closed 2	2	211.6845	189.7952				1	190.5245	201.8352			
383	507.1382	3	200.3128	193.4337	216.8968	185.1395	closed 2	2	211.6845	189.7952				1	190.5245	201.8352			
384	508.4623	2	211.6845	184.6352				1	203.7445	184.6352				1	216.9645	184.6352			
385	509.7864	3	217.4032	184.8733	242.5365	185.082	closed 2	2	227.5445	184.6352				1	216.9645	184.6352			
386	511.1106	3	208.8695	184.8787	225.4229	185.1184	closed 2	2	219.6145	184.6352				1	216.9645	184.6352			
387	512.4347	3	208.8654	184.878	225.4268	185.1175	closed 2	2	219.6145	184.6352				1	216.9645	184.6352			
388	513.7588	3	208.878	184.8779	225.4138	185.1167	closed 2	2	219.6145	184.6352				1	216.9645	184.6352			
389	515.0829	2	211.6845	184.6352				1	203.7445	184.6352				1	216.9645	184.6352			
390	516.407	2	219.6145	184.6352				1	216.9645	184.6352				1	216.9645	184.6352			
391	517.7312	3	208.8926	193.4262	225.411	185.1146	closed 2	2	219.6145	189.7952				1	216.9645	201.8352			
392	519.0553	3	191.8092	193.438	216.8477	185.1323	closed 2	2	203.7545	189.7952				1	190.5245	201.8352			
393	520.3794	1	219.6145	184.6352				1	216.9645	184.6352				1	216.9645	184.6352			
394	521.7035	3	208.8974	184.8757	225.3928	185.1123	closed 2	2	219.6145	184.6352				1	216.9645	184.6352			

uu

		3%						5%						10%					
frame #	in .avi	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data	ORIGIN	X	Y	X	Y	extra data
395	523.0276	2	219.6145	184.6352				1	216.9645	184.6352				1	216.9645	184.6352			
396	524.3518	3	183.2706	193.4433	216.8364	185.1296	closed 2	3	183.2706	193.4433	216.8364	185.1296	closed 2	2	190.5245	184.6352			
397	525.6759	3	183.2646	193.4428	216.8431	185.1295	closed 2	3	183.2646	193.4428	216.8431	185.1295	closed 2	2	190.5245	184.6352			
398	527	3	208.9054	184.8745	225.3837	185.1099	closed 2	2	219.6145	184.6352				1	216.9645	184.6352			

Closed1 & Closed2: indicate at which point of the algorithms the *close state* is detected

INCORRECT STATE

CORRECT CLOSE STATE DATA ALTHOUGH THE ALGORITHMS DID NOT DETECT IT

vv

ww

Appendix IV-I

Legend:

m:	male	c:	Caucasian	a:	African
f:	female	h:	Hispanic	as:	Asian

OK	Correct detection and correct segmentation
KO	Neither correct detection nor correct segmentation
~	Correct detection but no correct segmentation
	Not evaluated

TEST:
Segmentation algorithm (I)

Video sequence	Characteristics	Lips	Teeth	Darkness	Notes
1	f/c	OK		OK	
2	m/c	OK		OK	not well focused
3	m/c	OK	~	OK	
4	f/c	KO	KO	OK	tanned
5	f/c	~	~	OK	tanned / <50%
6	m/c	KO	KO	OK	
7	m/h	OK		~	~beard
8	f/c	OK	KO	OK	
9					
10	f/c	KO	~	OK	
11	f/c	~	KO	~	tongue / bad quality
12	m/a				it is out of track
13	f/c	OK	~	OK	
14	m/c	~	KO	OK	
15	f/h	OK		OK	
16	m/c	OK	OK	~	
17	m/c	OK		OK	
18	f/h	OK		OK	
19	f/c	OK	KO	OK	
20	f/c	~	KO	OK	tanned/pale lips
21	m/h	~		~	
22	m/c	OK	~	OK	
23	m/c	OK	KO	OK	
24	m/c	~	KO	OK	
25	f/c	~		OK	
26	f/c	OK	~	KO	
27	f/c	OK	~	OK	

Video sequence	Characteristics	Lips	Teeth	Darkness	Notes
28	m/c	OK	~	OK	
29	m/c	OK	~	OK	
30	m/c	~	~	OK	
31	m/c	OK	KO	~	beard
32	m/c	OK		~	beard
33	m/c	OK	~	~	moustache
34	m/c	OK	~	OK	
35	m/c	~	KO	~	tanned/beard
36	f/c	OK	KO	~	tanned
37	m/c	~	KO	~	white beard
38	f/c	KO	KO	OK	tanned
39	f/c	~	KO	OK	
40	m/a	KO	KO	KO	
41	m/c	~		KO	tanned
42	f/c	~	KO	OK	
43	f/c	~	KO	OK	
44	f/c	OK	KO	OK	
45	f/c	OK	KO	OK	
46	f/c	OK	KO	OK	tanned/moustache
47	m/c	~	KO	OK	
48	f/c	OK		OK	
49	f/c	~	KO	OK	
50	m/c	OK	KO	OK	moustache
51	m/c	OK		OK	
52	m/c	OK	KO	KO	beard
53	m/c	OK		KO	
54	f/c	KO	KO	OK	
55	f/c	OK		OK	
56	f/c	OK		OK	
57					
58	f/c	OK		~	
59	m/c	OK		KO	
60	m/c	OK	KO	~	beard
61	m/c	~	KO	~	beard
62	m/c	~	KO	KO	moustache

Test:
segmentation algorithm (II)

Video sequence	Characteristics	Lips	Teeth	Darkness	Notes
1	f/c	OK	OK	OK	
2	m/c	OK	OK	OK	not well focused
3	m/c	OK	OK	OK	
4	f/c	~	OK	OK	tanned
5	f/c	~	OK	OK	tanned / <50%
6	m/c	KO	KO	OK	white beard
7	m/h	OK		~	~beard
8	f/c	~	OK	OK	
9		KO	~	OK	
10	f/c	~	OK	OK	
11	f/c	~	OK	OK	tongue / bad quality
12	m/a				it is out of track
13	f/c	OK	OK	OK	
14	m/c	~	OK	OK	
15	f/h	OK		OK	very small
16	m/c	~	~	OK	
17	m/c	OK		OK	
18	f/h	OK		OK	
19	f/c	OK	OK	~	tanned
20	f/c	~	OK	~	tanned/pale lips
21	m/h	~		OK	
22	m/c	OK	OK	OK	
23	m/c	OK	OK	OK	
24	m/c	OK	OK	OK	
25	f/c	KO	~	~	too small
26	f/c	OK	OK	~	
27	f/c	OK	OK	OK	
28	m/c	OK	OK	OK	
29	m/c	OK	OK	OK	
30	m/c	OK	~	~	
31	m/c	OK	OK	OK	beard
32	m/c	OK	OK	OK	beard
33	m/c	~	OK	OK	moustache
34	m/c	OK	OK	OK	

zz

Video sequence	Characteristics	Lips	Teeth	Darkness	Notes
35	m/c	OK	OK	OK	tanned/beard
36	f/c	OK	OK	~	tanned
37	m/c	KO	~	KO	white beard
38	f/c	KO	OK	OK	tanned
39	f/c	~	OK	OK	
40	m/a	KO	KO	KO	very dark skin
41	m/c	~		OK	tanned
42	f/c	~		OK	
43	f/c	KO	~	OK	
44	f/c	OK	OK	OK	
45	f/c	OK	OK	OK	
46	f/c	OK	OK	OK	tanned/moustache
47	m/c	KO	~	OK	
48	f/c	OK		OK	
49	f/c	OK		OK	
50	m/c	OK		OK	moustache
51	m/c	OK		OK	
52	m/c	OK	OK	OK	beard
53	m/c	OK		OK	
54	f/c	KO	KO	KO	
55	f/c	OK	~	OK	
56	f/c	OK		OK	
57					
58	f/c	OK		~	
59	m/c	OK		OK	
60	m/c	OK	OK	OK	beard
61	m/c	~	OK	OK	beard
62	m/c	OK	OK	OK	moustache

Appendix VI-J

BIFS Syntax for Face Animation

Face {			
exposedField	SFNode	**fit**	NULL
exposedField	SFNode	**fdp**	NULL
exposedField	SFNode	**fap**	NULL
exposedField	SFNode	**ttsSource**	NULL
exposedField	MFNode	**renderedFace**	NULL
}			

NOTE — For the binary encoding of this node see Document MPEG-4 NODES A.1.36.

Functionality and semantics

The **Face** node is used to define and animate a face in the scene. In order to animate the face with a facial animation stream, it is necessary to link the **Face** node to a BIFS-Anim stream. The node shall be assigned a nodeID, through the DEF mechanism. Then, as for any BIFS-Anim stream, an animation mask is sent in the object descriptor of the BIFS-Anim stream (specificInfo field). The animation mask points to the **Face** node using its nodeID. The terminal shall then connect the facial animation decoder to the appropriate **Face** node.

The **FAP** field shall contain a **FAP** node, describing the facial animation parameters (FAPs). Each **Face** node shall contain a non-NULL **FAP** field.

The **FDP** field, which defines the particular look of a face by means of downloading the position of face definition points or an entire model, is optional. If the **FDP** field is not specified, the default face model of the terminal shall be used.

The **FIT** field, when specified, allows a set of FAPs to be defined in terms of another set of FAPs. When this field is non-NULL, the terminal shall use **FIT** to compute the maximal set of FAPs before using the FAPs to compute the mesh.

The **ttsSource** field shall only be non-NULL if the facial animation is to determine the facial animation parameters from an audio TTS source (see ISO/IEC 14496-3, section 6). In this case the **ttsSource** field shall contain an **AudioSource** node and the face shall be animated using the phonemes and bookmarks received from the TTS.

renderedFace is the scene graph of the face after it is rendered (all FAP's applied).

FaceDefMesh

FaceDefMesh {			
field	SFNode	**faceSceneGraphNode**	NULL
field	MFInt32	**intervalBorders**	[]
field	MFInt32	**coordIndex**	[]
field	MFVec3f	**displacements**	[]
}			

NOTE — For the binary encoding of this node see Document MPEG-4 NODES A.1.37.

The **FaceDefMesh** node allows for the deformation of an **IndexedFaceSet** as a function of the amplitude of a FAP as specified in the related **FaceDefTable** node. The **FaceDefMesh** node defines the piece-wise linear motion trajectories for vertices of the **faceSceneGraphNode** field, which shall contain an **IndexedFaceSet** node. This **IndexedFaceSet** node belongs to the scenegraph of the **faceSceneGraph** field of the **FDP** node.

The **intervalBorders** field specifies interval borders for the piece-wise linear approximation in increasing order. Exactly one interval border shall have the value 0.

The **coordIndex** field shall contain a list of indices into the **Coordinate** node of the **IndexedFaceSet** node specified by the **faceSceneGraphNode** field.

For each vertex indexed in the **coordIndex** field, displacement vectors are given in the **displacements** field for the intervals defined in the **intervalBorders** field. There must be exactly (num(**intervalBorders**)-1)*num(**coordIndex**) values in this field.

In most cases, the animation generated by a FAP cannot be specified by updating a **Transform** node. Thus, a deformation of an **IndexedFaceSet** node needs to be performed. In this case, the **FaceDefTables** shall define which **IndexedFaceSets** are affected by a given FAP and how the **coord** fields of these nodes are updated. This is done by means of tables.

If a FAP affects an **IndexedFaceSet**, the **FaceDefMesh** shall specify a table of the following format for this **IndexedFaceSet**:

Table 0-1

VERTEX DISPLACEMENTS

Vertex no.	**1st Interval [I1, I2]**	**2nd Interval [I2, I3]**	**…**
Index 1	Displacement D11	Displacement D12	…
Index 2	Displacement D21	Displacement D22	…
…	…	…	…

Exactly one interval border I_k must have the value 0:
$[I_1, I_2], [I_2, I_3], \ldots[I_{k-1}, 0], [0, I_{k+1}], [I_{k+1}, I_{k+2}], \ldots[I_{max-1}, I_{max}]$
During animation, when the terminal receives a FAP, which affects one or more **IndexedFaceSets** of the face model, it shall piece-wise linearly approximate the motion trajectory of each vertex of the affected **IndexedFaceSets** by using the appropriate table.

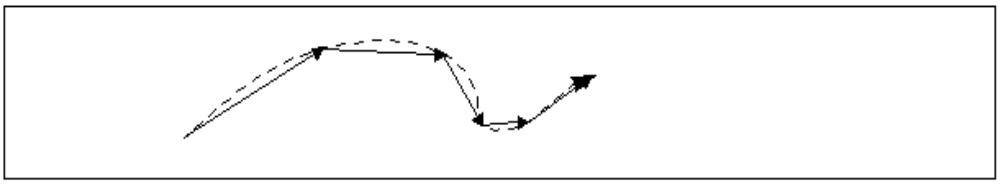


Figure 0-9 . An arbitrary motion trajectory is approximated as a piece-wise linear one.

If $\boldsymbol{P}_m$ is the position of the m^{th} vertex in the **IndexedFaceSet** in neutral state (FAP = 0), $\boldsymbol{P'}_m$ the position of the same vertex after animation with the given FAP and $\boldsymbol{D}_{mk}$ the 3D displacement in the k^{th} interval, the following algorithm shall be applied to determine the new position $\boldsymbol{P'}_m$.

Determine, in which of the intervals listed in the table the received FAP is lying.

If the received FAP is lying in the j^{th} interval [Ij, Ij+1] and 0=Ik ≤ Ij, the new vertex position P'm of the mth vertex of the **IndexedFaceSet** is given by:

$P'_m = FAPU * ((I_{k+1}-0) * D_{m,k} + (I_{k+2}-I_{k+1}) * D_{m,\,k+1} + \ldots + (I_j - I_{j-1}) * D_{m,\,j-1} + (FAP-I_j) * D_{m,\,j}) + P_m.$ (Eq. 1)

If FAP > I_{max}, then P'_m is calculated by using equation Eq. 1 and setting the index j = max.

If the received FAP is lying in the jth interval [I_j, I_{j+1}] and $I_{j+1} \leq I_k$=0, the new vertex position P'_m is given by:

$P'_m = FAPU * ((I_{j+1} - FAP) * D_{m,\,j} + (I_{j+2} - I_{j+1}) * D_{m,\,j+1} + \ldots + (I_{k-1} - I_{k-2}) * D_{m,\,k-2} + (0 - I_{k-1}) * D_{m,\,k-1}) + P_m$ (Eq. 2)

If FAP < I_1, then P'_m is calculated by using equation Eq. 1 and setting the index j+1 = 1.

If for a given FAP and **IndexedFaceSet** the table contains only one interval, the motion is strictly linear:

P'm = FAPU * FAP * Dm1 + Pm.

EXAMPLE —

```
FaceDefMesh {
   objectDescriptorID UpperLip
   intervalBorders [ -1000, 0, 500, 1000 ]
   coordIndex [ 50, 51]
   displacements [1 0 0,  0.9 0 0, 1.5 0 4, 0.8 0 0, 0.7 0 0, 2 0 0 ]
}
```

This **FaceDefMesh** defines the animation of the mesh "UpperLip". For the piecewise-linear motion function three intervals are defined: [-1000, 0], [0, 500] and [500, 1000]. Displacements are given for the vertices with the indices 50 and 51. The displacements for the vertex 50 are: (1 0 0), (0.9 0 0) and (1.5 0 4), the displacements for vertex 51 are (0.8 0 0), (0.7 0 0) and (2 0 0). Given a FAPValue of 600, the resulting displacement for vertex 50 would be:

displacement(vertex 50) = $500*(0.9\ 0\ 0)^T + 100 * (1.5\ 0\ 4)^T = (600\ 0\ 400)^T$.

If the FAPValue is outside the given intervals, the boundary intervals are extended to +I or -I, as appropriate.

FaceDefTables {

field	SFInt32	**fapID**	0
field	SFInt32	**highLevelSelect**	0
exposedField	MFNode	**faceDefMesh**	[]
exposedField	MFNode	**faceDefTransform**	[]

}

NOTE — For the binary encoding of this node see Document MPEG-4 NODES A.1.38.

Functionality and semantics

The **FaceDefTables** node defines the behavior of a facial animation parameter FAP on a downloaded face model in **faceSceneGraph** by specifying the displacement vectors for moved vertices inside **IndexedFaceSet** objects as a function of the FAP **fapID** and/or specifying the value of a field of a **Transform** node as a function of FAP **fapID**.

ddd

The **FaceDefTables** node is transmitted directly after the BIFS bitstream of the **FDP** node. The **FaceDefTables** lists all FAPs that animate the face model. The FAPs animate the downloaded face model by updating the **Transform** or **IndexedFaceSet** nodes of the scene graph in **faceSceneGraph**. For each listed FAP, the **FaceDefTables** node describes which nodes are animated by this FAP and how they are animated. All FAPs that occur in the bitstream have to be specified in the **FaceDefTables** node. The animation generated by a FAP can be specified either by updating a **Transform** node (using a **FaceDefTransform**), or as a deformation of an **IndexedFaceSet** (using a **FaceDefMesh**).

The FAPUs shall be calculated by the terminal using the feature points that shall be specified in the FDP. The FAPUs are needed in order to animate the downloaded face model.

Semantics

The **fapID** field specifies the FAP, for which the animation behavior is defined in the **faceDefMesh** and **faceDefTransform** fields.

