Appears in: 2^{nd} *Symposium "Advances in Digital Image Communication", April 25, 1997, Erlangen, Germany*

Model–Based Coding and Virtual Teleconferencing

Stéphane Valente, Jean–Luc Dugelay

Multimedia Communications Department
Institut EURÉCOM
2229, route des Crêtes,
B.P. 193,
F-06904 Sophia–Antipolis Cedex
E–mail: {valente,dugelay}@eurecom.fr

Abstract

This paper presents early results in the context of a televirtuality project (named TRAIVI). The goal of this project is to create and run virtual meetings via low bit–rate links and virtual reality paradigms. We propose to enable several persons located at different physical sites to meet each other in a common virtual meeting room. In order to preserve a high level of realism, we describe how to animate $3D$ human–like synthetic interfaces based on CYBERWARE models.

1 Introduction

Classical teleconferencing systems generally rely on waveform signal processing techniques, and try to efficiently encode the redundancy of images considered as stochastic signals. In the case of multisite teleconferencing, such systems lead at best to views (like figure 1) where each site is represented by a distinct picture. However, due to bandwidth requirements, multiple views transmission is not always possible, and the different sites will be displayed alternatively to show the current speaking person. In any case, people may experience a distance feeling, and they have the impression neither to be in the same room, nor to share a common meeting.

Model–based coding, which considers images as being perspective projections of physical $3D$ objects, can cut off the bandwidth requirements with *a priori* knowledge of the involved objects. Associated with virtual reality techniques, a model–based system can clearly outperform

Figure 1: Multiple views representing the different speakers.

waveform–based systems in terms of compression capability and human communication aspects. The key idea is to provide speakers with a common meeting space as if they were all meeting in the same physical room, to synthesize the individual points of view they would naturally experience, and to give them the opportunity of having eye contact with each other via synthetic $3D$ models of the participants (clones), in short, to obtain a synthetic view like Figure 2. In order to achieve a good level of realism, several audio and video techniques have to be investigated and implemented. Among them are audio spatialization and multiplexing, echo cancelation, video spatialization, speakers face cloning, audio–video synchronization... purposedly to get a perfect immersion into the meeting space.

1.1 Related Work

A virtual teleconference system is in fact related to two different issues found in the literature (aside networking aspects): being able to reproduce the participants images, and providing a virtual environment to immerse those images with the real users.

In recent years, research has been conducted

Figure 2: Example of a synthetic meeting as it could be viewed by a fourth participant.

on topics related to teleconferencing and model–based coding such as human image analysis and reproduction [1], and human face cloning [2], that can be useful to reproduce the participants. Historically, all facial animation research works on the basis of a generic face model (the well known CANDIDE model is the most common), activated by a few control nodes according to the FACS system [3]. Whereas such an approach might give unrealistic faces as results [4], more and more research teams try to improve the synthesized generic models: the INA (Institut National de l'Audiovisuel, France) builds a special texture to be mapped onto the face model [5], Reinders *et al.* adapt a generic face model for individuals [6], and Choi *et al.* obtain amazingly realistic expressions with their textured model [7]; however, the inverse idea has not been investigated yet in the literature: start with a model depending on a person, and make it more generic so that it can be handled by an automatic framework. This is what we propose to do in our research work on face cloning.

As far as building virtual environments is concerned, some experiments have also been done via the TELEPORT project with a wall–sized display. It uses a synthetic $3D$ scene that carefully matches the room in which the display is located, and video images of remote participants are blended into the virtual extension of the room [8]. Once again, instead of building a special $3D$ environment, we would like to start from existing ones to be closer to what users are used to in the real world.

1.2 The TRAIVI Project

The TRAIVI[1] project aims at implementing virtual meeting rooms over low–rate bindings with a high level of realism. Some authors, like Kanade [9], made clear that *virtualized reality* is superior to *virtual reality* since it takes into account the real world fine details and not a simplistic CAD model. This is the reason why we want to use $3D$ textured wire frame models scanned from real people (CYBERWARE models) which are everything but generic to perform the face cloning. Our primary goal is neither to build an imaginary world that has no equivalent in reality, nor to exactly synthesize the real world and the true speakers expressions, but to render the real world in a way that is visually coherent and *comfortable* for its users. The same philosophy stands true for the meeting space environment with video spatialization [10]: this technique uses real room uncalibrated pictures and *virtualizes* them to create the meeting room views, as opposed to building $3D$ room models from scratch.

