EYE STATE TRACKING FOR FACE CLONING

A. C. Andrés del Valle and J.-L. Dugelay

Institut Eurécom. 2229, route des Crêtes. 06904 Sophia Antipolis - France {andres,dugelay@eurecom.fr}
http://www.eurecom.fr/~image

ABSTRACT

This article presents an efficient approach to eye movement estimation by combining color and energy based image analysis algorithms. The movement is first analyzed and then described in terms of action units. A temporal state diagram is used to control the behavior of the analysis over the time so the movement of the eye can be synthesized from the former description, after translating them into face animation parameters.

1. INTRODUCTION

Face animation has become a need for many multimedia applications where human interaction with virtual and augmented environments enhances the interface. It is also a solution for face image transmission in low bit rate communications, video-telephony, virtual teleconference, etc.

We can classify synthetic faces in two major groups: avatars and clones. Avatars are a rough or symbolic representation of the person. Clones are more realistic and their animation is by definition speaker-dependent. Face animation is mainly performed in two ways: by image or feature reconstruction (model-based) [1, 2] or by 3-D mesh movement synthesis [3, 4].

Our work is focussed on realistic 3-D head model movement synthesis from video sequences. The goal of our analysis-synthesis system is to synthesize realistic face expressions by generating Face Animation Parameters (FAP) to be applied on an MPEG-4 compliant head model. To deduce face movement from video frames [5] we first study the illumination conditions of the face in the sequences; this information will enable our algorithms to work under any lighting. Then, we estimate the pose of the face obtaining translation and rotation parameters. Our face tracking algorithm provides us with the global pose parameters and tracks the location of the most interesting features (eyes, eyebrows and mouth) allowing the definition of specific regions on the video frame (feature images). Finally, we apply some analysis techniques on the feature image to obtain face animation parameters. Many analysis schemes [1, 5] apply the same technique independently of the feature or expression they analyze. In our preliminary approach [5], we have used an image correlation approach that utilizes Principal Component Analysis to build the image databases. Storing the images of all possible lighting conditions, global pose situations and FAP combinations becomes unbearable for features like the mouth and the eyes where expressions can be quite complex. This approach has nevertheless proved to be very suitable for the eyebrow movement [6], but different techniques must be developed for other features. Regarding the eyes, we have adopted an analysis approach that uses the color components of the feature image, to be as light independent as possible, and that permits estimating the eye state (open, close, etc.).

Although eye movement detection has been widely investigated, among all, in eyesight tracking, we have not found in the literature an algorithm that completely exploits the physical characteristics of the eye. Many movement analysis techniques rely on optical flow or on model-based techniques [7], which are very sensitive to illumination changes and generally computing expensive.

In this paper we propose an efficient approach for real-time lighting-independent eye movement estimation. We consider applying a combination of color and energy analysis algorithms using the HSI (Hue, Saturation, Intensity) pixel components of the feature image. Then we interpret the results of the analysis in terms of some specific action units that we associate to the temporal states. Following a logical state diagram we relate our analysis results to the final parameters that describe the eye movement.

Section 2 contains the description of the HSI analysis algorithms. Section 3 explains the result interpretation through the temporal state diagram that obtains the action parameters. In Section 4 we describe our preliminary experiments and their associated results. We conclude by exposing our future perspectives in Section 5.

2. EYE ANALYSIS ALGORITHMS

Our analysis strategy decomposes the eye tracking actions in two categories: the open-close movement and the eyeball movement. To best exploit the physical characteristics of the eyes, a different algorithm analyzes each action.