If **fapID** has value 1 or 2, the **highLevelSelect** field specifies the type of viseme or expression. In other cases this field has no meaning and shall be ignored.

The **faceDefMesh** field shall contain a **FaceDefMesh** node.

The **faceDefTransform** field shall contain a **FaceDefTransform** node.

FaceDefTransform {

field	SFNode	**faceSceneGraphNode**	NULL
field	SFInt32	**fieldId**	1
field	SFRotation	**rotationDef**	0, 0, 1, 0
field	SFVec3f	**scaleDef**	1, 1, 1
field	SFVec3f	**translationDef**	0, 0, 0

}

NOTE — For the binary encoding of this node see Document MPEG-4 NODES A.1.39.

Functionality and semantics

The **FaceDefTransform** node defines which field (**rotation**, **scale** or **translation**) of a **Transform** node (**faceSceneGraphNode)** of **faceSceneGraph** (defined in an **FDP** node) is updated by a facial animation parameter, and how the field is updated. If the face is in its neutral position, the **faceSceneGraphNode** has its **translation**, **scale**, and **rotation** fields set to the neutral values $(0,0,0)^T$, $(1,1,1)^T$, (0,0,1,0), respectively.

The **faceSceneGraphNode** field specifies the **Transform** node for which the animation is defined. The node shall be part of **faceScenegraph** as defined in the **FDP** node.

The **fieldId** field specifies which field in the **Transform** node, specified by the **faceSceneGraphNode** field, is updated by the FAP during animation. Possible fields are **translation**, **rotation**, **scale**.

- If **fieldID**==1, **rotation** shall be updated using **rotationDef** and FAPValue.
- If **fieldID**==2, **scale** shall be updated using **scaleDef** and FAPValue.
- If **fieldID**==3, **translation** shall be updated using **translationDef** and FAPValue.

The **rotationDef** field is of type SFRotation. With **rotationDef**=(r_x,r_y,r_z,θ), the new value of the **rotation** field of the **Transform** node **faceSceneGraphNode** is:

rotation: =(r_x,r_y,r_z,θ*FAPValue*AU) [AU is defined in ISO/IEC FCD 14496-2]

The **scaleDef** field is of type SFVec3f. The new value of the **scale** field of the **Transform** node **faceSceneGraphNode** is:

scale:= FAPValue***scaleDef**

The **translationDef** field is of type SFVec3f. The new value of the **translation** field of the **Transform** node **faceSceneGraphNode** is:

translation:= FAPValue***translationDef**

FAP {

ExposedField	SFNode	**viseme**	NULL
ExposedField	SFNode	**expression**	NULL
exposedField	SFInt32	**open_jaw**	+I
exposedField	SFInt32	**lower_t_midlip**	+I
exposedField	SFInt32	**raise_b_midlip**	+I
exposedField	SFInt32	**stretch_l_corner**	+I
exposedField	SFInt32	**stretch_r_corner**	+I
exposedField	SFInt32	**lower_t_lip_lm**	+I
exposedField	SFInt32	**lower_t_lip_rm**	+I
exposedField	SFInt32	**lower_b_lip_lm**	+I
exposedField	SFInt32	**lower_b_lip_rm**	+I
exposedField	SFInt32	**raise_l_cornerlip**	+I
exposedField	SFInt32	**raise_r_cornerlip**	+I
exposedField	SFInt32	**thrust_jaw**	+I
exposedField	SFInt32	**shift_jaw**	+I
exposedField	SFInt32	**push_b_lip**	+I
exposedField	SFInt32	**push_t_lip**	+I
exposedField	SFInt32	**depress_chin**	+I
exposedField	SFInt32	**close_t_l_eyelid**	+I
exposedField	SFInt32	**close_t_r_eyelid**	+I
exposedField	SFInt32	**close_b_l_eyelid**	+I
exposedField	SFInt32	**close_b_r_eyelid**	+I
exposedField	SFInt32	**yaw_l_eyeball**	+I
exposedField	SFInt32	**yaw_r_eyeball**	+I
exposedField	SFInt32	**pitch_l_eyeball**	+I
exposedField	SFInt32	**pitch_r_eyeball**	+I
exposedField	SFInt32	**thrust_l_eyeball**	+I
exposedField	SFInt32	**thrust_r_eyeball**	+I
exposedField	SFInt32	**dilate_l_pupil**	+I
exposedField	SFInt32	**dilate_r_pupil**	+I
exposedField	SFInt32	**raise_l_i_eyebrow**	+I
exposedField	SFInt32	**raise_r_i_eyebrow**	+I
exposedField	SFInt32	**raise_l_m_eyebrow**	+I
exposedField	SFInt32	**raise_r_m_eyebrow**	+I
exposedField	SFInt32	**raise_l_o_eyebrow**	+I
exposedField	SFInt32	**raise_r_o_eyebrow**	+I
exposedField	SFInt32	**squeeze_l_eyebrow**	+I
exposedField	SFInt32	**squeeze_r_eyebrow**	+I

fff

exposedField	SFInt32	**puff_l_cheek**	+I
exposedField	SFInt32	**puff_r_cheek**	+I
exposedField	SFInt32	**lift_l_cheek**	+I
exposedField	SFInt32	**lift_r_cheek**	+I
exposedField	SFInt32	**shift_tongue_tip**	+I
exposedField	SFInt32	**raise_tongue_tip**	+I
exposedField	SFInt32	**thrust_tongue_tip**	+I
exposedField	SFInt32	**raise_tongue**	+I
exposedField	SFInt32	**tongue_roll**	+I
exposedField	SFInt32	**head_pitch**	+I
exposedField	SFInt32	**head_yaw**	+I
exposedField	SFInt32	**head_roll**	+I
exposedField	SFInt32	**lower_t_midlip_o**	+I
exposedField	SFInt32	**raise_b_midlip_o**	+I
exposedField	SFInt32	**stretch_l_cornerlip**	+I
exposedField	SFInt32	**stretch_r_cornerlip_o**	+I
exposedField	SFInt32	**lower_t_lip_lm_o**	+I
exposedField	SFInt32	**lower_t_lip_rm_o**	+I
exposedField	SFInt32	**raise_b_lip_lm_o**	+I
exposedField	SFInt32	**raise_b_lip_rm_o**	+I
exposedField	SFInt32	**raise_l_cornerlip_o**	+I
exposedField	SFInt32	**raise_r_cornerlip_o**	+I
exposedField	SFInt32	**stretch_l_nose**	+I
exposedField	SFInt32	**stretch_r_nose**	+I
exposedField	SFInt32	**raise_nose**	+I
exposedField	SFInt32	**bend_nose**	+I
exposedField	SFInt32	**raise_l_ear**	+I
exposedField	SFInt32	**raise_r_ear**	+I
exposedField	SFInt32	**pull_l_ear**	+I
exposedField	SFInt32	**pull_r_ear**	+I

}

NOTE — For the binary encoding of this node see Document MPEG-4 NODES A.1.40.

Functionality and semantics

This node defines the current look of the face by means of expressions and FAPs and gives a hint to TTS controlled systems on which viseme to use. For a definition of the facial animation parameters see ISO/IEC 14496-2, Annex C.

The **viseme** field shall contain a **Viseme** node.

The **expression** field shall contain an **Expression** node.

The semantics for the remaining fields are described in the ISO/IEC 14496-2, Annex C and in particular in Table C-1.

A FAP of value +I shall be interpreted as indicating that the particular FAP is uninitialized.

FDP {

exposedField	SFNode	**featurePointsCoord**	NULL
exposedField	SFNode	**textureCoords**	NULL
exposedField	SFBool	**useOrthoTexture**	FALSE
ExposedField	MFNode	**faceDefTables**	[]
ExposedField	MFNode	**faceSceneGraph**	[]

ggg

}

NOTE — For the binary encoding of this node see Document MPEG-4 NODES A.1.41.

Functionality and semantics

The **FDP** node defines the face model to be used at the terminal. Two options are supported:

1. If **faceDefTables** is NULL, calibration information is downloaded, so that the proprietary face of the terminal can be calibrated using facial feature points and, optionally, the texture information. In this case, the **featurePointsCoord** field shall be set. **featurePointsCoord** contains the coordinates of facial feature points, as defined in ISO/IEC 14496-2, Annex C, Figure C-1, corresponding to a neutral face. If a coordinate of a feature point is set to +I, the coordinates of this feature point shall be ignored. The **textureCoord** field, if set, is used to map a texture on the model calibrated by the feature points. The **textureCoord** points correspond to the feature points. That is, each defined feature point shall have corresponding texture coordinates. In this case, the **faceSceneGraph** shall contain exactly one texture image, and any geometry it might contain shall be ignored. The terminal shall interpret the feature points, texture coordinates, and the **faceSceneGraph** in the following way:

   Feature points of the terminal's face model shall be moved to the coordinates of the feature points supplied in **featurePointsCoord**, unless a feature point is to be ignored, as explained above.
   If **textureCoord** is set, the texture supplied in the **faceSceneGraph** shall be mapped onto the terminal's default face model. The texture coordinates are derived from the texture coordinates of the feature points supplied in **textureCoords**. The **useOrthoTexture** field provides a hint to the decoding terminal that, when FALSE, indicates that the texture image is best obtained by cylindrical projection of the face. If **useOrthoTexture** is TRUE, the texture image is best obtained by orthographic projection of the face.

2. A face model as described in the **faceSceneGraph** is decoded. This face model replaces the terminal's default face model in the terminal. The **faceSceneGraph** shall contain the face in its neutral position (all FAPs = 0). If desired, the **faceSceneGraph** shall contain the texture maps of the face. The functions defining the way in which the **faceSceneGraph** shall be modified, as a function of the FAPs, shall also be decoded. This information is described by **faceDefTables** that define how the **faceSceneGraph** is to be modified as a function of each FAP. By means of **faceDefTables**, **IndexedFaceSets** and **Transform** nodes of the **faceSceneGraph** can be animated. Since the amplitude of FAPs is defined in units that are dependent on the size of the face model, the **featurePointsCoord** field defines the position of facial features on the surface of the face described by **faceSceneGraph**. From the location of these feature points, the terminal computes the units of the FAPs. Generally, only two node types in the scene graph of a decoded face model are affected by FAPs: **IndexedFaceSet** and **Transform** nodes. If a FAP causes a deformation of an object (e.g. lip stretching), then the coordinate positions in the affected **IndexedFaceSets** shall be updated. If a FAP causes a movement which can be described with a **Transform** node (e.g. FAP 23, yaw_l_eyeball), then the appropriate fields in this **Transform** node shall be updated. It shall be assumed that this **Transform** node has its **rotation**, **scale**, and **translation** fields set to neutral values if the face is in its neutral position. A unique nodeId shall be assigned via the DEF statement to all **IndexedFaceSet** and **Transform** nodes which are affected by FAPs so that they can be accessed unambiguously during animation.

   The **featurePointsCoord** field shall contain a **Coordinate** node that specifies feature points for the calibration of the terminal's default face. The coordinates are specified in the **point** field of the **Coordinate** node in the prescribed order, that a feature point with a lower label number is listed before a feature point with a higher label number.

EXAMPLE — Feature point 3.14 before feature point 4.1

The **textureCoords** field shall contain a **Coordinate** node that specifies texture coordinates for the feature points. The coordinates are listed in the **point** field in the **Coordinate** node in the prescribed order, that a feature point with a lower label is listed before a feature point with a higher label.

The **useOrthoTexture** field may contain a hint to the terminal as to the type of texture image, in order to allow better interpolation of texture coordinates for the vertices that are not feature points. If **useOrthoTexture** is FALSE, the terminal may assume that the texture image was obtained by cylindrical projection of the face. If **useOrthoTexture** is 1, the terminal may assume that the texture image was obtained by orthographic projection of the face.

The **faceDefTables** field shall contain **FaceDefTables** nodes. The behavior of FAPs is defined in this field for the face in **faceSceneGraph**.

The **faceSceneGraph** field shall contain a **Group** node. In the case of option 1 (above), this may be used to contain a texture image as described above. In the case of option 2, this shall be the grouping node for the face model rendered in the compositor and shall contain the face model. In this case, the effect of facial animation parameters is defined in the **faceDefTables** field.

FIT {

exposedField	MFInt32	**FAPs**	[]
exposedField	MFInt32	**graph**	[]
exposedField	MFInt32	**numeratorTerms**	[]
exposedField	MFInt32	**denominatorTerms**	[]
exposedField	MFInt32	**numeratorExp**	[]
exposedField	MFInt32	**denominatorExp**	[]
exposedField	MFInt32	**numeratorImpulse**	[]
exposedField	MFFloat	**numeratorCoefs**	[]
exposedField	MFFloat	**denominatorCoefs**	[]

}

NOTE — For the binary encoding of this node see Document MPEG-4 NODES A.1.42.

Functionality and semantics

The **FIT** node allows a smaller set of FAPs to be sent during a facial animation. This small set can then be used to determine the values of other FAPs, using a rational polynomial mapping between parameters. In a **FIT** node, rational polynomials are used to specify interpolation functions.

EXAMPLE — The top inner lip FAPs can be sent and then used to determine the top outer lip FAPs. Another example is that only viseme and/or expression FAPs are sent to drive the face. In this case, low-level FAPs are interpolated from these two high-level FAPs.

To make the scheme general, sets of FAPs are specified, along with a FAP interpolation graph (FIG) between the sets that specifies which sets are used to determine which other sets. The FIG is a graph with directed links. Each node contains a set of FAPs. Each link from a parent node to a child node indicates that the FAPs in the

child node can be interpolated from the parent node. **Expression** (FAP#1) or **Viseme** (FAP #2) and their fields shall not be interpolated from other FAPs.

In a FIG, a FAP may appear in several nodes, and a node may have multiple parents. For a node that has multiple parent nodes, the parent nodes are ordered as 1st parent node, 2nd parent node, etc. During the interpolation process, if this child node needs to be interpolated, it is first interpolated from 1st parent node if all FAPs in that parent node are available. Otherwise, it is interpolated from 2nd parent node, and so on.

An example of FIG is shown in

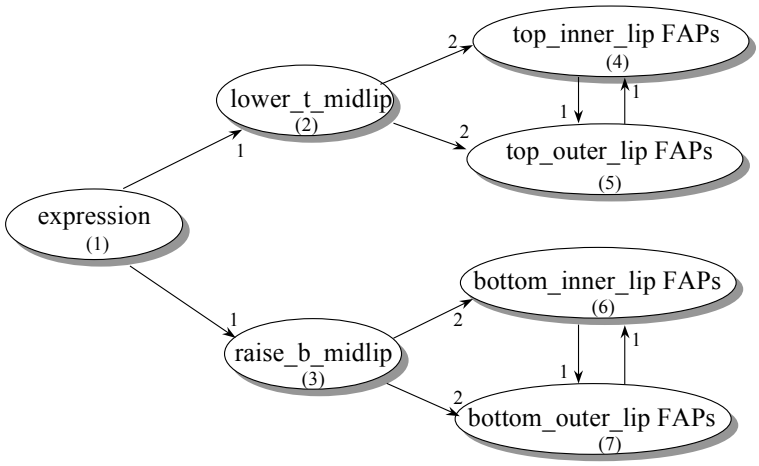


Figure 0-10. Each node has a nodeID. The numerical label on each incoming link indicates the order of these links.

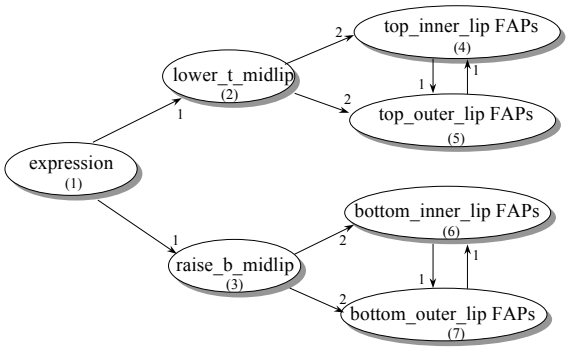


Figure 0-10 - A FIG example

The interpolation process based on the FAP interpolation graph is described using pseudo-C code as follows:

```
do {
   interpolation_count = 0;
   for (all Node_i) {   // from Node_1 to Node_N
      for (ordered Node_i's parent Node_k) {
         if (FAPs in Node_i need interpolation and
                FAPs in Node_k have been interpolated or are
available) {
```

jjj

```
            interpolate Node_i from Node_k;  //using interpolation
function
                                              // table here
            interpolation_count ++;
            break;
          }
       }
    }
} while (interpolation_count != 0);
```

Each directed link in a FIG is a set of interpolation functions. Suppose F_1, F_2, …, F_n are the FAPs in a parent set and f_1, f_2, …, f_m are the FAPs in a child set.

