Eye Tracking and Animation for MPEG-4 Coding

Stefan Bernögger[‡*] Lijun Yin[†] Anup Basu[†] Axel Pinz[‡]

[†]Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2H1
{bernoegg,lijun,anup}@cs.ualberta.ca

[‡]Institute for Computer Graphics and Vision
Graz University of Technology
A-8010 Graz, Austria
{bernoegg,pinz}@icg.tu-graz.ac.at

Abstract

Accurate localization and tracking of facial features are crucial for developing high quality model-based coding (MPEG-4) systems. For teleconferencing applications at very low bit rates, it is necessary to track eye and lip movements accurately over time. These movements can be coded and transmitted to a remote site, where animation techniques can be used to synthesize facial movements on a model of a face. In this paper we describe simple heuristics which are effective in improving the results of well-known facial feature detection and tracking algorithms. Animation models are also presented, along with experimental results to demonstrate the system being developed. We focus our discussion only on the detection, tracking and modeling of eye movements.

1. Introduction

From the image analysis point of view, images can be considered as having structural features or objects such as contours and regions. These image features or objects have been exploited to encode images at very low bit rates. Research on this approach, known as model-based coding, which is related to both image analysis and computer graphics, has recently intensified. Up to now, most of the contributions to 3D model-based coding have focused on human facial images. Although a number of schemes for model-based coding have been proposed [3, 9], automatic facial feature detection and tracking along with facial expression analysis and synthesis still poses a big challenge to the problem of finding accurate features and their motion.

A variety of approaches have been proposed for detection of facial features. These include deformable template matching [4, 6, 14], Hough transforms, and color image processing [2, 13]. Matching deformable templates requires a fairly accurate initial localization of the template because the energy minimization process only finds a local minimum. Other problems are caused by using several energy terms and weighting factors during the different epochs of matching. Because of the definition of the energy terms in [14] the template also inclines to shrink. In this paper we overcome some of these difficulties by improving the initial localization process. We show that simple processing on color images coupled with Hough transform and deformable template matching can produce very accurate results.

Another important component in model-based coding is synthesizing facial movements and expression at a remote site, using the motion parameters detected on an actual face image and animation on a model of this face. To represent a facial expression, several approaches have been proposed relying on feature detection [1, 14], facial expression analysis [11, 12], and facial expression synthesis [7, 10]. However, little work has been done specifically on the eye expression synthesis. Because the eye is one of the most significant organs contributing to a vivid face expression, subtle changes of the eye movement can result in a different expression. Therefore accurate eye expression analysis and synthesis are necessary.

In this paper, we present an approach to synthesize the eye movement by using the extracted eye features to compute the deformation of the eyes of the 3D model. After creating an individualized 3D face model, we map this model to the first frame of the face sequence. Based on the extracted eye features we apply the deformation parameters to the 3D model to synthesize the eye movement in successive frames.

The remainder of this paper is organized as follows: In Section 2, we describe the approach proposed for accurate eye feature detection and tracking. In Section 3, animation techniques for presenting eye movement are described. Ex-

*Supported by a Kurt Gödel scholarship from the Austrian government Stefan Bernögger was on leave at the Department of Computing Science, University of Alberta, Edmonton, in 1997/98.

perimental results are shown in Section 4. Finally, concluding remarks are given in Section 5.

2. Robust detection and tracking of eye features

Our approach to detecting the eyes is similar to [4] in that it uses Hough transform and deformable template matching, however, it also exploits color information to extract the eyes accurately. The algorithm can be outlined as follows:

- Determine two coarse regions of interest for the eyes.
- Search the iris of the eyes using a gradient based Hough transform.
- Determine a fine region of interest for extracting the boundaries of the eyes.
- Using color information to get an initial approximation for the eye lids.
- Localize the eye lids using deformable templates.

After detecting the face region two coarse regions of interest in the upper left and upper right half of an image can be defined to detect the eyes. Also a coarse range for the size of the eyes can be derived.

Since the iris is the most significant feature of the eye and has a simple circular shape, it is detected first by using a gradient-based Hough transform for circles [5, 8]. The

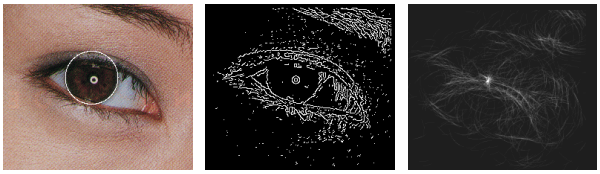

Figure 1. Extraction of circle (image and result, edge image, Hough space).

necessary information position, magnitude and direction of significant edges is extracted by convolving the intensity image with a Sobel kernel, and performing a non-maximum suppression by removing all edges with a magnitude lower than 30% of the maximum magnitude. Figure 1 shows the robustness of the Hough transform. In spite of the large amount of edgels which are not produced by the contour of the iris the Hough space shows a significant maximum at the exact position of the center of the iris.

After extracting the circles the deformable templates for the eye lids have to be initialized. The model, along with all the parameters used is shown in Figure 2; with the parameters being set as follows:

$$a = 1.5r_{iris} \qquad c = 0.5r_{iris} \qquad b = 2.2r_{iris} \qquad (1)$$

where r_{iris} is the radius of the extracted circle. The orientation α is determined by the center points of the two circles. The initialized deformable template is also shown in Figure 2.