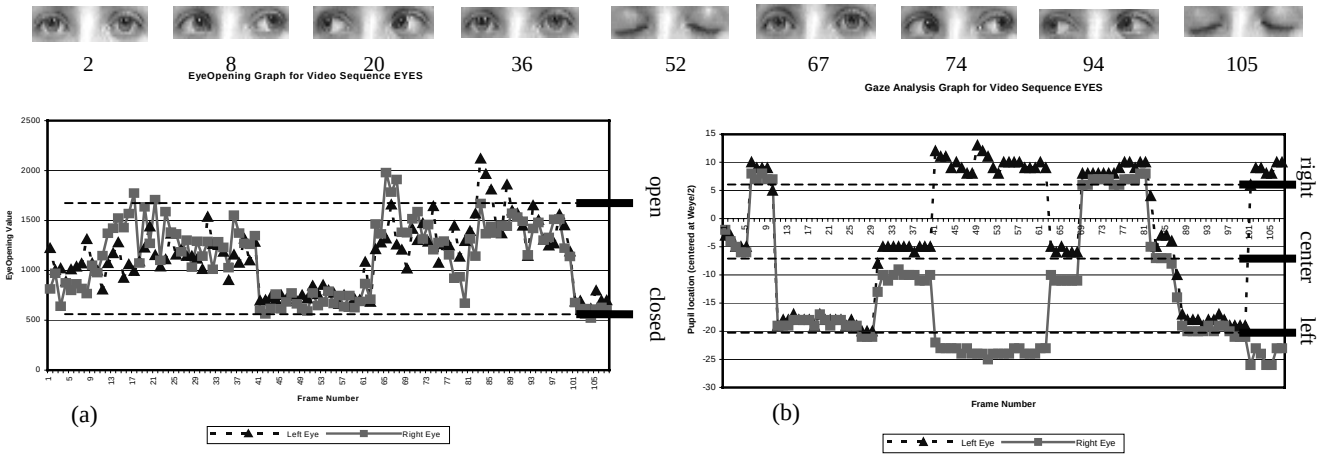

Figure 1. Analysis Graphs for a tested sequence. (a) *EyeOpening* (two quantization levels). (b) *GazeDetection* (three quantization levels).

2.1. 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 feature image (Eq. 1).

To measure the quantity of skin on the eye feature 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:

$$EyeOpening \propto \frac{1}{probSKIN} \qquad (1)$$

where

$$probSKIN \propto \sum_{h}\sum_{s} NUMpel_{h,s} \cdot PDF_{SKIN}(H=h, S=s)\,. \quad (2)$$

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 [8], 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.

2.2. Gaze detection

The gaze orientation is related to the position of the pupil on the feature region. The pupil can be defined as the lowest energy point of the eye. Thanks to our tracking mechanism [5] we can ensure that the analyzed area is always proportional to the eye and that the latter one is as centered as possible on it. Detecting the lowest energy point on the feature and relating it to its location on the image gives us the gaze orientation.

To determine the pupil position we first define the analysis zone. Then, we perform an exhaustive scan of the energy contained in this specific region, sweeping vertically and horizontally thorough the feature image whose dimensions are $Wfeature \times Hfeature$ (W for width and H for height). Each frame will have a different analysis zone depending on the eye size in that frame. We define α as the ratio *Wpupil / Weye*. α remains constant for all the frames and determines the analysis region.

The algorithm aims at obtaining the position *X, Y* of the analysis zone on the feature image so it corresponds to the minimum energy computed inside that region along the complete analysis of the feature image (Eq. 3). The coordinates *X* and *Y* give the pupil position from which we derive the eyesight orientation. The shape of the zone will depend on the kind of analysis we are interested in. For a vertical and horizontal movement detection, the zone we use is a square of area $\alpha \cdot Weye_i \times \alpha \cdot Weye_i$, where $Weye_i$ is the width of the eye at frame *i*:

$$X,Y = \min\left[\sum_{l=1}^{\alpha\cdot Weye}\sum_{m=1}^{\alpha\cdot Weye} I^2\left(x+l-Weye/2, y+m-Weye/2\right)\right] \quad (3)$$

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

For practical purposes, we 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 (3) is then transformed to only look for coordinate *X*, which indicates if the eye looks right or left:

$$X = \min\left[\sum_{l=1}^{\alpha\cdot Weye}\sum_{m=1}^{Hfeature} I^2\left(x+l-Weye/2, m\right)\right] \qquad (4)$$

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

The energy of a pixel is computed as the square of its intensity component, I. This algorithm relies on the intensity information of the image therefore is dependent of the lighting conditions. Its strength lies on its simplicity that allows a high control in possible misleading results. As we explain in Section 4, gradual changes in lighting do not influence the algorithm since the pupil mainly stays as the lowest energy point of the eye.