Then, there are m interpolation functions denoted as:

$f_1 = I_1(F_1, F_2, \ldots, F_n)$

$f_2 = I_2(F_1, F_2, \ldots, F_n)$

…

$f_m = I_m(F_1, F_2, \ldots, F_n)$

Each interpolation function I_k () is in a rational polynomial form if the parent node does not contain viseme FAP or expression FAP.

$$I(F_1, F_2, ..., F_n) = \sum_{i=0}^{K-1} (c_i \prod_{j=1}^{n} F_j^{l_{ij}}) \Big/ \sum_{i=0}^{P-1} (b_i \prod_{j=1}^{n} F_j^{m_{ij}})$$

Otherwise, an impulse function is added to each numerator polynomial term to allow selection of expression or viseme.

$$I(F_1, F_2, ..., F_n) = \sum_{i=0}^{K-1} \delta(F_{s_i} - a_i)(c_i \prod_{j=1}^{n} F_j^{l_{ij}}) \Big/ \sum_{i=0}^{P-1} (b_i \prod_{j=1}^{n} F_j^{m_{ij}})$$

In both equations, K and P are the numbers of polynomial products, c_i and b_i are the coefficient of the ith product. l_{ij} and m_{ij} are the power of F_j in the ith product. An impulse function equals 1 when $F_{s_i} = a_i$, otherwise, equals 0. F_{s_i} can only be viseme_select1, viseme_select2, expression_select1, and expression_select2. a_i is an integer that ranges from 0 to 6 when F_{s_i} is expression_select1 or expression_select2, ranges 0 to 14 when F_{s_i} is viseme_select1 or viseme_select2. The encoder shall send an interpolation function table which contains K, P, a_i, s_i, c_i, b_i, l_{ij}, m_{ij} to the terminal.

To aid in the explanation below, it is assumed that there are N different sets of FAPs with index 1 to N, and that each set has n_i, i=1,..,N parameters. It is also assumed that there are L directed links in the FIG and that each link points from the FAP set with index P_i to the FAP set with index C_i, for i = 1, .. , L

The FAPs field shall contain a list of FAP-indices specifying which animation parameters form sets of FAPs. Each set of FAP indices is terminated by –1. There shall be a total of $N + n_1 + n_2 + \ldots + n_N$ numbers in this field, with N of them being –1. FAP#1 to FAP#68 are of indices 1 to 68. Fields of the **Viseme** FAP (FAP#1), namely, **viseme_select1**, **viseme_select2**, **viseme_blend**, are of indices from 69 to 71. Fields of the **Expression** FAP (FAP#2), namely, **expression_select1**, **expression_select2**, **expression_intensity1**, **expression_intensity2** are of indices from 72 to 75. When the parent node contains a **Viseme** FAP, three indices, 69, 70, 71, shall be included in the

node (but not index 1). When a parent node contains an **Expression** FAP, four indices, 72,73,74,75, shall be included in the node (but not index 2).

The **graph** field shall contain a list of pairs of integers, specifying a directed links between sets of FAPs. The integers refer to the indices of the sets specified in the **FAPs** field, and thus range from 1 to N. When more than one direct link terminates at the same set, that is, when the second value in the pair is repeated, the links have precedence determined by their order in this field. This field shall have a total of 2L numbers, corresponding to the directed links between the parents and children in the FIG.

The **numeratorTerms** field shall be a list containing the number of terms in the polynomials of the numerators of the rational functions used to interpolate parameter values. Each element in the list corresponds to *K* in equation 1 above). Each link i (that is, the *i*th integer pair) in the **graph** field must have n_{Ci} values specified, one for each child FAP. The order in the **numeratorTerms** list shall correspond to the order of the links in the **graph** field and the order that the child FAP appears in the **FAPs** field. There shall be $n_{C1} + n_{C2} + \ldots + n_{CL}$ numbers in this field.

The **denominatorTerms** field shall contain a list of the number of terms in the polynomials of the denominator of the rational functions controlling the parameter value. Each element in the list corresponds to *P* in equation 1. Each link i (that is, the *i*th integer pair) in the **graph** field must have n_{Ci} values specified, one for each child FAP. The order in the **denominatorTerms** list corresponds to the order of the links in the **graph** field and the order that the child FAP appears in the **FAPs** field. There shall be $n_{C1} + n_{C2} + \ldots + n_{CL}$ numbers in this field.

The **numeratorImpulse** field shall contain a list of impulse functions in the numerator of the rational function for links with the **Viseme** or **Expression** FAP in parent node. This list corresponds to the $\delta(F_{s_i} - a_i)$. Each entry in the list is (s_i, a_i).

The **numeratorExp** field shall contain a list of exponents of the polynomial terms in the numerator of the rational function controlling the parameter value. This list corresponds to l_{ij}. For each child FAP in each link i, n_{Pi}*K values need to be specified. The order in the **numeratorExp** list shall correspond to the order of the links in the **graph** field and the order that the child FAP appears in the **FAPs** field.

NOTE — K may be different for each child FAP.

The **denominatorExp** field shall contain a list of exponents of the polynomial terms of the denominator of the rational function controlling the parameter value. This list corresponds to m_{ij}. For each child FAP in each link i, n_{Pi}*P values need to be specified. The order in the **denominatorExp** list shall correspond to the order of the links in the **graph** field and the order that the child FAP appears in the **FAPs** field.

NOTE — P may be different for each child FAP.

The **numeratorCoefs** field shall contain a list of coefficients of the polynomial terms of the numerator of the rational function controlling the parameter value. This list corresponds to c_i. The list shall have *K* terms for each child parameter that appears in a

link in the FIG, with the order in **numeratorCoefs** corresponding to the order in **graph** and **FAPs**.

NOTE — K is dependent on the polynomial, and is not a fixed constant.

The **denominatorCoefs** field shall contain a list of coefficients of the polynomial terms in the numerator of the rational function controlling the parameter value. This list corresponds to b_i. The list shall have *P* terms for each child parameter that appears in a link in the FIG, with the order in **denominatorCoefs** corresponding to the order in **graph** and **FAPs**.

NOTE — P is dependent on the polynomial, and is not a fixed constant.

EXAMPLE — Suppose a FIG contains four nodes and 2 links. Node 1 contains FAP#3, FAP#3, FAP#5. Node 2 contains FAP#6, FAP#7. Node 3 contains an expression FAP, which means contains FAP#72, FAP#73, FAP#74, and FAP#75. Node 4 contains FAP#12 and FAP#17. Two links are from node 1 to node 2, and from node 3 to node 4. For the first link, the interpolation functions are

$F_6 = (F_3 + 2F_4 + 3F_5 + 4F_3F_4^2)/(5F_5 + 6F_3F_4F_5)$

$F_7 = F_4$.

For the second link, the interpolation functions are

$$F_{12} = \delta(F_{72} - 6)(0.6F_{74}) + \delta(F_{73} - 6)(0.6F_{75})$$

$$F_{17} = \delta(F_{72} - 6)(-1.5F_{74}) + \delta(F_{73} - 6)(-1.5F_{75})$$.

The second link simply says that when the expression is surprise (FAP#72=6 or FAP#73=6), for FAP#12, the value is 0.6 times of expression intensity FAP#74 or FAP#75; for FAP#17, the value is –1.5 tims of FAP#74 or FAP#75.

After the FIT node given below, we explain each field separately.

```
FIT {
   FAPs            [ 3 4 5 -1 6 7 -1 72 73 74 75 -1 12 17 -1]
   graph           [ 1 2 3 4]
   numeratorTerms  [ 4 1 2 2 ]
   denominatorTerms   [2 1 1 1]
   numeratorExp    [1 0 0   0 1 0   0 0 1  1 2 0  0 1 0
                        0 0 1 0  0 0 0 1  0 0 1 0  0 0 0 1 ]
   denominatorExp  [ 0 0 1  1 1 1  0 0 0
                        0 0 0 0  0 0 0 0 ]
   numeratorImpulse   [ 72 6  73 6  72 6  73 6 ]
   numeratorCoefs  [1 2 3 4   1   0.6 0.6  -1.5 -1.5 ]
   denominatorCoefs   [5 6 1 1 1 ]
}
```

FAPs [3 4 5 -1 6 7 –1 72 73 74 75 –1 12 17 -1]

Four sets of FAPs are defined, the first with FAPs number 3, 4, and 5, the second with FAPs number 6 and 7, the third with FAPs number 72, 73, 74, 75, and the fourth with FAPs number 12, 17.

graph [1 2 3 4]

The first set is made to be the parent of the second set, so that FAPs number 6 and 7 will be determined by FAPs 3, 4, and 5. Also, the third set is made to be the parent of the fourth set, so that FAPs number 12 and 17 will be determined by FAPs 72, 73, 74, and 75.

numeratorTerms [4 1 2 2]

The rational functions that define F6 and F7 are selected to have 4 and 1 terms in their numerator, respectively. Also, the rational functions that define F12 and F17 are selected to have 2 and 2 terms in their numerator, respectively.

denominatorTerms [2 1 1 1]

The rational functions that define F6 and F7 are selected to have 2 and 1 terms in their denominator, respectively. Also, the rational functions that define F12 and F17 are selected to both have 1 term in their denominator.

numeratorExp [1 0 0 0 1 0 0 0 1 1 2 0 0 1 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 1]

The numerator selected for the rational function defining F6 is F3 + 2F4 + 3 F5 + 4F3F42. There are 3 parent FAPs, and 4 terms, leading to 12 exponents for this rational function. For F7, the numerator is just F4, so there are three exponents only (one for each FAP). Values for F12 and F17 are derived in the same way.

denominatorExp [0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0]

The denominator selected for the rational function defining F6 is 5F5+ 6F3F4F5 , so there are 3 parent FAPs and 2 terms and hence, 6 exponents for this rational function. For F7, the denominator is just 1, so there are three exponents only (one for each FAP). Values for F12 and F17 are derived in the same way.

numeratorImpulse [72 6 73 6 72 6 73 6]

For the second link, all four numerator polynomial terms contain impulse function $\delta(F_{72}-6)$ or $\delta(F_{73}-6)$.

numeratorCoefs [1 2 3 4 1 0.6 0.6 -1.5 –1.5]

There is one coefficient for each term in the numerator of each rational function.

denominatorCoefs [5 6 1 1 1]

There is one coefficient for each term in the denominator of each rational function.

nnn

Appendix VI-K

Table K-2

FAP DEFINITIONS, GROUP ASSIGNMENTS AND STEP SIZES

#	FAP name	FAP description	units	Uni-or Bidir	Pos motion	Grp	FDP subgrp num	Quant step size	Min/Max I-Frame quantized values	Min/Max P-Frame quantized values
1	Viseme	Set of values determining the mixture of two visemes for this frame (e.g. pbm, fv, th)	na	na	na	1	na	1	viseme_blend: +63	viseme_blend: +-63
2	expression	A set of values determining the mixture of two facial expression	Na	na	na	1	na	1	expression_intensity1, expression_intensity2: +63	expression_intensity1, expression_intensity2: +-63
3	open_jaw	Vertical jaw displacement (does not affect mouth opening)	MNS	U	down	2	1	4	+1080	+360
4	lower_t_midlip	Vertical top middle inner lip displacement	MNS	B	down	2	2	2	+-600	+-180
5	raise_b_midlip	Vertical bottom middle inner lip displacement	MNS	B	up	2	3	2	+-1860	+-600
6	stretch_l_cornerlip	Horizontal displacement of left inner lip corner	MW	B	left	2	4	2	+-600	+-180
7	stretch_r_cornerlip	Horizontal displacement of right inner lip corner	MW	B	right	2	5	2	+-600	+-180
8	lower_t_lip_lm	Vertical displacement of midpoint between left corner and middle of top inner lip	MNS	B	down	2	6	2	+-600	+-180
9	lower_t_lip_rm	Vertical displacement of midpoint between right corner and middle of top inner lip	MNS	B	down	2	7	2	+-600	+-180

ppp

#	FAP name	FAP description	units	Uni-or Bidir	Pos motion	Grp	FDP subgrp num	Quant step size	Min/Max I-Frame quantized values	Min/Max P-Frame quantized values
10	raise_b_lip_lm	Vertical displacement of midpoint between left corner and middle of bottom inner lip	MNS	B	up	2	8	2	+-1860	+-600
11	raise_b_lip_rm	Vertical displacement of midpoint between right corner and middle of bottom inner lip	MNS	B	up	2	9	2	+-1860	+-600
12	raise_l_cornerlip	Vertical displacement of left inner lip corner	MNS	B	up	2	4	2	+-600	+-180
13	raise_r_cornerlip	Vertical displacement of right inner lip corner	MNS	B	up	2	5	2	+-600	+-180
14	thrust_jaw	Depth displacement of jaw	MNS	U	forward	2	1	1	+600	+180
15	shift_jaw	Side to side displacement of jaw	MW	B	right	2	1	1	+-1080	+-360
16	push_b_lip	Depth displacement of bottom middle lip	MNS	B	forward	2	3	1	+-1080	+-360
17	push_t_lip	Depth displacement of top middle lip	MNS	B	forward	2	2	1	+-1080	+-360
18	depress_chin	Upward and compressing movement of the chin (like in sadness)	MNS	B	up	2	10	1	+-420	+-180
19	close_t_l_eyelid	Vertical displacement of top left eyelid	IRISD	B	down	3	1	1	+-1080	+-600
20	close_t_r_eyelid	Vertical displacement of top right eyelid	IRISD	B	down	3	2	1	+-1080	+-600

qqq

#	FAP name	FAP description	units	Uni-or Bidir	Pos motion	Grp	FDP subgrp num	Quant step size	Min/Max I-Frame quantized values	Min/Max P-Frame quantized values
21	close_b_l_eyelid	Vertical displacement of bottom left eyelid	IRISD	B	up	3	3	1	+-600	+-240
22	close_b_r_eyelid	Vertical displacement of bottom right eyelid	IRISD	B	up	3	4	1	+-600	+-240
23	yaw_l_eyeball	Horizontal orientation of left eyeball	AU	B	left	3	na	128	+-1200	+-420
24	yaw_r_eyeball	Horizontal orientation of right eyeball	AU	B	left	3	na	128	+-1200	+-420
25	pitch_l_eyeball	Vertical orientation of left eyeball	AU	B	down	3	na	128	+-900	+-300
26	pitch_r_eyeball	Vertical orientation of right eyeball	AU	B	down	3	na	128	+-900	+-300
27	thrust_l_eyeball	Depth displacement of left eyeball	ES	B	forward	3	na	1	+-600	+-180
28	thrust_r_eyeball	Depth displacement of right eyeball	ES	B	forward	3	na	1	+-600	+-180
29	dilate_l_pupil	Dilation of left pupil	IRISD	B	growing	3	5	1	+-420	+-120
30	dilate_r_pupil	Dilation of right pupil	IRISD	B	growing	3	6	1	+-420	+-120
31	raise_l_i_eyebrow	Vertical displacement of left inner eyebrow	ENS	B	up	4	1	2	+-900	+-360
32	raise_r_i_eyebrow	Vertical displacement of right inner eyebrow	ENS	B	up	4	2	2	+-900	+-360
33	raise_l_m_eyebrow	Vertical displacement of left middle eyebrow	ENS	B	up	4	3	2	+-900	+-360
34	raise_r_m_eyebrow	Vertical displacement of right middle eyebrow	ENS	B	up	4	4	2	+-900	+-360

rrr

#	FAP name	FAP description	units	Uni-or Bidir	Pos motion	Grp	FDP subgrp num	Quant step size	Min/Max I-Frame quantized values	Min/Max P-Frame quantized values
35	raise_l_o_eyebrow	Vertical displacement of left outer eyebrow	ENS	B	up	4	5	2	+-900	+-360
36	raise_r_o_eyebrow	Vertical displacement of right outer eyebrow	ENS	B	up	4	6	2	+-900	+-360
37	squeeze_l_eyebrow	Horizontal displacement of left eyebrow	ES	B	right	4	1	1	+-900	+-300
38	squeeze_r_eyebrow	Horizontal displacement of right eyebrow	ES	B	left	4	2	1	+-900	+-300
39	puff_l_cheek	Horizontal displacement of left cheeck	ES	B	left	5	1	2	+-900	+-300
40	puff_r_cheek	Horizontal displacement of right cheeck	ES	B	right	5	2	2	+-900	+-300
41	lift_l_cheek	Vertical displacement of left cheek	ENS	U	up	5	3	2	+-600	+-180
42	lift_r_cheek	Vertical displacement of right cheek	ENS	U	up	5	4	2	+-600	+-180
43	shift_tongue_tip	Horizontal displacement of tongue tip	MW	B	right	6	1	1	+-1080	+-420
44	raise_tongue_tip	Vertical displacement of tongue tip	MNS	B	up	6	1	1	+-1080	+-420
45	thrust_tongue_tip	Depth displacement of tongue tip	MW	B	forward	6	1	1	+-1080	+-420
46	raise_tongue	Vertical displacement of tongue	MNS	B	up	6	2	1	+-1080	+-420
47	tongue_roll	Rolling of the tongue into U shape	AU	U	concave upward	6	3, 4	512	+300	+60
48	head_pitch	Head pitch angle from top of spine	AU	B	down	7	na	170	+-1860	+-600