Some European projects, namely HUMANOID [11] and its continuation VIDAS [12], are investigating interpersonal audio/video communication using virtual reality paradigms, their two main objects being also the $3D$ talking interfaces and the background, each of them optionally natural, synthetic or hybrid. Our approach can be compared to their "natural" methodology.

This paper presents how our clone models are handled both globally (their pose in the $3D$ space) in section 2, and locally (their facial expressions) in section 3.

2 Global Animation

Global animation consists in determining the speaker's pose in the $3D$ space by image analysis techniques in real–time, and to use the extracted parameters to render the clone in a coherent manner. Our analysis scheme achieves it without requiring to tape marks on user's faces.

In this section, we will describe which parameters are analyzed from the video input, and how they are tracked.

[1] TRAIVI stands for "TRAItement des images VIrtuelles" (Virtual Images Processing)

2.1 Extracted Parameters

To derive the degrees of freedom of the head global motions, we use the parameters obtained by image analysis and feature tracking showed in Figure 3: the face outline is materialized by the rectangular window W, and within the window, we detect the eyes L and R, the eyes horizontal axis H and the vertical one V. The head degrees of freedom are then evaluated by:

left/right and up/down translations: given by the window W center coordinates

forward/backward translation: derived from the width of W

left/right rotation: given by the position of V within W

up/down rotation: given by the position of H within W

last rotation degree: given by the angle between the eye positions L and R

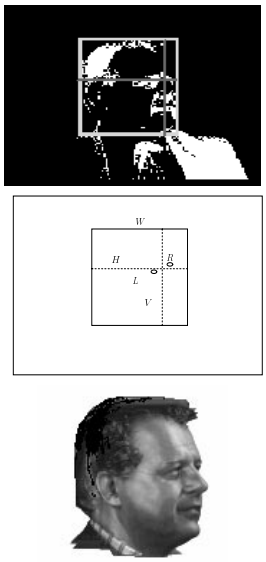

Figure 3: Analysis parameters and $3D$ model pose

One could object that this derivation does not output *calibrated* values for the different degrees of freedom, but this is of no importance since the actual clone scale and (x, y, z) position will mainly depend on the $3D$ environment setup. The extracted global position parameters will only be used to animate the clone in a visually coherent manner around its virtual position.

2.2 Analysis and Tracking Algorithm

We are now going to explain how the parameters in Figure 3 are estimated in a video sequence. We proceed in two steps: first we determine the head outline (referred as window W in the figure) and then, all other parameters by tracking the speaker's eyes within W.

2.2.1 Head Outline Determination

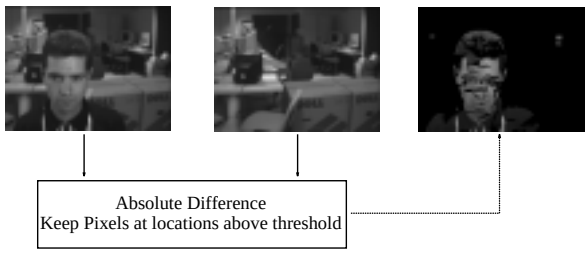

Figure 4: Speaker's outline thresholding.

Our head outline tracking algorithm assumes that the background behind the speaker remains static throughout the session, for three reasons: the first reason is that this assumption enables us to simply subtract the current speaker image from a background reference view and threshold the difference to get the outline (Figure 4); then, we compute the binary horizontal and vertical histograms to find the top, left and right edges of the head (Figure 5). We do not look for the bottom edge of the head, since it is not necessary regarding the parameters mentioned in section 2.1; the second reason is that we can all the more afford to "cut" the speaker from the background as its clone will be inserted in another room, and each clone will have its own background depending on the viewer's position in the virtual space; and the third and last reason is that it will be possible to later develop a procedure to update the reference background image when there are illumination changes;

This algorithm has been extended to achieve tracking by using the formerly found values for a new video frame: the previous window W is slightly widened and used as a search region to compute the binary histograms. If the head edges cannot be found, the search region is extended again, and ultimately, the whole frame is searched.