3. ANALYSIS INTERPRETATION FOR PARAMETRIC DESCRIPTION

To be able to synthesize eye movements, we have to parameterize the analysis result data so it can be interpreted. We also set a tight cooperation between the two previously described analysis techniques in a temporal state analysis, that allows us to double-check possible erroneous results from the algorithms. Next sub-sections develop the complete process for the state diagram specification.

3.1. Parameterization of eye movements

We define parameters to describe the eye movement to be synthesized. At this stage, 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 for movement synthesis. To test our procedure and to be able to evaluate its viability in real time, EO and HPO take the minimum possible values to describe the action. Table 1 depicts the actions and the corresponding values.

Table 1. Action unit description.

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

3.2. 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) (Fig. 1). 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 cannot count on a priori results, we must 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, $\Delta_{i,avg}$, is greater than a certain threshold (Th_{EO}) the eye has opened, if it is smaller the eye has closed, otherwise it remains as in the previous frame.

Table 2. The 36 combinatory results from the eye analysis.

S_L EO	S_L HPO	S_R EO	S_R HPO		S_L EO	S_L HPO	S_R EO	S_R HPO		S_L EO	S_L HPO	S_R EO	S_R HPO	
0	-1	0	-1	**A**	0	1	0	-1	S^1	1	0	0	-1	**A**
0	-1	0	0	**X**	0	1	0	0	**A**	1	0	0	0	**A**
0	-1	0	1	**X**	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	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	**X**	1	-1	0	0	**A**	1	1	0	0	**A**
0	0	0	1	**X**	1	-1	0	1	**A**	1	1	0	1	**A**
0	0	1	-1	**A**	1	-1	1	-1	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	S^4

S_LS_R = left&right analysis results; **X** = both results are erroneous; **A** = at lest one of them is correct & $S_L \neq S_R$; **S^i** = defined state (**S^1**=closed, **S^2**=look left, **S^3**= look center, **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). Fig. 2 shows on how we quantify the *X* value. If the analysis is performed when the eyes are closed, the minimum energy point usually lies on the side connected to the nose.

To have a more precise synthesis, both action units, EO and HPO, could take more intermediate values. In such a case the quantization of the space of analysis results would slightly change by adding more levels. The number of quantization levels must be chosen based on the capacity of synthesizing those details by clone animator. We must also evaluate if increasing the complexity of the quantization is appreciated when watching the real-time synthesis.

This first parameterization provides our preliminary analysis data. We can estimate if our guess has been correct by combining the information obtained from both eyes in a temporal state diagram.

3.3 Applying the temporal state diagram

Table 2 shows all the possible combinations of analysis results, S_LS_R. They can be completely erroneous for both eyes (X), 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 Fig. 3.

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

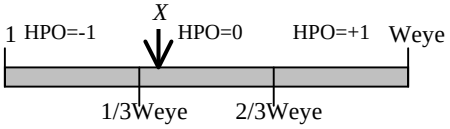

Figure 2. Quantization of HPO looking for left eye.

4. EXPERIMENTS AND RESULTS

We have used two sets of images for our experiments. One set with the recorded closed eyes of the person to obtain the PDF of HS values; the other where the eyes of the recorded face are 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 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 fairly encouraging (Fig. 1). 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 more performing 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. The main restriction of this approach is the assumption of having similar movement in both eyes, but it has better performance than analyzing each eye individually. The algorithm, as it is, avoids generating strange eye behavior but does not allow some little natural actions, for instance, only one eye blinking.

In terms of speed of performance at the moment, 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 judge opportune to adapt these algorithms to the YUV components (S:U, H:V, I:Y) since many graphics cards provide YUV output.