#	FAP name	FAP description	units	Uni-or Bidir	Pos motion	Grp	FDP subgrp num	Quant step size	Min/Max I-Frame quantized values	Min/Max P-Frame quantized values
49	head_yaw	Head yaw angle from top of spine	AU	B	left	7	na	170	+-1860	+-600
50	head_roll	Head roll angle from top of spine	AU	B	right	7	na	170	+-1860	+-600
51	lower_t_midlip _o	Vertical top middle outer lip displacement	MNS	B	down	8	1	2	+-600	+-180
52	raise_b_midlip_o	Vertical bottom middle outer lip displacement	MNS	B	up	8	2	2	+-1860	+-600
53	stretch_l_cornerlip_o	Horizontal displacement of left outer lip corner	MW	B	left	8	3	2	+-600	+-180
54	stretch_r_cornerlip_o	Horizontal displacement of right outer lip corner	MW	B	right	8	4	2	+-600	+-180
55	lower_t_lip_lm _o	Vertical displacement of midpoint between left corner and middle of top outer lip	MNS	B	down	8	5	2	+-600	+-180
56	lower_t_lip_rm _o	Vertical displacement of midpoint between right corner and middle of top outer lip	MNS	B	down	8	6	2	+-600	+-180
57	raise_b_lip_lm_o	Vertical displacement of midpoint between left corner and middle of bottom outer lip	MNS	B	up	8	7	2	+-1860	+-600
58	raise_b_lip_rm_o	Vertical displacement of midpoint between right corner and middle of bottom outer lip	MNS	B	up	8	8	2	+-1860	+-600

ttt

#	FAP name	FAP description	units	Uni-or Bidir	Pos motion	Grp	FDP subgrp num	Quant step size	Min/Max I-Frame quantized values	Min/Max P-Frame quantized values
59	raise_l_cornerlip_o	Vertical displacement of left outer lip corner	MNS	B	up	8	3	2	+-600	+-180
60	raise_r_cornerlip_o	Vertical displacement of right outer lip corner	MNS	B	up	8	4	2	+-600	+-180
61	stretch_l_nose	Horizontal displacement of left side of nose	ENS	B	left	9	1	1	+-540	+-120
62	stretch_r_nose	Horizontal displacement of right side of nose	ENS	B	right	9	2	1	+-540	+-120
63	raise_nose	Vertical displacement of nose tip	ENS	B	up	9	3	1	+-680	+-180
64	bend_nose	Horizontal displacement of nose tip	ENS	B	right	9	3	1	+-900	+-180
65	raise_l_ear	Vertical displacement of left ear	ENS	B	up	10	1	1	+-900	+-240
66	raise_r_ear	Vertical displacement of right ear	ENS	B	up	10	2	1	+-900	+-240
67	pull_l_ear	Horizontal displacement of left ear	ENS	B	left	10	3	1	+-900	+-300
68	pull_r_ear	Horizontal displacement of right ear	ENS	B	right	10	4	1	+-900	+-300

Table K-3

FAP GROUPING

Group	Number of FAPs
1: visemes and expressions	2
2: jaw, chin, inner lowerlip, cornerlips, midlip	16
3: eyeballs, pupils, eyelids	12
4: eyebrow	8
5: cheeks	4
6: tongue	5
7: head rotation	3
8: outer lip positions	10
9: nose	4
10: ears	4

Table K-4

VALUES FOR EXPRESSION_SELECT

Expression _select	Expression name	Textual description
0	na	Na
1	joy	The eyebrows are relaxed. The mouth is open and the mouth corners pulled back toward the ears.
2	sadness	The inner eyebrows are bent upward. The eyes are slightly closed. The mouth is relaxed.
3	anger	The inner eyebrows are pulled downward and together. The eyes are wide open. The lips are pressed against each other or opened to expose the teeth.
4	fear	The eyebrows are raised and pulled together. The inner eyebrows are bent upward. The eyes are tense and alert.
5	disgust	The eyebrows and eyelids are relaxed. The upper lip is raised and curled, often asymmetrically.
6	surprise	The eyebrows are raised. The upper eyelids are wide open, the lower relaxed. The jaw is opened.

Table K-5

VALUES FOR VISEME_SELECT

Viseme_select	Phonemes	Example
0	none	na
1	p, b, m	put, bed, mill
2	f, v	far, voice
3	T,D	think, that
4	t, d	tip, doll
5	k, g	call, gas
6	tS, dZ, S	chair, join, she
7	s, z	sir, zeal
8	n, l	lot, not
9	r	red
10	A:	car
11	e	bed
12	I	tip
13	Q	top
14	U	book

Bibliographical References

A--G

Ahlberg, J. (2002). An active model for facial feature tracking. *Eurasip Journal on Applied Signal Processing*, Vol. 6, pp. 566-571.

Al-Qayedi, A. M.,& Clark, A. (2000). Constant-rate eye tracking and animation for model-based-coded video. *Proceedings of the IEEE International Conference on Acoustic Speech and Signal Processing.*

Andrés del Valle, A. C& Dugelay, J.-L.(2002) Online face analysis: coupling head pose-tracking with face expression analysis [Technical demo] ACM Multimedia .

Andrés del Valle, A. C& Dugelay, J.-L.(2003) Making machines understand facial motion and expression like humans do. *10th International Conference in Human-Computer Interaction,*, "HCI – Theory and Practice (Part II)", Vol. 2, pp. 581-585.

Avaro, Basso, Casner, Civanlar, Gentric, Herpel, et al. (2001) *RTP payload format for MPEG-4 streams [work in progress].* IETF: draft-gentric-avt—mpeg4-multiSL-03.txt

Bartlett, M. S. (2001). *Face image analysis by unsupervised learning*. Boston: Kluwer Academic Publishers.

Bartlett, M. S., Braathen, B. LittleWort-Ford, G., Hershey, J., Fasel, I., Marks, T., Smith, E., Sejnowski, T. J., & Movellan, J. R. (2001). *Automatic analysis of spontaneous facial behavior: A final project report*. (Tech. Rep. No. 2001.08). San Diego, CA: University of California, San Diego, MPLab.

Bascle B., & Blake A. (1998). Separability of pose and expression in facial tracking and animation. *Proceedings of the 6th International Conference on Computer Vision*, 323-328.

Benedes, O. (1999). Watermarking of 3-D polygon based models with robustness against mesh simplification. *Proceedings of SPIE: Security and Watermarking of Multimedia Contents,* pp. 329—340.

Black, M. J., & Yacoob, Y. (1997). Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, 25(1), 23-48.

Breton, G. (2002). Animation des visages 3D parlants pour nouveaux IHM et services de télécommunication [Animation of 3D talking heads for new HCI and telecom services]. (Doctoral dissertation. Université de Rennes 1, 2002).

Brulé, J. F. (1985). Fuzzy systems – A tutorial. Retrieved October, 1, 2002, from http://www.austinlinks.com/Fuzzy/tutorial

Chen, K., & Kambhamettu, C. (1997). Real-time facial animation through the Internet. (Tech. Rep. No. 98-14). DE: University of Delaware, Department of Computer and Information Sciences.

Chen, L. S., & Huang, T. S. (2000). Emotional expressions in audiovisual human computer interaction. *Proceedings of IEEE the International Conference on Multimedia and Expo.*

Chen, T. (January, 2001). Audiovisual speech processing. Lip reading and lip synchronization. *IEEE Signal Processing Magazine,*pp. 9-21.

Chou, J.-C., Chang, Y.-J., & Chen, Y.-C. (2001). Facial feature point tracking ad expression analysis for virtual teleconferencing systems. *Proceedings of the International Conference on Multimedia and Expo*, pp. 25-28.

Cordea, M. D., Petriu, E. M., Georganas, N. D., Petriu, D. C., & Whalen, T. E. (2001). 3D head pose recovery for interactive virtual reality avatars. *Proceedings of the IEEE Instrumentation and Measurement Technology Conference*

Curinga, S. (1998). Use of a statistical model for lip synthesis. *IEEE.*

Cyberware Home Page (2003). Retrieved July, 3rd 2003 from:
http://www.cyberware.com/

Debevec, P., Hawkins, T., Tchou, C., Duiker, H.-P., Sarokin, W., & Sagar, M. (2000). Acquiring the reflectance field of a human face. *Proceedings of SIGGRAPH 2000,* 145-156. ACM Press/ACM SIGGRAPH/Addison Wesley Longman.

DeCarlo, D., & Metaxas, D. (1996). The integration of optical flow and deformable models with applications to human face shape and motion estimation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 231-238.

DirectX® Home Page (2003). Microsoft®'s website about DirectX. Retrieved June, 16th 2003 from:
http://www.microsoft.com/windows/directx/default.aspx

Dubuc, C., & Budreau, D. (2001, December). The design and simulated performance of a mobile video telephony application for satellite third-generation wireless system. *IEEE Transactions on Multimedia*, 3(4).

Dugelay, J.-L., & Andrés del Valle, A. C. (2001) Analysis-synthesis cooperation for MPEG-4 realistic clone animation. *Proceedings of the ICAV3D*.

D--I

Dugelay, J.-L., Fintzel, K., & Valente, S. (1999). Synthetic Natural Hybrid video processing for virtual teleconferencing systems. *Picture Coding Symposium*

Dugelay, J.-L., Garcia, E., & Mallauran, C. (2002). *Protection of 3D object usage through texture watermarking*. Proceedings of the XI European Signal Processing Conference

Eisert, P., & Girod, B. (1998). Analyzing facial expression for virtual conferencing. *Proceedings of the IEEE Computer Graphics & Applications,* 70-78.

Eisert, P., & Girod, B. (2002). Model-based enhancement of lighting conditions in image sequences. *Proceedings of the Visual Communications and Image Processing*

Ekman, P., & Friesen, W. V. (1978). *The facial action coding system*. Palo Alto, Ca.: Investigator's Guide, Consulting Psychologists Press.

Essa, I., Basu, S., Darrel, T., & Pentland, A. (1996). Modeling, tracking and interactive animation of faces and heads using input from video. *Proceedings of Computer Animation.*

Eveno, N., Caplier, A. & Coulon, P. Y. (2002) Key points based segmentation of lips. *IEEE International Conference on Multimedia and Expo.*

Eveno, N., Caplier, A., & Coulon, P. Y. (2001). A new color transformation for lips segmentation. *Workshop on Multimedia Signal Processing.*

Eye movements in HCI. (2003) Regular session at the 5th Int. Conf. on Engineering Psychology and Cognitive Ergonomics. *10th International Conference in Human-Computer Interaction,* Vol. 3.

Ezzat, T., & Poggio, T. (1996a). Facial analysis and synthesis using image based models. *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition.*

Ezzat, T., & Poggio, T. (1996b). Facial analysis and synthesis using image based models. *Proceedings of the Workshop on the Algorithm Foundations of Robotics.*

Fellenz, W. A., Taylor, J. G., Cowie, R., Douglas-Cowie, E., Piat, F., Kollias, S., Orovas, C., & Apolloni, B. (2000). On emotion recognition of faces and of speech using neural networks, fuzzy logic and the ASSESS system. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks.*

Fidaleo, D., & Neumann, U. (2002). Co-Art: Co-articulation region analysis for control of 2D characters. *Proceedings of IEEE Computer Animation*, 12-17.

Garau, M., Slater, M., Bee, S., & Sasse, M. A. (2001). The impact of eye gaze on communication using humanoid avatars. *Proceedings of the SIG-CHI Conference on Human Factors in Computing Systems*, 309-316.

Garcia, C., & Tziritas, G. (1999, September). Face detection using quantized skin color regions merging and wavelet packet analysis. *IEEE Transactions on Multimedia*. 1(3), 264-277.

Gemmell, J., Zitnick, L., Kang, T., & Toyama, K. (2000). Software-enabled gaze-aware videoconferencing. *IEEE Multimedia*, 7(4), 26-35.

Goto, T., Escher, M., Zanardi, C., & Magnenat-Thalmann, N. (1999). MPEG-4 based animation with face feature tracking. *In Computer Animation and Simulation.*

Goto, T., Kshirsagar, S., & Magnenat-Thalmann, N. (2001, May). Automatic face cloning and animation. *IEEE Signal Processing Magazine*. 17-25.

Haverlant, V., & Dax, P. (1999). *Portage de VRENG sur RTP* [Porting VRENG over RTP]. (Diploma thesis, Telecom Paris, September 1999).

Holbert, S., & Dugelay, J.-L. (1995). Active contours for lip-reading: combining snakes with templates. *Quinzième colloque GRETSI*, 717-720.

Huang, C.-L., Huang, Y.-M. (1997, September). Facial expression recognition using model-based feature extraction and action parameters classification. *Journal of Visual Communication and Image Representation*, 8(3), 278-290.

Huang, F. J., & Chen, T. (2000). Tracking of multiple faces for human-computer interfaces and virtual environments. *Proceeding of the IEEE International Conference and Multimedia Expo.*

Huang, Y. S., Tsai, Y. H., & Shieh, J. W. (2001). Robust face recognition with light compensation. *Proceedings of the Second IEEE Pacific-Rim Conference on Multimedia*

Huntsberger, T. L., Rose, J., & Ramaka, A. (1998). Fuzzy-Face: A hybrid wavelet/fuzzy self-organizing feature map system for face processing. *Journal of Biological Systems*

Images of muscle and bones of the head. (2002). Retrieved: December, 2002, from http://mywebpages.comcast.net/wnor/homepage.htm
http://www.umanitoba.ca/faculties/dentistry/oral_biology/tutorials/musfacex.pdf

ICA (2003) Independent Component Analysis. Retrieved: October, 2003, from http://www.cis.hut.fi/projects/ica/

INTERFACE IST-1999-10036. (1999) IST-European Project. Retrieved May, 13, 2003, from http://www.ist-interface.org/

I--N

ISO/IEC 14496-1 MPEG-4. (1998, November). Part 1: Systems. Atlantic City

ISO/IEC 14496-2 MPEG-4. (1999, December). Part 2: Visual. Maui

IST-European Project: INTERFACE IST-1999-10036. Retrieved from: http://www.cordis.lu/ist/projects/99-10036.htm

Jones, M. J., & Rehg, J. M. (1999). Statistical color models with application to skin detection. *Proceedings of the Computer Vision and Pattern Recognition*. 274-280.

Kalra, P., Mangili, A., Magnenat-Thalmann, N., & D. Thalmann. (1992). Simulation of facial muscle actions based on rational free-form deformations. *Eurographics.*

Kampmann, M. (2002) Automatic 3-D face model adaptation for model-based coding of videophone sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(3): 183-192.

Kay, S. M. (1993). Chapter 13 in *Fundamentals of statistical signal processing estimation theory* (pp. 419□477). Englewood Cliffs, NJ: PTR Prentice Hall.

King, S. A., & Parent, R. E. (2001). A parametric tongue model for animated speech. *Journal of Visualization and Computer Animation,* 12(3), 107-115.

Kouvelas, I., Hardman, V., & Watson, A. (1996). Lip Synchronization for Use over the Internet: Analysis and Implementation. *Proceedings of the IEEE Globecom*

Lee, C. (Producer). (2001) *Final Fantasy – the spirits within* [Motion picture]. United States: Square Pictures, Inc. Columbia Tristar Interactive

Leroy, B., & Herlin, I. L. (1995). Un modèle deformable paramétrique pour la reconnaissance de visages et le suivi du movement des lèvres [A parametric deformable model for face recognition and lip motion tracking]. *Quinzième colloque GRETSI*, 701-704.

Li, H., Roivainen, P., & Forchheimer, R. (1993). 3-D motion estimation in model-based facial image coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6), 545-555.

Liévin, M. & Luthon, F. (2000) A hierarchical segmentation algorithm for face analysis. *Proceedings of the IEEE International Conference in Multimedia and Expo,* Vol. 2, pp. 1085-1088.

Liévin, M., Delmas, P., Coulon, P. Y., Luthon, F., & Fristot, V. (1999) Automatic lip tracking: Bayesian segmentation and active contours in a cooperative scheme. *Proceedings of the IEEE Int. Conf. on Multimedia Computing and Systems*, Vol. 1, pp. 691-696.

Luettin, J., Thacker, N. A., & Beer, S. W. (1996). Statistical lip modeling for visual speech recognition. *Proceedings of the VIII European Signal Processing Conference*.

Luong, Q.-T., & Faugeras, O. D. (1997). Self-calibration of a moving camera from point correspondences and fundamental matrices. *International Journal of Computing Vision,* 22(3), 261-289.

Luong, Q.-T., Fua, P., & Leclerc, Y. (2002, January). The radiometry of multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1), 19-33.

Malciu, M., & Prêteux, F. (2001). MPEG-4 compliant tracking of facial features in video sequences. *Proceedings of the International Conference on Augmented Virtual Environments and 3-D Imaging*, 108-111.

Metaxas, D. (1999). Deformable model and HMM-based tracking, analysis and recognition of gestures and faces. *Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems.*

Morishima, S. (2001, May). Face analysis and synthesis for duplication expression and impression. *IEEE Signal Processing Magazine*. 26-34.

Morishima, S., Ishikawa, T., & Terzopoulos, D. (1998). Physics model based very low bit rate 3D facial image coding. *Very Low Bit Video Workshop*.

Moses, Y., Reynard, D., & Blake, A. (1995). Determining facial expressions in real time. *Proceedings of the International Workshop on Automatic Face and Gesture Recognition*, 332-337.

Motion capture websites (2002):
http://www.vicon.com/
http://www.biovision.com /
http://www.motionanalysis.com/
http://www.metamotion.com/

MPEG-4. (2000, January) *Signal Processing: Image Communication*. Tutorial Issue on the MPEG-4 Standard, Vol. 15, Nos. 4-5.

Nikolaidis, A., & Pitas, I. (2000). Facial feature extraction and pose determination. *The Journal of the Pattern Recognition Society*, 33, 1783-1791.

O--S

Odisio, M., Elisei, F., Bailly, G., & Badin, P. (2001). Clones parlants 3D vidéo-réalistes: Application à l'analyse de messages audiovisuals. [Video-realistic 3D talking-clones: applied to the analysis of audiovisual messages]. *Proceedings of Compression et Réprésentation des Signaux Audiovisuels*

Ostermann, J., & Millen, D. (2000, August). Talking Head and synthetic speech: an architecture for supporting electronic commerce. *Proceedings of the IEEE International Conference on Multimedia and Expo*

Ostermann, J., Rurainsky, J., & Civanlar, R. (2001). *RTP payload format for phoneme/facial animation parameters (PFAP)* [Expired April 2002]. RTF: draft-ietf-avt-rtp-pfap-00.txt

Ostermann, J., Rurainsky, J., & Civanlar, R. (2002). Real-time streaming for the animation of talking faces in multiuser environments. *Proceedings of the IEEE International Symposium on Circuits and Systems.*

Pahor, V. & Carrato, S. (1999). A fuzzy approach to mouth corner detection. *Proceedings of the IEEE International Conference on Image Processing,* pp. 667-671.