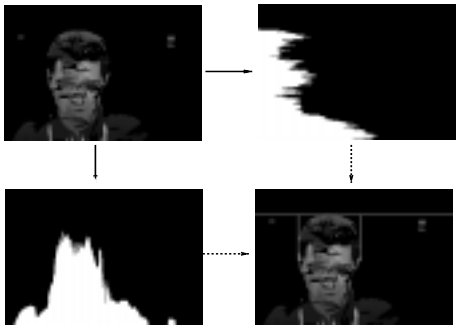

Figure 5: Speaker's outline binary histograms.

2.2.2 Eyes Tracking

The H, V, L and R parameters are derived from the position of the eyes. The eyes tracking algorithm must face four different challenges to be robust to:

scale changes: when the user moves forward or backward in front of the video camera

$2D$ **eyes rotations:** $2D$ means that the projected eye shape is rotating, for example when the user bends his head on his shoulder

$3D$ **eyes rotations:** $3D$ means that the projected shape changes non-linearly, when the speaker turns his head to the left or right hand side

face illumination changes: when the speaker moves in general

The last three challenges can be summed up by saying that the algorithm must cope with eye pattern changes (rotations or illumination). This section describes a two pass template matching algorithm that meets these requirements.

We addressed the problem of eye pattern modification by introducing dynamic templates: whenever the eyes are found in a frame, the template patterns are updated with the current eye images, and therefore adapt automatically to any change of eye shapes or illuminations. Parallely, the window W is used as an indication of the face scale to modify the template size. This way, the templates tend to keep the same amount of significant details and do not catch parasite details: if the initial templates contained the eyes and eyebrows, and the user moves away from the video camera, the templates will be updated with smaller image portions still displaying the eye and eyebrows, and will not include the user's nose or ears despite the scale change.

However, such dynamic templates are likely to deviate from the feature they are supposed to track, due to the fact that they are constantly updated: the new template pattern is extracted from the current frame relatively to the best match position. Let us imagine that the best match does not occur on the eye center (Figure 6). As a consequence, the center of the template will no longer be the center of an eye, and repeatedly, this process will make that the eyes will disappear from the updated templates. This problem is overcome by running a second template matching pass over the found eye area to find its center before updating the eye pattern. This second pass operates with a reference eye center pattern that is rescaled (not updated) to be robust to scale changes.

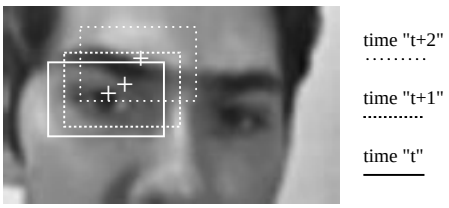

Figure 6: Dynamic templates deviation.

Figure 7 provides a few examples of eyes tracking, regardless to the speaker's pose. The first dynamic templates contained the user's eyes and eyebrows, and the recentring templates only the irises: the reader can notice that thanks to the eyebrows, the eyes can still be found even if they are closed.

3 Local Animation

Global animation just controls the position of the model in the $3D$ space without changing its facial expression. A local animation scheme has to be implemented in order to provide the user with an interface displaying the other participants facial expressions. Section 3.1 discusses the specificities of CYBERWARE models, and why they cannot directly perform local animations. The next section (3.2) deals with the different local animation strategies that are currently being investigated. And the last section (3.3) focuses on how it is possible to tie the local animation algorithms to video analysis techniques.