5. CONCLUSIONS AND FUTURE WORK

We have presented a two step technique that exploits the physical characteristics of the eye to analyze its movement. Combining a color and an energy analysis algorithm through a state diagram that shows left-right consistency, we define the basic actions of the eyes so we can then reproduce them.

Future work will involve the use of our MPEG-4 face animator to first, using highly realistic models to obtain the speaker-dependent parameters (Th_{EO}, PDF); second, adapt these algorithms, designed to optimally analyze a frontal position, to also work with different poses; and third, to relate our basic action units to the equivalent MPEG-4 FAPs. We also envisage refining the quantization so the synthesis of the movements can be more precise. As a final step we will merge these techniques with the already designed tracking pose system and other feature movement analysis procedures.

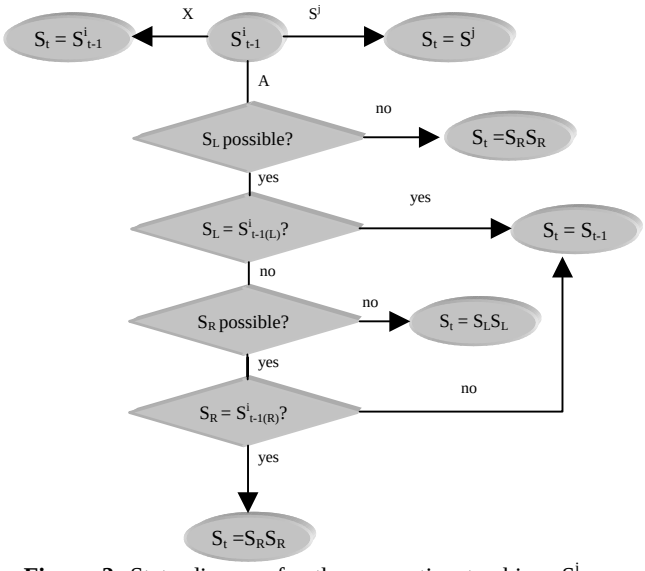

Figure 3. State diagram for the eye action tracking. $S^i_{t(R/L)}$ represents a determined state *i* at time *t* for either the right of left eye and S_t the final result. Check Table 2 for the state combinations.

6. ACKNOWLEDGEMENTS

Research partially supported by FRANCE TELECOM R&D.

7. REFERENCES

[1] E. Cosatto, G. Potamianos and H.P. Graf, "Audio-Visual Selection for the Synthesis of Photo-Realistic Talking-Heads", *ICME 2000*, New York City, NY, August 2000.

[2] L. Yin and A. Basu, "Partial Update of Active Textures for Efficient Expression Synthesis in Model-Based Coding", *ICME 2000*, New York City, NY, August 2000.

[3] G. Breton, C. Bouville and D. Pelé, "FaceEngine, un Moteur d'Animation Faciale 3D Destiné aux Applications Temps Réel", *France Telecom Symposium*, Poitiers, France, October 2000.

[4] F. Lavagetto and R. Pockaj, "The Facial Engine: Toward a High-Level Interface for Design of MPEG-4 Compliant Animated Faces", *IEEE Transactions on circuits and systems for video technology*, Vol. 9.

[5] S. Valente and J.-L. Dugelay, "Face Tracking and Realistic Animations for Telecommunicant Clones", *IEEE Multimedia Magazine*, pp. 34-43, February 2000.

[6] S. Valente, A. C. Andrés del Valle and J.-L. Dugelay. "Analysis and Reproduction of Facial Expressions for Realistic Communicating Clones", *The Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology*, Kluwer Academic Publishers. Vol. 9 Nos. 1/2, 2001.

[7] A.M. Al-Qayedi and A.F. Clark, "Constant-rate eye tracking and animation for model-based-coded video", *Proceedings of ICASSP 2000*, Istambul, Turkey, pp. 2353-2356, June 2000.

[8] H. Sahbi and N. Boujemaa, "From Coarse to Fine Skin and Face Detection", *Proceedings of ACM Multimedia 2000*, Los Angeles, CA, pp. 432-434, October 2000.