Pandzic, I. S., & Forchheimer, R. (Eds.). (2002). *MPEG-4 Facial Animation. The Standard, Implementation and Applications*. England: John Wiley & Sons Ltd.

Pantic, M., & Rothkrantz, L. J. M. (2000, December). Automatic analysis of facial expression: the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12), 1424□1445.

Pardàs, M., & Bonafonte, A. (2002). Facial animation parameters extraction and expression recognition using Hidden Markov Models. *Eurasip Signal Processing: Image Communication*, 17(9), 675-688.

Parke, F. I. (1974). *A parametric model for human faces*. (Report No. UTEC-CSc-75-047). University of Utah: Computer Science.

Pentland, A., Moghaddam, B., & Starner, T. (1994). View-based and modular eigenspaces for face recognition. *Proceedings of the International Conference on Computer Vision and Pattern Recognition.*

Piat, F., & Tsapatsoulis, N. (2000). Exploring the time course of facial expressions with a fuzzy system. *Proceedings of the International Conference on Multimedia and Expo.*

Pighin, F., Hecker, J., Lischinski, D., Szeliski, R., & Salesin, S. (1998). Synthesizing realistic facial expressions from photographs. *Proceedings of ACM SIGGRAPH* 98, 75-84

Pighin, F., Szeliski, R., & Salesin, D. H. (1999). Resynthesizing facial animation through 3D model-based tracking. *Proceedings of the International Conference on Computer Vision.*

Platon (2000). French government project RNRT. Retrieved May, 26th 2003 from:
http://www.telecom.gouv.fr/rnrt/projets/res_61_ap00.htm

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257-285

RAT & VIC. (2002). Robust Audio Tool and Videoconferencing Tool. [Computer software, manual and project data]. Retrieved from
http://www-mice.cs.ucl.ac.uk/multimedia/software/rat and
http://www-mice.cs.ucl.ac.uk/multimedia/software/vic

Ravyse, I., Sahli, H., Reinders, M. J. T., & Cornelis, J. (2000). Eye activity detection and recognition using morphological scale-space decomposition. *Proceedings of the 15th International Conference on Pattern Recognition*, Vol. 1, pp. 1080-1083.

Sahbi, H., & Boujemaa, N. (2000). From coarse to fine skin and face detection. *Proceedings of ACM Multimedia 2000*, 432-434.

Sahbi, H., Geman, D., & Boujemaa, N. (2002). Face detection using coarse-to-fine support vector classifiers. *Proceedings of the IEEE International Conference on Image Processing*

Sarris, N., & Strintzis, M. G. (2001, July-September). Constructing a video phone for the hearing impaired using MPEG-4 tools. *IEEE Multimedia*, 8(3).

Schulzrinne, H., Casner, S., Fredderick, R., & Jacobson, V. (1996). *RTP: a transport protocol for real-time applications*. IETF: RFC 1889.

Serra, J. (Ed.). (1982). *Image analysis and mathematical morphology.* London: Academic Press.

Serra, J. (Ed.). (1988). *Image analysis and mathematical morphology. Volume 2: Theoretical advances*. London: Academic Press.

Shimizu, I., Zhang, Z., Akamatsu, S., & Deguchi, K. (1998). Head pose determination from one image using a generic model. *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition*, 100-105.

Similar (2003). Network of Excellence. Sixth European Research Framework. Information retrieved July, the 15th 2003, from
http://www.similar.cc

Simunek, M. (2003). Visualization of talking human head. (2003). Electronic version retrieved July, the 15th, 2003, from:
http://www.cg.tuwien.ac.at/studentwork/CESCG/CESCG-2001/MSimunek/

S--Z

Spors, S., & Rabestein, R. (2001). A real-time face tracker for color video. *Proceedings of the International Conference on Acoustics Speech and Signal Processing*

Ström, J., Jebara, T., Basu, S., & Pentland, A. (1999). Real time tracking and modeling of faces: and EKF-based analysis by synthesis approach. *Proceedings of the Modelling People Workshop at ICCV'99.*

Sturman, D. J. (1994). A brief History of Motion Capture for Character Animation. *Proceedings of SIGGRAPH*

Sum, K. L., Lau, W. H., Leung, S. H., Liew, A. W. C., & Tse, K. W. (2001) A new optimization procedure for extracting the point-based lip contour using active shape model. *Proceedings of the Int. Conf. Acoustics Speech and Signal Processing*.

Talking Heads Websites. (2002).
http://playmail.research.att.com/
http://www.winteractive.fr/
http://www.lifefx.com/

Tang, L., & Huang, T. S. (1994). Analysis-based facial expression synthesis. *Proceedings of the IEEE International Conference on Image Processing*. 98-102.

Terzopoulos, D., & Waters, K. (1993, June). Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6).

Thalmann, D. (1996, November). The Complexity of Simulating Virtual Humans *Supercomputing Review*. EPFL – SCR No 8

Tian, Y., Kanade, T., & Cohn, J. F. (2001, February). Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2), 97-115.

Trucco, E., & Verri, A. (1998). *Introductory to techniques for 3D computer vision*. Prentice Hall

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1).

UIB (2002). Universitat de les Illes Balears. Mathematics and Computer Science Department. Computer Graphics and Vision Group. Information retrieved from: http://dmi.uib.es/research/GV/

Valente, S. & Dugelay, J.-L. (2000, February). Face tracking and realistic animations for telecommunicant clones. *IEEE Multimedia Magazine*, 34-43.

Valente, S. (1999). Analyse, synthèse et animation de clones dans un context de téléréunion virtuelle [Analysis, synthesis and animation of clones within a virtual teleconference framework]. (Doctoral dissertation. École Polytechnique de Laussane, Inst. Eurécom, 1999).

Valente, S., & Dugelay, J.-L. (2001). A visual analysis/synthesis feedback loop for accurate face tracking. *Signal Processing: Image Communication*, 16(6), 585-608.

Valente, S., Andrés del Valle, A. C., & Dugelay, J.-L. (2001). Analysis and reproduction of facial expressions for realistic communicating clones. *Journal of VLSI and Signal Processing*, 29, 41-49.

Valente, S., Dugelay, J.-L. (2000, February). Face tracking and realistic animations for telecommunicant clones. *IEEE Multimedia Magazine*, 34-43.

Varakliotis, S., Ostermann, J., & Hardman, V. (2001). Coding of animated 3-D wireframe models for Internet streaming applications. *Proceedings of the IEEE International Conference on Multimedia and Expo.*

Video Cloning (1999). Video Cloning and Virtual Teleconferencing. Image and Video Group for Multimedia Communications and Applications. Institut Eurécom. Retrieved May, 26th 2003 from http://www.eurecom.fr/~image/Clonage/vc_mainpage.html

VRML (2003). VRML standard information provided by the Web3D Consortium. Retrieved June, 13th 2003 from http://www.web3d.org/vrml/vrml.htm

Waters, K. (1987, July). A muscle model for animating three-dimensional facial expression. *ACM Computer Graphics*, 21(4).

Wiskott, L. (2001, July 11). Optical Flow Estimation. Retrieved September, 26, 2002, from http://www.cnl.salk.edu/~wiskott/Bibliographies/FlowEstimation.html

Yacoob, Y., & Davis, L. (1994). Computing spatio-temporal representations of human faces. *Proceedings of Computer Vision and Pattern Recognition Conference*, 70-75.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), 1330-1334.

Zhenyun, P., Wei, H., Luhong, L., Guangyou X., & Hongjian, Z. (2001). Detecting facial features on image sequences using cross-verification mechanism. *Proceedings of the Second IEEE Pacific-Rim Conference on Multimedia..*

RESUME ETENDU EN FRANÇAIS

Note de l'auteur :

Tout d'abord, je voudrais m'excuser pour mon niveau de français. Ensuite, je voudrais ajouter que le but principal de ce résumé est de fournir une compilation des points les plus importants de ma thèse. Merci pour votre compréhension.

I Introduction

1 Motivation

Le clonage de visages est devenu un besoin pour beaucoup d'applications multimédia pour lesquelles l'interaction humaine avec les environnements virtuels et augmentés ameilleure l'interface. Son futur prometteur dans différents secteurs tels que la téléphonie mobile et l'Internet l'a transformé en sujet important de recherche. La preuve de cet intérêt est l'apparition croissante de compagnies offrant à leurs clients la création des visages synthétiques adaptés aux besoins du client et le soutien gouvernemental comme le projet européen INTERFACE (1999). Nous pouvons classer les visages synthétiques en deux groupes principaux : les avatars et les clones. Les avatars sont généralement une représentation approximative ou symbolique de la personne. Leur aspect n'est pas très précis. Ils sont indépendants des locuteurs parce que leur animation suit des règles générales indépendamment de la personne qu'on suppose qu'ils représentent. La plupart des visages synthétiques commerciaux actuels tombent dans cette catégorie. Les clones sont plus réalistes et leur animation tient compte de la nature de la personne : ils sont dépendants du locuteur.

Motivés par les avantages et les améliorations multiples qu'utiliser les caractères virtuels réalistes pourrait fournir aux télécommunications, nous voulons étudier la praticabilité de les employer dans les systèmes traditionnels de vidéoconférence, en utilisant uniquement un caméra. Cette dissertation couvre la recherche développée sur la création de nouveaux algorithmes faciaux d'analyse de mouvement et d'expression afin de replier le mouvement humain sur les modèles réalistes de visages qui seront employés dans des applications de télécommunication.

Le développement complet de notre cadre d'analyse est basé sur l'hypothèse qu'un modèle 3D réaliste du locuteur qui est devant la caméra est disponible. Nous croyons que des mouvements réalistes peuvent seulement être reproduits sur des modèles réalistes et, en ce cas, le modèle 3D est déjà disponible au système. L'information la plus précise obtenue à partir des séquences visuelles monoculaires prises dans des environnements standards (avec un éclairage inconnu ; aucun marqueur ; ...), peut seulement être obtenue si quelques données sur la géométrie de l'utilisateur sont connues, par exemple, en employant son clone réaliste, comme le faisons nous.

2 Contributions

Nous proposons de nouveaux algorithmes d'analyse d'image pour les traits spécifiques du visage (oeil, sourcils et bouche) qui essaient de profiter autant que possible de la physionomie et de l'anatomie du visage du locteur. D'abord, ces techniques ont été définies et examinées pour une position frontale :

Suivi de l'État d'Oeil : Nous avons développé des algorithmes indépendants de l'éclairage pour évaluer les mouvements d'œil. Ils emploient des contraintes anatomiques d'intra-trait naturelles pour obtenir le regard et le comportement de la paupière à partir de l'analyse de la distribution d'énergie sur la région de l'oeil. Nous avons également examiné la possibilité d'employer l'information de couleur pendant l'analyse. Nous interprétons les résultats d'analyse en termes de quelques unités spécifiques d'action que nous associons aux états temporels. En suivant un diagramme d'état temporel qui emploi des contraintes d'inter-trait pour placer la concordance entre les deux yeux, nous rapportons nos résultats d'analyse aux paramètres finaaux qui décrivent le mouvement de l'oeil.

Analyse de Mouvement de Sourcil : Pour étudier le comportement des sourcils des séquences visuelles, nous utilisons une nouvelle technique d'analyse d'image basée sur un modèle anatomique-mathématique de mouvement. Cette technique conçoit le sourcil comme un objet incurvé simple (arc) qui est sujet à la déformation due aux interactions musculaires. Le modèle d'action définit les déplacements 2D (verticaux et horizontaux) simplifiés de l'arc. Notre algorithme d'analyse visuelle récupère les données nécessaires de la représentation de l'arc pour déduire les paramètres qui ont déformé le modèle proposé.

L'analyse oculaire d'expression complète est obtenue après application de quelques contraintes d'inter-trait parmi les yeux et les sourcils. Ceci nous permet d'enrichir la quantité d'information de mouvement obtenue à partir de chaque trait, en le complétant avec l'information provenant des autres.

La Bouche : C'est la caractéristique faciale la plus difficile à analyser ; donc, nous croyons qu'une stratégie hybride pour dériver son mouvement devrait être utilisée : voix et image conjointement. Notre analyse est basée sur les faits suivants : le mouvement de la bouche peut exister même si aucun mot n'est prononcé et les actions sans parole de la bouche sont importantes pour exprimer l'émotion dans des communications. Cette thèse présente les premiers résultats obtenus à partir d'une technique d'analyse conçue pour étudier les aspects visuels du comportement de la bouche. Nous déduisons ce que sont les caractéristiques de la bouche fournies par le visage, les plus utiles lorsque les conditions d'illumination ne sont pas connues, et comment ces caractéristiques peuvent être analysées conjointement pour extraire l'information qui commandera le modèle de mouvement musculaire proposé pour son analyse.

La contribution principale de notre travail vient de l'étude de la connection de ces algorithmes avec l'information de la pose du visage extraite du système de suivi du mouvement rigide du visage. La technique présentée permet à l'utilisateur plus de liberté de mouvement parce que nous pouvons employer aussi ces algorithmes indépendamment de l'endroit où se trouve l'orateur comme possible.

Analyse d'Expressions Faciales Robuste au Mouvement de la Pose 3D du Visage : Le filtrage de Kalman est souvent utilisé dans les systèmes de suivi de visages pour

deux buts différents : d'abord, il lisse temporellement hors des paramètres globaux principaux estimés, ensuite, il converti les positions des observations 2D des traits faciaux en évaluations 3D et en prédictions de la position et de l'orientation principales du visage. Dans notre application, le filtre de Kalman est le noeud central de notre système de suivi : il récupère la position et l'orientation globales principales, il prévoit les positions 2D des points des traits pour l'algorithme d'appariement, et c'est le point exploité pour des applications de télécommunication, il fait au modèle synthétisé avoir la même échelle, position, et orientation que le visage du locuteur dans la vraie vue, en dépit d'avoir fait une acquisition par une caméra non calibré.

Après avoir déjà développé et testé positivement des algorithmes d'analyse de traits de visage pour des têtes étudiées depuis une perspective frontale, nous avons besoin d'adapter ces algorithmes à n'importe quelle pose. La solution que nous proposons définit les régions des traits à analyser et les paramètres des modèles de mouvement de chaque trait en 3D, au-dessus du modèle principal dans sa position neutre. Le procédé complet peur se résumer :

(i) Nous définissons et formons le secteur à analyser sur l'image. Pour faire ainsi, nous projetons le ROI 3D défini au-dessus du modèle du visage sur l'image en employant les paramètres de pose prédits, de ce fait obtenant le ROI 2D.

(ii) Nous appliquons l'algorithme d'analyse d'image du trait sur ce secteur extrayant les données demandées.

(iii) Nous interprétons ces données depuis une perspective tridimensionnelle en inversant la projection et les transformations dues à la pose (passage de données de 2D à 3D). En ce moment, nous pouvons comparer les résultats aux paramètres d'analyse du trait déjà prédéfinis sur le clone en position neutre et décider quelle action a été faite.

La technique que nous employons diffère d'autres approches précédentes puisque nous employons explicitement les données du clone pour définir l'algorithme d'analyse en 3D. Les avantages principaux de notre solution sont la commande complète de l'endroit et de la forme de la région d'intérêt (ROI), et la réutilisation d'algorithmes d'analyse d'image des visages déjà examinés qui sont robustes sur des visages qui regardent frontalement la caméra.

D'autres contributions : La thèse contient des analyses et des discussions au sujet du rôle de l'animation faciale dans les télécommunications. Nous avons également donné une description formelle de ce qu'est l'animation faciale en utilisant les modèles synthétiques en termes de génération et compréhension des paramètres de mouvement. Cette explication théorique permet la classification des systèmes d'animation faciale complets en comparant leur exécution concernant le degré de réalisme qu'ils permettent. Il décrit également un cadre pour comprendre le niveau de l'interopérabilité parmi différents systèmes d'animation faciale.

I Techniques d'Analyse d'Images Faciales et leur Principes Fondamentaux Reliés

Beaucoup de codeurs visuels font l'analyse de mouvement pour rechercher l'information de mouvement qui aidera la compression. Le concept de *vecteur de mouvement*, d'abord conçu à l'heure du développement des premières techniques visuelles de codage, est intimement lié à l'analyse de mouvement. Ces premières techniques d'analyse aident à régénérer les ordres visuels comme reproduction exacte ou approximative des séquences originales, en employant la compensation de mouvement sur les images voisines. Ils peuvent compenser mais ne peuvent pas comprendre les actions des objets se déplaçant dans la vidéo et donc, ils ne peuvent pas reconstituer les mouvements de l'objet sous un point de vue différent, ou immergé dans un scénario tridimensionnel.