3.1 CYBERWARE Models

CYBERWARE models are produced by three–dimensional cylindrical scanners, and arrive in

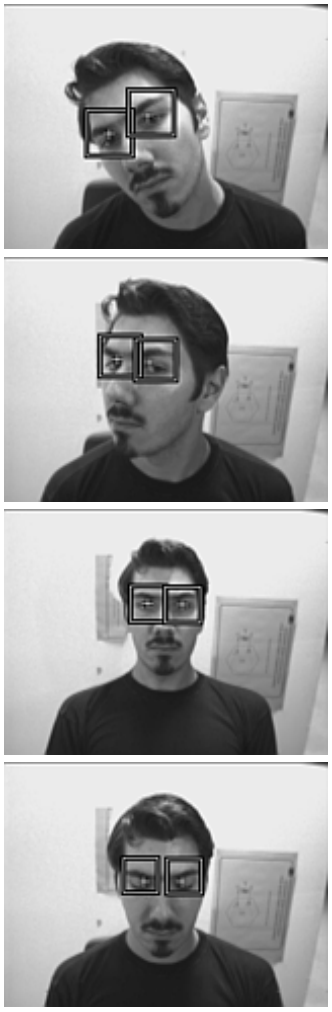

Figure 7: Dynamic template matching robustness to $2D$ rotation, $3D$ rotation, scale changes and closed eyes.

two files: the first one contains a set of $3D$ coordinates representing the scanned head geometry (i.e. a wire frame), and the second one holds texture information to be mapped to the geometrical coordinates.

These models are highly realistic, and contain precise information about the individual who was scanned. This can turn out to be a drawback, because each model is valid only for one person and a static facial expression, and lacks generality. Besides, the mesh model is unoptimized in terms of $3D$ nodes number, approximatively 1.4 million. The face $3D$ surface sampling was made along regular cylindrical coordinates, and as a result, there are as many points to represent smooth surfaces (like the cheeks) as to represent the sharpest ones (like the nose). It has neither anatomical knowledge (like bones and muscles under the skin to model human face deformation possibilities) nor a physical model (to be able to deform it along the time axis) [1]. In order to be capable of local deformations, further processings are under investigation to alter the plain $3D$ node set (see section 3.2.2).

3.2 Synthesis Strategies

Animating a CYBERWARE model is not common in the litterature, and we identifyed two different strategies to realize it: the first one consists in simulating animation by altering the texture attached to the wire frame, and the second one in manipulating the wire frame itself.

3.2.1 Textural Animation

We tried to animate the eyes first because we already know how to track them. The point of textural animation is to map different textures onto the wire frame at rendition time. We have already implemented routines in our synthesis software that dynamically switch between several eye textures (Figure 8) representing three distinct gaze directions, with almost no computation overload in our synthesis software. The middle texture is the original data produced by the CYBERWARE scanner, and the two others were pre–calculated using commercial image editing products.

Figure 8: Gaze control via texture modification.

3.2.2 Wire Frame Animation

Other features, like the eyebrows or wrinkles, can be animated either by texture modification, or by wire frame control.

We propose to transform the wire frame using Delingette's approach: it is adaptively remodelled to adapt the mesh complexity to the surface geometry, to get approximatively 1,400 nodes combined in 2800 triangles (starting from more than 1 million) [13] (Figure 9). This kind of active mesh has been successfully used in applications requiring realistic physical deformations, like surgery simulations [14], and supports also

rendering via simplex interpolations, which can lead to accurate expression wrinkles [15]. Further work has to be done to relate the processed wire frame to *Facial Action Coding System* (FACS) real–time software control units.

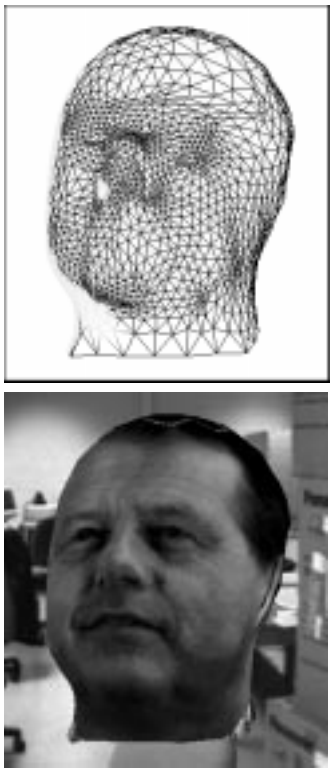

Figure 9: Wire frame and textured model.

3.2.3 Combination

Textural animation is both a low–cost and powerful alternative to wire frame animation, and a palliative technique to active mesh animation limitations: let us consider for instance the problem of animating the $3D$ model jaws. If the user opens his mouth, his teeth and tongue will become visible, but unfortunately, these features are not part of the CYBERWARE data (because the scanner digitalizes only the face surface, and not its inner parts). It will be necessary to resort to portions of real images to make them appear on the synthesis side to preserve the level of realism.

We then have a choice between using live portions of images, or pre–defined textures, depending on the bandwidth available for the system. On the one hand, using a dictionary of pre–defined textures just means referencing an index, and represents no bandwidth cost. On the other hand, sending live portions of image is more expensive, and its cost is difficult to evaluate, since it can take place at any time during the session. Some algorithms do exist, like the Graham–Schmidt orthonormalization process [7], which tend to minimize the bandwidth required to send a new texture portion considering the previously sent images, but still do not solve the problem of bandwith requirements predictability. Using live images might also be difficult to achieve because extracted $2D$ images have to be pasted into the CYBERWARE texture cylindrical coordinate system on order to be viewed under any point of view: the problems implied here are the $2D \rightarrow 2D_{cylindrical}$ transformation, and the photometric and lightning differences between the live and CYBERWARE textures (these considerations were not developped in [7]).

Building pre–defined texture dictionaries (possibly from typical teleconferencing sessions images) and referencing indexes seem to be the most efficient solution for now.

3.3 Prospective Studies

Once the synthesis policy is set in our demonstration software for each animated feature, we will need local analysis techniques to tie the animation to video input. It is important to realize that different facial features might require different animation procedures.

As far as textural animation is concerned, *Image Matching* seems to be the most appropriate technique to find the best pre–defined texture for the eyes (see section 3.2.1), or the user's forehead wrinkles, if they are simulated by "texture only". If necessary, the similarity measure can also be made illumination, scale and rotation invariant [16].

Animating the wire frame is a far more complex task. Although our CYBERWARE model cannot be used "as–is", it offers a nice property compared to other models: its precise mapping between texture and $3D$ geometry, allowing a high level of realism. We would like to take advantage of this realism with an analysis/synthesis cooperation. There is a very interesting possibility to create it off–line with *eigenfeatures*. Usually, eigenfeatures are used for their probabilistic learning capability: they give a compact representation of a rather complex space (like the space spanned by a training set of face images) by

finding the set of orthogonal vectors that represent the maximum energy subspaces. They have been widely used to recognize faces as a discriminative measure [17], and an attempt to make them classify poses can be found in [18].

The main difficulty to compute eigenfeatures is to set up a good training database. The features have to be well–scaled, well–centered, and the lightning conditions constant, which is far from being easy if out–of–the–lab images are exploited. This is where the CYBERWARE model can take its full modelling power: it can accurately be rendered under any pose, and with any facial expression defined by the synthesis software and its control parameters to produce a set of training images. In this case, not only will we get optimally trained eigenfeatures, but also they will come in an optimal basis to map the facial expressions found in images to the right control parameters to activate the wire frame.

And finally, in the case the speaker's face is not turned towards the video camera, it will also be quite acceptable to apply a realistic (but not real) deformation on the clone lips based on the audio signal analysis [19].

4 Summary and Conclusions

We have discussed the TRAIVI project and the associated video cloning issue, with more emphasis on virtual*ized* reality than virtual reality.

Video Cloning aims at providing people with $3D$ interfaces within a virtualized meeting environment. CYBERWARE models are used to ensure a high level of realism. We discussed the specificities of such models, and how they can be efficiently animated, regardless of the point of view under which they are rendered. Whereas the global animation scheme is mature, the local animation procedure is currently under development though some partial animation has been achieved for the gaze control.