Les visages jouent un rôle essentiel dans la communication humaine. En conséquence, ils ont été les premiers objets dont le mouvement a été étudié afin de recréer l'animation sur les modèles synthétisés ou interpréter pour le mouvement pour un usage postérieur. La Figure I-1 illustre l'organigramme de base pour des systèmes consacrés à l'expression et à l'analyse de mouvement facial sur des images monoculaires. La vidéo, ou encore des images, sont d'abord analysées pour détecter, commander et déduire l'endroit de visage sur l'image et les conditions environnementales sous lesquelles l'analyse sera faite (la pose principale, les conditions de l'illumination, les occlusions de visage, etc.). Puis, quelques algorithmes d'analyse de mouvement et d'expression extraient les données spécifiques qui sont finalement interprétées pour produire la synthèse de mouvement de visage.

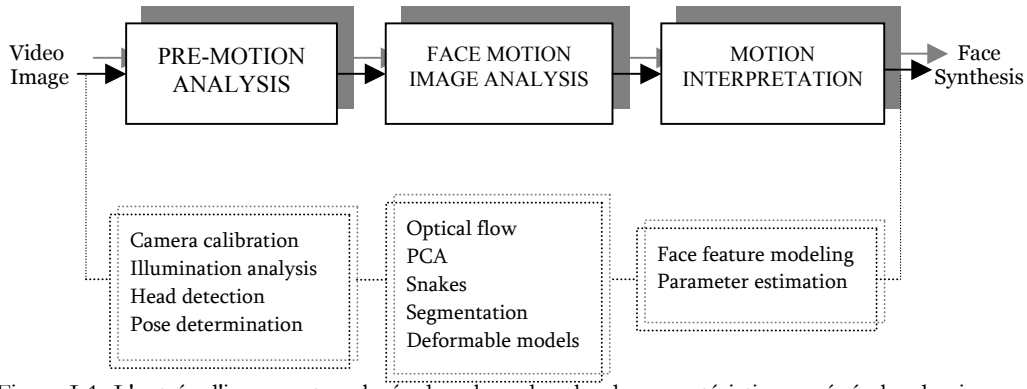


Figure I-1. L'entrée d'image est analysée dans la recherche des caractéristiques générales de visage globaux : mouvement, éclairage, etc.. À ce point le traitement d'image est effectué pour obtenir les données utiles qui peuvent être, ensuite interprétées pour obtenir la synthèse d'animation de visage

Chacun des modules peut être plus ou moins complexe selon le but de l'analyse (i.e., de la compréhension du comportement général à l'extraction du mouvement 3D exacte). Si l'analyse est prévue pour l'animation postérieure d'expression de visage, le type de synthèse de l'Animation Faciale (AF) détermine souvent la méthodologie utilisée pendant l'analyse

d'expressions. Quelques systèmes peuvent ne pas passer par les premières ou les dernières étapes, ou quelques autres peuvent mélanger ces étapes dans l'analyse d'image principale de mouvement et d'expression. Les systèmes manquant de l'étape d'analyse de pré-mouvement sont le plus susceptibles d'être limités par des contraintes environnementales comme des situations spéciales d'éclairage ou une pose principale prédéterminée. Ces systèmes qui n'effectuent pas l'interprétation de mouvement ne se concentrent pas sur le délivrance d'aucune information pour exécuter la synthèse d'animation de visage ensuite. Un système qui est censé analyser la vidéo pour produire des données d'animation de visage d'une manière robuste et efficace doit développer les trois modules. Les approches actuellement étudiées en recherche et celles qui seront exposées dans cette section effectuent clairement l'analyse de mouvement facial et d'expression et font l'interprétation de mouvement pour pouvoir animer. Néanmoins, bon nombre d'entre elles échouent à avoir une étape forte d'analyse de pré-mouvement pour assurer de la robustesse pendant l'analyse suivante.

Ce chapitre passe en revue des techniques courantes pour l'analyse des images simples pour dériver l'animation. Ces méthodes peuvent être classées en se basant sur différents critères :

1. la nature de l'analyse : global contre basé sur traits, orienté temps réel... ;
2. la complexité d'information a rechercher : génération générale d'expression contre le mouvement spécifique de visage ;
3. les outils utilisés pendant l'analyse, par exemple, la coopération d'un modèle 3D principal ;
4. le degré de réalisme obtenu à partir de la synthèse de l'animation de visage (FA) ; et
5. les conditions environnementales pendant l'analyse : l'éclairage commandé ou uniforme, l'indépendance de la pose du visage.

Dans cette section, les systèmes seront présentés dans trois catégories principales, classées par le rapport existant entre l'analyse d'image et la synthèse prévue d'AF, à savoir :

Méthodes qui recherchent l'information d'émotion : ce sont les systèmes dont l'analyse de mouvement et d'expression vise à comprendre le mouvement de visage d'une façon générale. Ces techniques évaluent les actions en termes d'expressions : tristesse, bonheur, crainte, joie, etc. Ces expressions sont parfois mesurées et puis interprétées par des systèmes de AF mais les techniques d'analyse ne se préoccupent pas par l'AF en elle-même.

Les méthodes qui obtiennent des paramètres liés à la synthèse d'AF employée : ceci inclut les méthodes qui appliquent des techniques d'analyse d'image au-dessus des images dans la recherche pour des mesures spécifiques directement liées à la synthèse d'animation.

Méthodes qui emploient la synthèse explicite des visages pendant l'analyse d'image : quelques techniques emploient la synthèse explicite du modèle 3D animé pour calculer des déplacements de nœuds du maillage, généralement par l'intermédiaire d'une boucle de rétroaction.

Indépendamment de la catégorie à laquelle ils appartiennent, plusieurs des méthodes qui exécutent l'analyse faciale sur des images monoculaires pour produire de l'animation partagent quelques techniques de traitement d'image et outils mathématiques.

II Clonage Réaliste d'Animation Facial

Évaluer l'animation des systèmes faciaux est une tâche ambiguë parce que des critères de qualité prédéfinis n'existent pas. La majeure partie du temps, le degré de réalisme et de naturalité de la reproduction faciale synthétique est déterminée à partir du jugement subjectif.

Ce chapitre contient la définition de quelques concepts théoriques liés à l'animation faciale. Nous avons essayé de formaliser la notion de réalisme dans le contexte de nos travaux de recherche. Nous visons à fournir une base conceptuelle où les notions d'avatar et de clone soient clairement énoncées. Ce cadre formel nous permet de décrire l'interaction existant entre la génération faciale de mouvement et sa synthèse depuis une perspective globale.

Nous concluons le chapitre avec quelques considérations au sujet du clonage de visage vu d'une perspective morale.

III Cadre Étudié de AF pour les Télécommunications

1 Introduction

La demande prévue pour les systèmes déployant de l'animation faciale fortement réaliste est large. L'animation faciale peut être utile en vidéo lorsque elle est utilisée pour communiquer par l'intermédiaire de plus nouvelles et flexibles liaisons comme, l'Internet ou la téléphonie mobile, qui n'ont pas des capacités élevées de débit et ne peuvent pas assurer la qualité optimale de service. Les communications mobiles de prochaine génération contemplent déjà la possibilité de conversations tête à tête. L'e-commerce, qui emploie des vendeurs virtuels augmente le contact avec des clients en utilisant les interfaces homme - ordinateur. L'industrie du jeu peut également tirer bénéfice d'employer des clones des joueurs au lieu d'avatars simples. En conclusion, quelques systèmes de communication avancés faisant participer plusieurs personnes (système de téléconférences visuel et virtuel) pourraient être conçus pour réduire le sentiment de distance entre les participants en présentant quelques éléments qui existent lors de vraies réunions avec des environnements artificiels mais réalistes. Dans ce sens, notre recherche a été faite avec l'esprit de développer des emplacements plus avancés de téléconférence. Il est important de noter que jusqu'ici, toutes les applications mentionnées auparavant ont préféré employer des avatars plutôt que d'animer insuffisamment des visages artificiels réalistes. Ceci justifie le grand effort et les ressources mises dans la recherche du clonage de visages et la pertinence de la thèse par rapport aux télécommunications de nos jours.

Pour créer un clone nous avons besoin d'un modèle 3D du locuteur fortement réaliste. Au contraire des avatars ou têtes parlantes (même si réalistes), le clonage de visages implique pour le système complet de la génération d'animation pour être dépendant du locuteur. Ce domaine tombe dans la catégorie plus grande de la réalité *virtualisée*, en opposition à la réalité virtuelle puisque le réalisme de la restitution n'est pas atteint à partir de rien par des techniques avancées de vision par ordinateur mais il est inspiré et contraint par de vraies données audiovisuelles du locuteur. Le clonage de visages est un exemple appropriée du phénomène récent de la convergence entre différents domaines de recherche : analyse d'image (i.e. traitement des signaux), synthèse d'image (infographie), et télécommunications.

Les visages synthétiques modelisés sont animés après les actions dérivées de l'interprétation de quelques paramètres d'animation. Produire des paramètres d'animation devient une tâche difficile si elle est faite manuellement ; donc des systèmes automatiques ou semi-automatiques de génération de paramètres ont été développés. Ces systèmes extraient l'information de mouvement du visage à partir du discours, de l'image ou de toutes les deux. Les synthétiseurs visuels de Texte-À-Discours (TTS visuel), qui se rapportent à ces TTS qui fournissent également la synthèse de visage, produisent de leurs paramètres d'animation du texte d'entrée donné au TTS. Le TTS visuel analyse le texte et fournit les phonèmes

correspondants. Ces phonèmes ont leur représentation synthétique de mouvement de bouche, assortie également de visemes, qui peuvent être synthétisés ultérieurement. Les TTS présentent plusieurs avantages : ils sont les systèmes d'analyse les plus simples pour produire des paramètres d'animation de visage, ils n'ont pas besoin d'interaction humaine et ils peuvent produire un mouvement tout à fait précis de bouche. Pour ces raisons, certains des produits d'animation de visage disponibles utilisent cette technique. Nous pouvons également utiliser la dualité phonème-viseme pour dériver l'animation de la parole. Dans ce cas, la parole est analysée pour déduire les phonèmes. Si nous extrayons les phonèmes à partir du texte ou à partir de la parole, l'inconvénient principal est qu'ils produisent seulement du mouvement automatique pour la bouche donc une autre source de génération d'action est nécessaire pour accomplir l'animation de visage. Ils donnent des résultats acceptables en animant les caractères non réalistes (dessins animés, animaux, etc.) mais puisque leur information produite n'est pas personnelle, ils donnent à peine un sentiment humain normal.

En plus de rendre plus réaliste l'animation faciale générique, nous avons besoin également des techniques d'analyse de mouvement pour étudier immédiatement les actions des locuteur à un moment donné. En utilisant l'AF dans les communications, l'environnement applicatif exige des méthodes d'analyse non envahissantes en temps réel pour produire des paramètres d'animation ; donc la plupart des approches adoptées pour adapter les systèmes d'animation ne sont plus utiles pour être appliquées aux communications.

2 Vue d'ensemble du Sytème

La Figure III-1 illustre le système que nous proposons pour le clonage facial de mouvements et d'expression. Pendant que l'information est analysée (ligne verte) sur le locuteur, principalement de l'information visuelle bien qu'il pourrait également être d'origine différente, est obtenu et employé pour reproduire le comportement facial (dénoté par λ), sur un modèle 3D du visage fortement réaliste. Les paramètres produits auraient pu être codés et envoyés pour être directement interprétés ; au lieu de cela, il est préférable de simuler le décodage et la synthèse pendant l'analyse d'image. Ajoutant cette rétroaction de synthèse, nous pouvons commander l'erreur commise et nous pouvons ajuster les paramètres analysés pour les adapter à un mouvement plus précis (α). Les données finales (μ) doivent être compréhensibles par le moteur d'animation faciale du décodeur dans l'emplacement à distance (copie orange), suivant la sémantique spécifique ou peut-être après avoir été adapté à une norme. L'utilisation d'un modèle principal fortement réaliste du locuteur nous permet non seulement l'utilisation d'une rétroaction visuelle commode et exploitable mais également la connaissance des données anthropométriques qui peuvent également être utilisées pendant l'analyse.

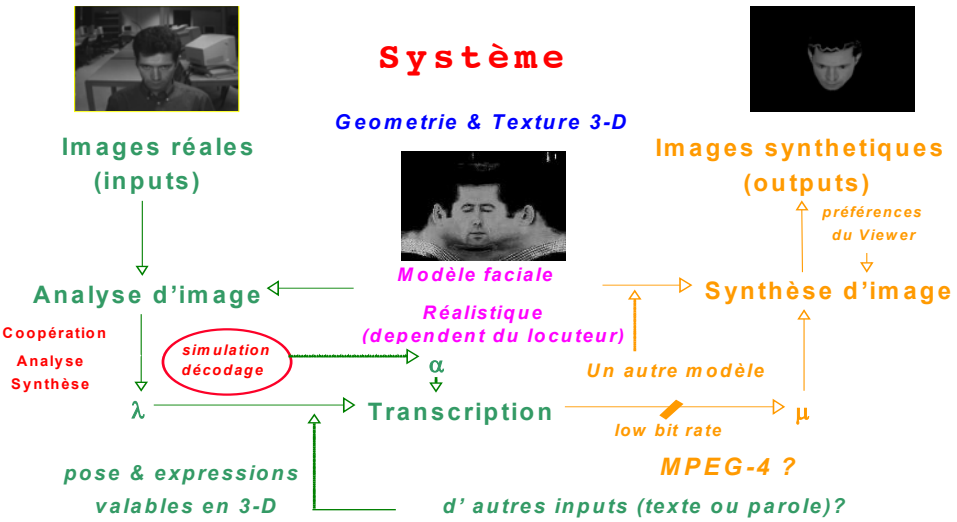


Figure III-1. En utilisant l'animation de clone pour les communications, là existent deux parts actives principales. Le générateur facial de paramètre d'animation (copie verte), qui est inclut dans la partie de codage/transmission et fait le traitement d'image lourd ; et le moteur facial d'animation (copie orange), qui est placé dans le récepteur et dont la tâche est de régénérer le mouvement facial sur le clone du locuteur en interprétant des faps. Le cadre ci-dessus a présenté à des utilisations la rétroaction synthétique d'image de clone d'améliorer l'analyse d'image et de produire de l'information plus précise de mouvement

Fondamentalement, le développement complet de ce système contient 4 blocs principaux : (i) acquisition ou création du modèle principal synthétique artificiel 3D, réaliste et dépendant du locuteur ; (ii) analyse vidéo d'un locuteur étant enregistré dans un environnement normal pour extraire des paramètres pour l'animation ; (iii) compression et transmission des paramètres entre l'encodeur et le décodeur ; (iv) synthèse du modèle 3D et de son animation des paramètres reçus.

Le noyau principal du travail de recherche présenté dans cette thèse : la stratégie d'association pose-expression pour l'analyse faciale de mouvement, a été développée pour satisfaire aux exigences du bloc (ii).

IV Analyse Faciale de Mouvement depuis une Perspective Frontale

1 Introduction

Les modèles développés de mouvement de trait utilisent pas seulement des contraintes anatomiques (intra-trait) pour dériver des actions du trait, ils emploient également des contraintes standard normales humaines de mouvement pour produire de l'animation faciale réaliste exploitant la corrélation existante entre les yeux, et entre les yeux et les sourcils (inter-trait). Les techniques de traitement d'image ont proposé l'essai globalement pour réduire au minimum l'influence des sources d'erreur inconnues et pour améliorer le comportement global comprenant en imposant quelques restrictions humaines standard de mouvement aux données obtenues à partir de l'analyse de chaque trait. Ils se conforment une stratégie d'analyse qui vise à fournir la compréhension logique de mouvement qui peut produire des données fiables d'animation pour reproduire synthétiquement des expressions faciales de trait d'entrée visuelle analysée.

Les algorithmes développés supposent que l'endroit et la délimitation de la région du trait d'intérêt (ROI) (yeux, sourcils ou bouche) sont connus. Cette hypothèse est réaliste dans le contexte actuel parce que, comme expliqué au Chapitre V, le procédé qui prolonge l'utilisation de ces algorithmes à n'importe quelle autre pose prend également soin du suivi et de la définition du trait ROIs.

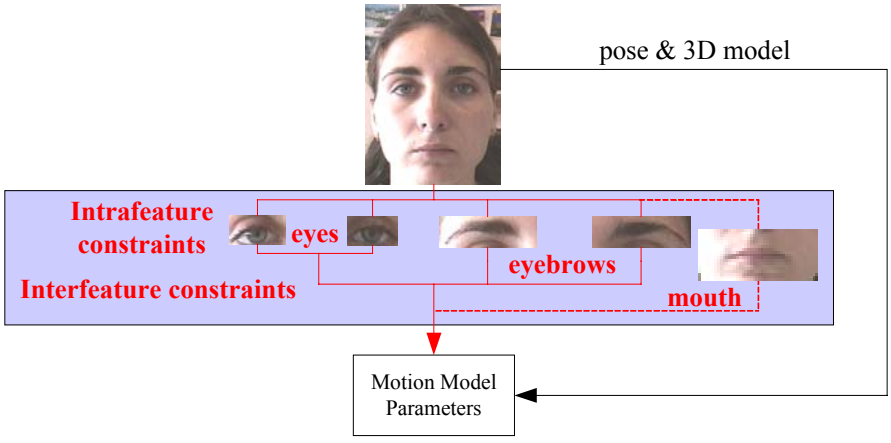


Figure IV-1. Diagramme général du cadre d'analyse proposé - les parties qui sont reliées à l'analyse faciale d'expression ont été accentuées

2 Algorithme d'Analyse de l'État des Yeux

L'importance du regard pour la communication humaine est significative. Le « regard est un comportement richement instructif dans l'interaction tête à tête. Il sert au moins cinq fonctions distinctes (...), écoulement de régulation de conversation, fournissant la rétroaction, l'information émotive communicante, communiquant la nature des rapports interpersonnels et évitant la distraction en limitant l'entrée visuelle »,(Garau et al., 2001). En développant de nouveaux systèmes de télécommunication pour la vidéoconférence la compréhension et la reproduction correctes du mouvement des yeux devient nécessaire. Un exemple de cela est le projet de recherche de Microsoft "GazeMaster", un outil visant à fournir une vidéoconférence avec un regard corrigé (Gemmell, Zitnick, Kang et Toyama, 2000).