We wrote a validation software that implements the presented algorithms between two SGI workstations. The analysis side runs on an SGI "Indy" at roughly 10 frames per second, and is mainly limited by the workstation video grabbing capability. The synthesis side runs on an SGI "High Impact", and renders the $3D$ model with 2800 triangles, texture mapping and background image restitution (producing synthetic images like figure 9).

References

[1] D. Terzopoulos and K. Waters. Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6), June 1993.

[2] A. Saulnier, M.-L. Viaud, and D. Geldreich. Real–time facial analysis and synthesis chain. In *International Workshop on Automatic Face— and Gesture— Recognition*, pages 86–91, Zurich, Switzerland, 1995.

[3] P. Ekman and W. V. Friesen. *Facial Action Coding System.* Consulting Psychologists Press, Palo Alto, California, 1977.

[4] I. A. Essa, T. Darrell, and A. Pentland. Tracking facial motion. In *IEEE Workshop on Nonrigid and Articulate Motion*, Austin, Texas, November 1994.

[5] Institut National de l'Audiovisuel. Televirtuality project: Cloning and real-time animation system. URL http://www.ina.fr/INA/Recherche/TV.

[6] M.J.T. Reinders, P.L.J. van Beek, B. Sankur, and J.C.A. van der Lubbe. Facial feature localization and adaptation of a generic face model for model–based coding. *Signal Processing: Image Communication*, 7:57–74, 1995.

[7] C. S. Choi, K. Aizawa, H. Harashima, and T. Takebe. Analysis and synthesis of facial image sequences in model–based image coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 4(3):257–275, June 1994.

[8] GMD's Digital Media Lab. TelePort: The Communication Wall. URL http://viswiz.gmd.de/DML/cwall/cwall.html.

[9] T. Kanade, P. J. Narayanan, and P. W. Rander. Virtual*ized* reality: Concepts and early results. In *IEEE Workshop on Representation of Visual Scenes*, Cambridge, Massachusetts, June 1995. In conjonction with ICCV'95.

[10] J.-L. Dugelay and K. Fintzel. Image reconstruction and interpolation in trinocular vision. In *IMAGE'COM*, pages 277–282, Bordeaux, France, May 1996.

[11] R. Boulic, T. Capin, Z. Huang, L. Moccozet, T. Molet, P. Kalra, B. Lintermann, N. Magnenat Thalmann, I. S. Pandzic, K. Saar, A. Schmitt, J. Shen, and D. Thalmann. The HUMANOID environment for interactive animation of multiple deformable human characters. In *Eurographics'95*, Maastricht, Netherlands, 1995.

[12] VIDAS — Video assisted audio coding and representation. URL http://www.uni-stuttgart.de/SONAH/Acts/AC057.html.

[13] H. Delingette. *Modélisation, Déformation et Reconnaissance d'Objets Tridimensionnelles à l'aide de Maillages Simplexes*. PhD thesis, Ecole Centrale de Paris, Châtenay–Malabry, France, 1994.

[14] H. Delingette, G. Subsol, S. Cotin, and J. Pignon. A craniofacial surgery simulation testbed. Research Report RR–2199, INRIA, 1994. URL http://www.inria.fr/rapports/sophia/RR–2199.html.

[15] M.-L. Viaud. *Animation Faciale avec Rides d'Expression, Vieillissement et Parole*. PhD thesis, Université de Paris XI–Orsay, Orsay, France, 1992.

[16] G. S. Cox and G. de Jager. Template matching with invariance. In *Proceedings of the Fourth South African Workshop on Pattern Recognition*, pages 152–156, 1993. URL http://dip1.ee.uct.ac.za/papers/cox93.html.

[17] A. Pentland, B. Moghaddam, and T. Starner. View–based and modular eigenspaces for face recognition. In *International Conference on Computer Vision and Pattern Recognition*, June 1994.

[18] T. Darell, B. Moghaddam, and A. Pentland. Active face tracking and pose estimation in an interactive room. Technical Report 356, M.I.T. Media Laboratory Perceptual Computing Group, 1996.

[19] R. R. Rao and T. Chen. Cross–modal prediction in audio–visual communication. In *International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, GA, May 1996.