En raison du vaste nombre d'applications où le mouvement des yeux obtenu par l'analyse d'image est utile (détection de leur fermeture dans la conduite, le codage basé sur modèle dans les télécommunications, la connaissance humaine des actions dans HCI, etc.), y existent beaucoup de techniques pour étudier l'activité de l'oeil sur des images monoculaires. Ce n'est pas le but de ce chapitre de revisiter toutes les méthodes possibles qui peuvent être trouvées dans différents domaines de recherche, mais nous regarderont quelques approches qui se relient à notre travail dans les communications visuelles.

Deux techniques principales ont été employées pour analyser le mouvement des yeux sur des images : APC et 'template' déformable (modelant mouvement), nous renvoyons le lecteur au Chapitre I pour les détails théoriques sur ces techniques. APC a été largement étudié pour analyser le mouvement facial, surtout couplé à l'utilisation d'un flot optique comme source de données de mouvement (Valente, 1999). La plupart des travaux récents préfèrent faire cette analyse par ACI (analyse composante indépendante) plutôt que d'employer APC (Fidaleo et Neumann, 2002). Dans les deux cas, leur inconvénient principal est la dépendance d'exécution sur les conditions environnementales de l'analyse, fondamentalement sur l'éclairage. L'utilisation des 'templates' de mouvement semble être la solution choisie pour rechercher des actions des yeux de manière robuste (Goto, Escher, Zanardi et Magnenat-Thalmann, 1999 ; Tian, Kanade et Cohn, 2001). Généralement, ces 'templates' de mouvement se composent par des ellipses et des cercles, représentant la forme de l'oeil, qui sont extraits à partir des images.

Si l'indépendance de l'éclairage est cherché, le flot optique ne peut pas être employé, et d'autres outils de traitement d'images, analyse en utilisant la morphologie mathématique, filtrage non linéaire, etc. sont utilisés. Travailler visant des conditions flexibles mène des chercheurs à rechercher des solutions où des résultats incorrects dans l'analyse devraient être compensés ou réduits au minimum, par exemple, en étudiant le comportement temporel des actions de l'oeil. Ravyse, Sahli, Reinders et Cornelis (2000) exécutent l'analyse des mouvements de l'oeil en employant une approche mathématique de mesure sur l'espace de

morphologie, formant les courbes spatio-temporelles hors des statistiques de mesures de balance. Les courbes résultantes fournissent une mesure directe du geste de l'oeil, qui peut alors être employé comme paramètre d'animation des yeux. Bien que dans leur article ils considèrent seulement l'ouverture et la fermeture des yeux, ils montrent déjà le potentiel d'employer l'évolution temporelle du mouvement des yeux pour leur analyse.

Notre approche suit la même philosophie d'analyse que celle présentée par Ravyse et al. Elle diffère dans le traitement d'image impliqué : nous proposons la déduction du mouvement par l'étude de l'emplacement de la pupille parce qu'il fournit le regard de l'oeil et les informations d'ouverture et de fermeture. Au lieu d'une analyse statistique, nous présentons un diagramme d'état temporel basé sur le comportement standard du mouvement humain qui contraint les actions en utilisant quelques restrictions intra-trait. En télécommunications, il est très important de produire des expressions faciales non dérangeantes. Il est déjà discuté par Al-Qayedi-Qayedi et Clark (2000), la connaissance du comportement humain standard peut être utile pour animer les yeux.

3 Algorithme d'Analyse de Mouvement des Sourcils

Historiquement, l'analyse de mouvement de sourcil a été moins étudiée que d'autres techniques d'analyse de traits (les yeux et la bouche). Dans la littérature nous constatons que le premiers essais pour analyser le comportement de sourcil (Huang, C.- L. Et Huang, Y.- M, 1997) sont concernés par la recherche d'information d'expression. Plus récemment, Goto, Kshirsagar et Magnenat-Thalmann (2001) ont également présenté une méthode pour analyser le mouvement des sourcils afin d'extraire des paramètres d'animation faciale. La méthodologie d'analyse suivie est plutôt heuristique et les approches proposées ne présentent pas l'influence des conditions environnementales. Kampmann (2002) propose une technique qui peut détecter les sourcils même si ils sont partiellement couverts par des cheveux. En général, nous n'avons trouvé aucune technique d'analyse de mouvement qui relie formellement les résultats de traitement d'analyse d'image à la génération des paramètres de mouvement.

Dans cette section nous décrivons une technique d'analyse de mouvement de sourcil où le traitement d'image a su adapter l'analyse aux caractéristiques de l'utilisateur et aux conditions environnementales. Nous rapportons les résultats de cette analyse d'image directement à un modèle de mouvement.

Pour étudier le comportement visuel des sourcils, nous utilisons une nouvelle technique d'analyse d'image basée sur un modèle anatomique-mathématique de mouvement. Cette technique conçoit le sourcil comme un objet incurvé simple (arc) qui est sujet à la déformation due aux interactions musculaires. Le modèle d'action définit les déplacements 2D (verticaux et horizontaux) simplifiés de l'arc. Notre algorithme d'analyse récupère les données

nécessaires de la représentation de l'arc pour déduire les paramètres qui ont déformé le modèle proposé.

4 Corrélation Spatiale entre les Yeux et les Sourcils : Étude des Expressions Extrêmes

Généralement, les modèles de mouvement le plus complexes sont le moins robustes en temps d'analyse aux conditions environnementales inattendues. Nos algorithmes d'analyse sont robustes grâce à leur simplicité. Cette simplicité limite le mouvement individuel propre à l'oeil et aux mouvements normaux et logiques du sourcil ; ces contraintes conviennent dans des communications homme/homme mais elle peut regrettablement filtrer certains détails qui ajoutent la force à l'expression, surtout, en présence d'émotions extrêmes (joie, colère, etc.).

Pour compenser partiellement cette limitation, nous proposons également d'exploiter la corrélation existante entre le mouvement de l'œil et du sourcil, pour enrichir l'expression oculaire globale provenant de l'analyse individuelle de chaque trait. Quand les yeux sont fermés les paupières peuvent se comporter de deux manières différentes, elles peuvent être fermées sans tension si les sourcils sont neutres ou tirés vers le haut ; ou elles peuvent être fortement fermées si les sourcils sont abaissés. Quand les yeux sont ouverts, le niveau de la taille du sourcil indique le degré d'ouverture de la paupière. La Figure IV-3 illustre cette corrélation évidente entre la paupière et le sourcil. Les actions extrêmes du sourcil déterminent et raffinent le mouvement de l'oeil en :

(i) prolongeant l'information à l'intérieur du diagramme d'état temporel de l'oeil pour inclure les contraintes d'inter-trait dérivés de l'analyse du sourcil. Par exemple, avoir en bas une action forte de sourcil aura assurément comme conséquence une action de fermeture de l'oeil, même si les données de l'oeil ne sont pas fiables (Figure IV-2),

(ii) dérivant le comportement final de la paupière synthétique d'ajouter à la position obtenue à partir de l'endroit de pupille un mouvement supplémentaire limité par la force du mouvement du sourcil :

(IV-1)
$$y_{eyelid}^{new} = y_{eyelid}^{former} + \mu \cdot fap + \eta$$

avec

$$\mu = \frac{y_{eyelid}\big|_{MAX} - y_{eyelid}^{former}}{fap\big|_{MAX} - fap\big|_{0}} \text{ et } \eta = \frac{y_{eyelid}^{former} - y_{eyelid}\big|_{MAX}}{fap\big|_{MAX} - fap\big|_{0}} \cdot fap\big|_{0}.$$

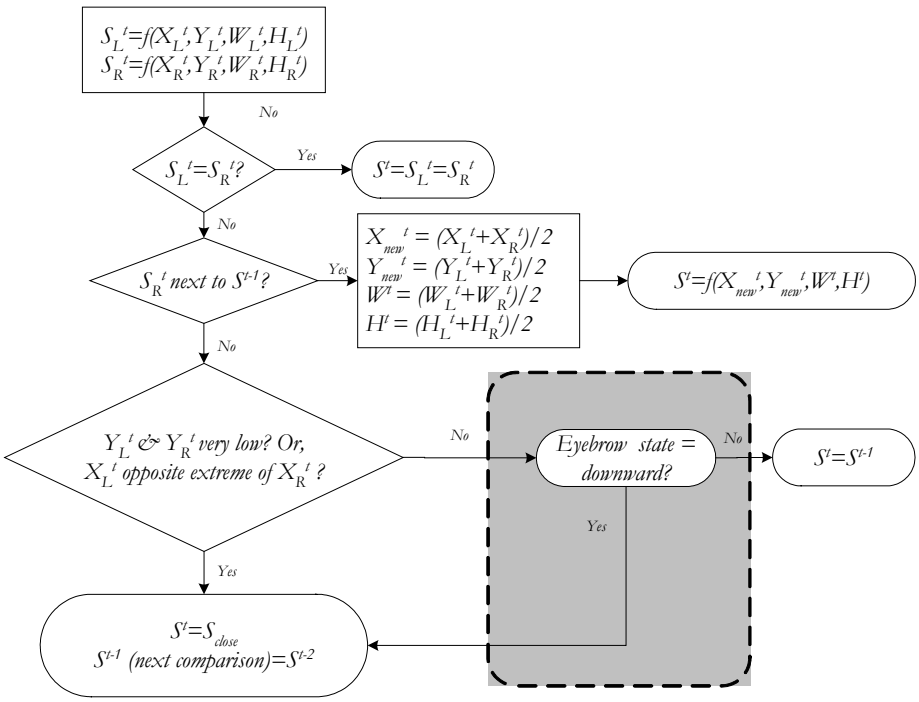


Figure IV-2. Le diagramme d'état temporel de base appliqué à l'analyse de l'oeil et établi sur seulement des contraintes d'inter-oeil peut être complété pour tenir compte des données obtenues à partir de l'analyse de sourcil

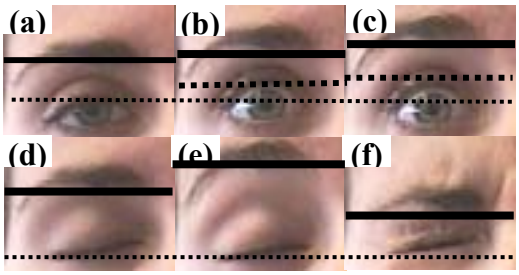


Figure IV-3. Quand l'oeil est fermé (rangée inférieure), le changement de paupière dû à l'action de sourcil peut être pris en tant que certaine animation spécifique. Quand l'oeil est ouvert (rangée supérieure) elle doit être prise en considération pour changer le mouvement vertical standard de la paupière

5 Analyse du mouvement de la bouche et des lèvres

5.1 Introduction

L'analyse du mouvement de la bouche a été étudiée depuis longtemps dans différents domaines. C'est devenu un champ de recherche large parce que plusieurs des techniques étudiées visent à fournir des outils utiles pour la vie quotidienne, comme par exemple, "lip-

reading automatique" pour les sourds. En suivant la philosophie de nos travaux de recherche, nous nous concentrons sur ces algorithmes qui aident à obtenir de manière plus efficace la transmission de l'information dans les systèmes de communication vidéo, en substituant la vidéo traditionnelle par l'animation de copies 3D des locuteurs. En effet, l'analyse de la bouche joue un rôle important dans ce scénario parce que l'exactitude du mouvement de la bouche et la synchronisation des actions de la bouche avec l'acoustique produite pendant la conversation sont cruciales pour obtenir des communications plaisantes et normales.

Nous pouvons considérer que le mouvement global de la bouche peut être vu comme résultat de deux facteurs :

$$M_{TOTAL} = M_{speech} + M_{expression}.$$

Là où M_{speech} représente le mouvement normal lié à l'articulation des bruits et des phonèmes tout en parlant et $M_{expression}$, est la partie du mouvement qui montre l'expression émotive et le comportement personnel de l'individu. Il est facile de distinguer les composants du mouvement venant de l'expression quand aucun discours n'est présent. Il est plus difficile de déduire comment les actions des deux natures agissent mutuellement l'un sur l'autre.

Examinant cette question de la perspective inverse, séparant des composants de mouvement de bouche en se basant sur leur nature (la parole ou l'expression) pendant l'analyse, est également un axe actuel de recherche dans la communauté d'animation faciale. Pendant la création du mouvement automatique à synthétiser sur les modèles 3D (habituellement avatars), nous combinons l'information phonétique de mouvement avec des données de mouvement d'expression. Cette combinaison doit être faite de telle manière que le comportement facial résultant agisse d'une manière naturelle. Dans la plupart des cas l'interaction phonétique et d'expression ne mènent pas à des résultats plaisants et normaux. La connaissance de l'interaction musculaire et du comportement facial normal doit être employée pour déduire le bon mouvement et pour adapter l'animation produite après avoir ensuite analysé l'aspect visuel des actions de la bouche.

Pour développer un cadre complet d'analyse, nous avons étudié les avantages et les inconvénients de la plupart des méthodes trouvées dans la littérature. Nous avons dérivé une approche qui convient à notre scénario en développant un modèle simple de mouvement pour s'assurer que ses paramètres d'action seront détectés de manière fiable pendant l'analyse, indépendamment des conditions environnementales.

V Prolongation de l'Utilisation des Modèles de Mouvement Frontaux à Tout Autre Pose

1 Introduction

Dans la littérature nous avons trouvé deux approches principales pour adapter le mouvement facial frontal et les algorithmes d'analyse d'expression :

1. Créer un 'template' de trait par chaque pose : après avoir développé et examiné des 'templates' de mouvement sur les visages frontaux, ils sont redéfinis en se basant sur différentes poses prédéterminées du visage. Par exemple, c'est la solution donnée par Tian, Kanade et Cohn (2001). Ils surmontent la limitation de pose dans leur analyse en définissant "un modèle d'états multiples de visage", où différents modèles sont employés pour différents états principaux (à gauche, à droit, vers le bas, etc.). Cette stratégie d'analyse est limitée. La complexité de cette solution augmente avec le nombre d'états, qui seront grands si beaucoup de liberté de mouvement est recherchée.
2. Rectification de l'image d'entrée : l'image à analyser est transformée pour obtenir une approximation du visage vu d'une perspective frontale. Puis, les algorithmes de traitement d'image définis pour les visages frontaux analysent cette nouvelle image pour obtenir les 'templates' correspondants du trait (Chang, et al., 2000). Cette solution fonctionne bien pour de légers mouvements rigides. Des rotations et les translations significatives ne peuvent pas être compensées avec des transformations simples d'image parce que :
   - l'aspect de chaque trait du visage dépend non seulement de la projection due à la pose mais également de sa forme 3D, donc une rectification 2D faite sans reconnaître la nature 3D du trait ne peut pas être précise ;
   - l'image rectifiée peut manquer quelques secteurs occlus sur l'image originale ;
   - et la rectification 2D peut changer la perception d'éclairage et la forme anatomique des traits, qui est très importante dans l'analyse d'image basée sur des traits ou caractéristiques faciales.

Nous proposons une approche différente pour faire l'adaptation frontale de l'analyse de mouvement. Notre solution emploit la connaissance de la pose principale et de la physionomie de l'utilisateur pour interpréter les expressions dans l'espace 3D au lieu de procéder sur l'image.

2 Adaptation des 'Templates' des Traits

Le procédé algorithmique d'adaptation suit ces étapes :

(a) Nous redéfinissons d'abord le modèle de mouvement, la région d'intérêt (ROI) et les paramètres de traitement d'image liés à chaque 'template' de trait en 3D, supposant que la tête fait face à la caméra, dans sa pose neutre.

(b) Après, nous employons l'information concernant le mouvement rigide du visage sur l'armature analysée pour projeter le ROIs défini par le 3D et d'autres contraintes d'analyse de chaque trait sur l'image visuelle. Puis, nous appliquons le traitement d'image pour extraire les données.

(c) Finalement, nous inversons la projection et la transformation de pose de ces données pour obtenir leur équivalent 3D qui sera prêt à être comparé avec les modèles de mouvement que nous avions déjà défini dans 3D.

La figure V-1 présente une interprétation graphique du procédé d'adaptation appliqué à l'analyse des traits de l'oeil.

Pour l'analyse adaptée nous devons définir :

(i) **un modèle d'observation**. Pour développer l'adaptation, nous considérons notre scénario d'analyse : un objet 3D (tête) devant un caméra qui acquiert les images visuelles qui sont analysées. Nous établissons la pose neutre de la tête, quand le visage est complètement porté sur l'image et regarde statiquement vers le centre de la caméra. Le modèle d'observation décrit mathématiquement le rapport entre les coordonnées de l'objet principal dans sa pose neutre et la vue finale du visage sur la vidéo. Ce modèle mathématique nous permet d'interpréter des données associées au visage 3D modelé dans l'espace d'image (2D) et vice versa.

(ii) **un modèle 3D de la tête.** Les techniques d'analyse de 'template' de mouvement définies pour une vue frontale aident à connaître l'endroit des traits de visage sur l'image. De même, pendant l'adaptation nous devons savoir la physionomie de la personne faisant face à la caméra et ainsi pouvoir localiser exactement les traits dans l'espace 3D. Nous employons une représentation 3D fortement réaliste de la personne pour déterminer la position du ROI de chaque trait.

(iii) une approximation de la surface de chaque trait. Les modèles d'analyse sont à l'origine définis pour analyser l'information dans le plan d'image. Nous

pouvons facilement adapter ces modèles de mouvement en traçant directement chacun d'eux sur une surface parallèle à ce plan image et située à l'emplacement déterminé du trait sur la tête 3D dans sa pose neutre. Pour obtenir le plan parallèle le plus approprié, nous développons l'approximation linéaire de la surface qui couvre la région du mouvement de chaque trait.

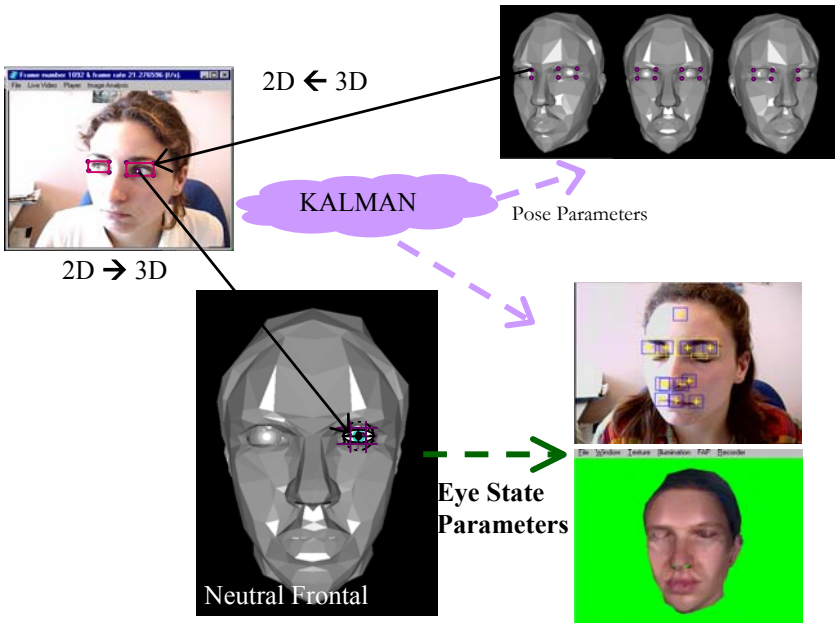


Figure V-1. Ce diagramme illustre le procédé général d'adaptation appliqué à l'algorithme d'analyse de l'oeil. D'abord, les sommets qui définissent le ROI 3D sur le modèle extérieur linéaire sont projetés sur le plan image. Alors l'algorithme de traitement d'images recherche l'information désirée en analysant l'intérieur le secteur délimité. Pour comprendre le mouvement du trait, des données sont interprétées dans l'espace 3D, au-dessus du modèle de mouvement qui a été défini sur l'approximation extérieure linéaire du trait de l'oeil vu d'une perspective frontale. Une fois que le mouvement est interprété il peut être reproduit sur un modèle principal synthétique.

VI Évaluation Technique du Regroupement du Suivi de la Pose Facial avec l'Analyse d'Expression

1 Introduction

Le travail compilé dans ce rapport de thèse a été la suite de la recherche scientifique sur l'analyse faciale pour l'animation synthétique que le groupe d'image du département de communications multimédia de l'Institut Eurécom avait réalisée les 6 dernières années.

Concernant l'analyse du mouvement facial rigide, le groupe de recherche avait développé un algorithme qui utilise une boucle de rétroaction à l'intérieur d'un filtre de Kalman pour obtenir des informations précises sur l'emplacement de la personne dans l'espace. Le filtrage de Kalman a été appliqué avec de bons résultats (Cordea, Petriu, E. M., Georganas, D., Petriu, D. C., Et Whalen, T. E., 2001 ; Ström, 2002) et permet la prédiction des paramètres de translation et de rotation de la tête du suivi 2D des points spécifiques du visage sur le plan d'image. Pour l'analyse faciale de mouvement, quelques techniques intéressantes pour l'analyse d'expression avaient été déjà examinées (Valente, 1999) mais l'approche basé sur APC, retenue à l'origine, avait trop de restrictions en adaptant son utilisation à n'importe quelle autre pose. Ceci nous a amené à développer la technique de couplage de pose et expression étudiée ici.

La caractéristique pratique principal de notre système de suivi est le besoin d'informations 3D sur la forme de la tête que nous analysons. Ceci implique que nous devons employer un modèle qui fournit les coordonnées 3D précises des points dont la projection est dépistée sur l'image et le donner au filtre pour obtenir la prévision des paramètres de pose. Très souvent, un modèle facial général est employé. L'inconvénient apparent peut devenir un avantage fort si une représentation 3D synthétique réaliste de l'utilisateur est disponible. Dans (Valente et Dugelay, 2001), nous avons prouvé que l'amélioration de la quantité de liberté de mouvement devant la caméra est possible si on utilise le clone du locuteur pendant le suivi. Les modèles doivent être une représentation 3D précise du locuteur, dans la forme et la texture, parce que notre approche compare les modèles au niveau d'image.

Nous avons inséré les algorithmes d'analyse de mouvement de caractéristiques faciales présentées dans ce rapport à l'intérieur du cadre de suivi original. Puisque le modèle réaliste du locuteur est nécessaire pour le suivi, et donc aussi disponible pendant l'analyse, les données 3D exigées pour prolonger l'utilisation des 'templates' d'analyse de mouvement sont obtenues à partir d'eux.

Pour que le procédé d'adaptation soit possible, l'algorithme de cherche de la pose et le traitement d'image doit partager le même modèle d'observation. D'autres analyses de suivi fonctionnent en utilisant des systèmes de référence d'image 2D sur lesquels ils peuvent seulement estimer la position de l'utilisateur sur l'écran (e.g. Algorithme 'CAMSHIFT' de

Bradski (1998)). Pour les processus à suivre, i.e., la détection des traits à analyser, l'analyse et l'interprétation des résultats obtenus, cette information n'est pas assez précise.

Dans notre approche, l'algorithme fonctionnant dans le plan d'image extrait les traits 2D à suivre, sur l'image synthétisée du modèle, sur lequel les paramètres de pose prédits ont déjà été appliqués. En ce moment, il fournit une vue ajustée de l'utilisateur dans sa future pose. La structure du système permet également de projeter des points appartenant aux secteurs. La figure VI-1 montre deux captures d'écran où on observe l'adaptation en ligne de la vue 3D du modèle.

Des détails au sujet du travail passé et actuel du laboratoire de recherche peuvent être trouvés au site Web du groupe (Face Cloning, 1999).

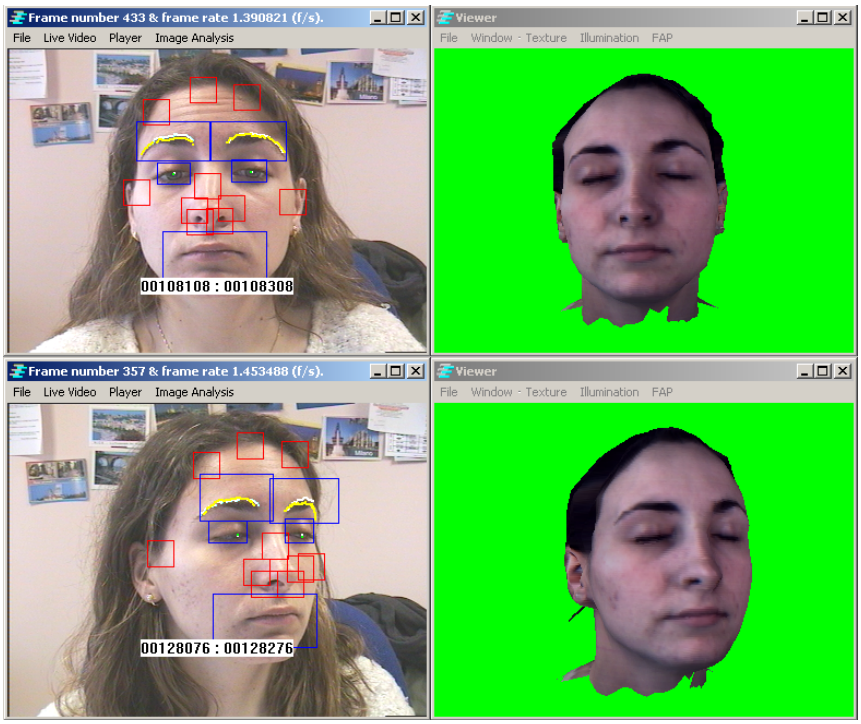


Figure VI-1. Deux projectiles d'écran des arrangements d'essai. Sur la fenêtre la plus à gauche nous présentons l'entrée visuelle, la projection des résultats d'analyse, et l'évolution du ROIs, appropriée à l'inspection visuelle de l'exécution d'analyse de trait ; sur la fenêtre la plus à droite la reproduction synthétique (projetée en utilisant OpenGL) du clone de l'utilisateur est représentée, nous permettant de commander l'évolution de l'algorithme de cheminement principal. Puisque nous employons un modèle fortement réaliste pour exécuter le cheminement nous utilisons ses données 3D pour faire l'adaptation algorithmique : nous redéfinissons les modèles d'animation de mouvement et leur ROIs là-dessus

Conclusions et Perspectives

1 Conclusions des contributions principales

Deux faits ont conduit la recherche dans la vidéoconférence. Techniquement, des manières plus efficaces de coder et transmettre l'information visuelle sont cherchées ; socialement, de nouveaux contextes de communication où l'information visuelle augmente l'interaction des locuteurs sont étudiés. Les nouvelles tendances des télécommunications considèrent réaliser ces buts en employant des données synthétiques. Dans un système de téléconférences virtuel, qui est le cadre de notre recherche, des locuteurs sont substitués par les modèles 3D synthétiques – des clones (réalistes) ou des avatars (symboliques). Le signal vidéo régulier est remplacé par un nombre limité de paramètres d'action déterminant le mouvement facial, de ce fait réduisant la bande exigée pour la transmission. En plus, en synthétisant les modèles dans un environnement interactif commun une situation normale de communication est recréée, i.e., où les personnes sont physiquement présentes.

En utilisant l'animation faciale dans les télécommunications nous transmettons des paramètres faciaux d'animation d'un générateur de faps et nous les rendons sur le récepteur. L'émetteur et le récepteur doivent partager la mêmes syntaxe et sémantique pour les faps afin de placer des communications logiques. En effet, les techniques de codage utilisées pendant le traitement des signaux ne doivent pas changer la syntaxe ou la sémantique ; autrement il dérangera les communications de manière significative. La perte de données ou l'interférence dans des suites de faps pourrait avoir un effet beaucoup plus mauvais que la rupture visuelle normalement due à la coupure dans les communications. La norme MPEG-4 a établi quelques points spécifiques de décodage à employer pour l'animation faciale. La norme indique la syntaxe commune pour décrire le comportement du visage, de ce fait permettant l'interopérabilité parmi différents systèmes d'animation de visage. En l'état actuel de l'évolution et du développement des applications conformes avec MPEG-4 plusieurs soucis sont apparus : Cette norme est-elle une solution globale que tous les systèmes spécifiques d'animation de visage peuvent adopter ? Ou, la syntaxe limite-t-elle trop la sémantique du mouvement possible ? Aucune réponse a été fournie, l'existence de tous ces doutes prouve qu'il reste beaucoup de chemin à parcourir pour maîtriser l'animation de visages et plus concrètement, la génération automatique de mouvements de visage réalistes.

La compréhension du comportement facial non verbal, particulièrement des yeux et des sourcils, critique pour produire de l'animation faciale normale et logique sur les modèles synthétiques en utilisant l'entrée classique de système de téléconférences (vidéo monoculaire). Spécifiquement, l'analyse faciale d'expression sur des images monoculaires est devenue un point clé important à aborder dans les domaines suivants :

Infographie (CG.) : pour créer des animations réalistes ;

Traitement d'Images (TI) : pour le codage basé sur des modèles;

Vision par Ordinateur (CV) : pour l'analyse d'expressions dans la reproduction d'image ;

Interaction Homme-Machine (IMM) : pour faire les machines réagir au comportement humain.

Les différentes approches adoptées pour réaliser l'analyse d'expression dans chaque domaine dépend de deux facteurs. D'abord, il dépend de la quantité de données détaillées de mouvement requises ; par exemple, plus d'information de mouvement est nécessaire dans le CG. que dans HCI. En second lieu, les méthodes diffèrent au niveau de la compréhension exigé dans leurs applications ; dans HCI nous devons comprendre quel genre d'action s'est produit, en identifiant un sentiment par exemple, tandis que dans le CG. ou l'IP, nous avons seulement besoin de la réplique de mouvement. Dans tous les cas, il devient crucial de maîtriser l'influence de la pose sur l'expression finale qui apparaît dans le visage sur l'image.

Le développement d'une analyse visuelle complète où le suivi de la pose du visage et l'analyse des expressions faciales sont séparément traitées aident à concevoir des algorithmes spécialisés d'analyse d'image ajustés par rapport aux besoins, des caractéristiques de trait, etc. Algorithmes qui sont universellement utilisables manquent de la précision. En effet, si aucune hypothèse précédente n'est prise alors, rendre approprié l'analyse à tous les cas implique un bon nombre de calculs et donc la perte de possibilité en temps réel. Pour compenser cette restriction, nous pouvons produire des algorithmes d'analyse moins précis (en employant des modèles simples de mouvement) mais en maintenant dans l'esprit la possibilité d'améliorer la complexité du système ; car les conditions informatiques deviennent de moins en moins restrictives, un arrangement flexible d'analyse nous permettra d'augmenter la complexité et d'extraire des données plus détaillées de mouvement.

2 Perspectives

Le cadre proposé d'analyse faciale de mouvement et d'expression a été examiné sur les traits les plus actifs ; il peut également être facilement adapté pour analyser n'importe quelle autre partie sur le visage, par exemple, rides et sillons. Un futur défi apparaîtra au moment du couplage des traits faciaux avec ke système de suivi sur de Kalman proposé. Car, plus de traits sont susceptibles de se déplacer, moins de points faciaux de suivi fixes seront disponibles pour l'algorithme. À ce point, des études au sujet de la robustesse de la pose contre le nombre de traits et la liberté de mouvement pour le locuteur devront être faites. Des solutions complémentaires pourraient être employées. Nous citerons, par exemple, l'insertion de la rétroaction visuelle complète du clone, c.-à-d., obtenant non seulement la rétroaction visuelle du mouvement rigide mais également de l'expression faciale comme complément pour le suiveur de pose basé sur Kalman. Nous pourrions même rechercher un algorithme qui

remplace le traqueur qui puisse compter sur les mêmes caractéristiques géométriques et qui puisse également fonctionner dans l'environnement actuel.

Le système présenté dans cette thèse est l'intégration de plusieurs modules qui peuvent fonctionner indépendamment les unes des autres. Nous pouvons profiter de ce fait en réutilisant les modules d'analyse séparément dans différents contextes et applications. Par exemple, algorithme proposé pour le suivi de l'état de l'oeil pourrait être employé sur les enregistrements visuels à grande vitesse des patients dans la recherche médicale pour les modèles corrélés de l'activité de cerveau.

Bien que la technique décrite ci-dessus ait été adressée comme solution pour analyser des expressions faciales pour obtenir des paramètres d'animation pour le mouvement facial synthétique, elle peut également être utilisée dans d'autres domaines scientifiques où la connaissance des actions instantanées de la personne devant la caméra est désirée, par exemple, dans l'analyse d'interaction homme-machine (Andrés del Valle et Dugelay, 2003). Les gens peuvent comprendre l'action faciale même lorsque les visages sont sous une éclairage très mauvais ou en présence d'objets inconnus au-dessus d'eux. C'est fondamentalement dû au fait que les humains peuvent réduire automatiquement la complexité de l'analyse dans différentes parties et faire cette analyse progressivement. D'abord, nous examinons les conditions dans lesquelles le visage est et nous décidons si davantage de compréhension est possible ; puis, nous ciblons la tête et obtenons son mouvement rigide (sa pose) et finalement, nous prêtons attention aux différents détails du visage qui nous sont intéressants parce qu'ils contiennent l'information d'expression. Quand les humains ne peuvent pas exécuter une analyse approfondie (l'éclairage est très mauvais, ou une partie significative du visage est occlue), ils compensent l'information absente (comportement humain standard généralement) ou ils acceptent simplement de ne pas comprendre le mouvement de visage qu'ils observent. Le cadre présenté est conçu pour exécuter l'analyse faciale de mouvement et d'expression sur des images monoculaire en essayant de reproduire cette conduite humaine normale et intuitive.

L'intérêt pour la compréhension faciale de mouvement augmente. Le nouveau réseau européen de l'excellence (NoE) SIMILAR (2003) a joint l'effort de plusieurs établissements qui visent à développer des outils comme le système virtuel de téléconférence visé par notre recherche. Entre d'autres activités, ce NoE développera des applications basées sur des techniques semblables à celle présentée ici, aussi fera la recherche pour améliorer des algorithmes actuels et placera les bases pour un réseau européen global dans l'interaction homme-machine multimodale.