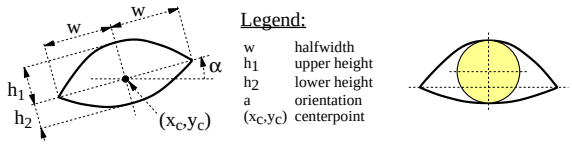

Figure 2. Deformable Template (model, initialization).

For the matching of the deformable template two different types of image information are used to create potential fields, one in each epoch. The image information extracted from a typical eye is shown in Figure 3.

Figure 3. Image fields used for computing the potential energy (image, saturation, edge).

The first step in localizing the eye lids is to approximate the position of the deformable template relative to the iris. This is done by minimizing the following energy (E_{sat}) which is similar to the valley energy in [14]:

$$E_{sat} = -\frac{1}{|A_w|}\int_{A_w} \Phi_{sat}(\vec{x})dA \qquad (2)$$

A_w is the area inside the parabolas but not inside the circle of the iris and $\Phi_{sat}(\vec{x})$ is the inverted saturation value of the color image. Since only the location (not the size) is changed this method does not have the shrinking effect.

The next step is to approximate the parameters h_1 and h_2. Especially the parameter h_1 is important because of the larger movement of the upper eye lid. In order to estimate the parameters, two regions of interest on both sides of the iris are defined (see Figure 4). The parameters of these regions are set as follows:

$$dx = 5 \qquad dy = 3r_{iris} \qquad k = r_{iris} + 5 \qquad (3)$$

Depending on the position of the iris inside the deformable template only the left or the right region of interest is used for the further computation.

By using the horizontal integral projection ([1]), and by detecting the two most significant opposite gradients in the projection, the position of a point on the upper and lower

eye lid can be detected (see Figure 4). The parameters h_1 and h_2 of the template are updated as follows:

$$h_1 = h_1 \frac{|y_{up} - y_c|}{h_1^*} \qquad h_2 = h_2 \frac{|y_{low} - y_c|}{h_2^*} \tag{4}$$

y_{up} and y_{low} are the y-coordinates of the detected points inside the region of interest of the upper and lower eye lid. h_1^* and h_2^* are the height of the actual parabolas inside the region of interest.

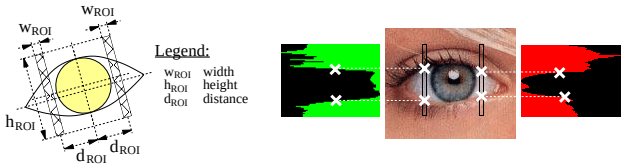

Figure 4. Integral projections beside iris (model and example).

The last step is to match the deformable template accurately to the eye lids by minimizing the following energy (E_{edge}):

$$E_{edge} = -\frac{1}{|B_w|} \int_{B_w} \Phi_{edge}(\vec{x}) ds \tag{5}$$

B_w is the boundary of the parabolas and $\Phi_{edge}(\vec{x})$ is the edge magnitude. During this minimization every parameter of the deformable template (location, orientation, height, width) can be changed.

The tracking of eye features is similar to detection of eye features with the following differences:

- The region of interest as well as the possible size and therefore the Hough space for the extraction of the iris can be restricted by using the position and the size of the extracted eye in the previous frame.
- Instead of using the deformable template shown in Figure 2, the matched template of the previous frame is used for the initialization of the new template.

3. Animation of eye movements

The eye detection and tracking algorithms extract the contours of the iris and eye lids in each frame of the image sequence. These eye features are used to synthesize the real motions of the eye on a 3D facial model. The main procedure is described as follows:

- Based on the 3D eye model which is part of our existing 3D facial model, 11 feature vertices are defined for each eye as shown in Figure 5. Once the eye lid contour and the iris are detected in the image sequence, the corresponding feature points on the template can be also determined simply by computing the points on the boundary of the parabolas and by using the center of the circle.

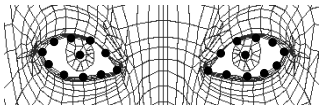

Figure 5. Defined eye feature vertices.

- For iris animation: After the iris is detected in the image sequence, the center of the iris in the eye model can be matched with the iris center in the eye image. The size of the iris model can be adjusted to the circle of the iris template.
- For eye lid animation: After the eye lids are detected in the image sequence, the feature vertices on the eye model can be fitted to the feature points on the eye image. The displacements of the feature vertices and feature points consist of a set of motion vectors (we call them *feature motion vectors* (FMV)). To deform and animate the eye movement, the set of motion vectors of non-feature vertices has to be found as well. They can be derived by interpolating the feature motion vectors which are around the non-feature vertices. Actually, in the interpolation procedure, the newly obtained motion vector of the non-feature vertex is also put into the set of FMV. Therefore the size of the FMV is increased dynamically. The motion vector of a non-feature vertex on the eye model (denoted as **v**) can be derived by the following equation:

$$\mathbf{v} = \sum_{n=1}^{N} (w(d_n) \cdot \mathbf{v}_n) \tag{6}$$

where d_n is the distance between the non-feature vertex (v) and the feature vertex v_n, $n = 1, 2, \ldots, N$; $(d_1, \ldots, d_N)$ are arranged in an increasing order, and $w(d_n)$ is a weight function which has a large output value for a small input of d_n:

$$w(d_n) = \frac{d_{n'}}{\sum_{n=1}^{N} d_n} \qquad \text{where} \qquad n' = (N+1) - n \tag{7}$$

Based on these vertices, a deformed eye model can be built for each frame. The resulting eye model will show us the synthesized eye movement (see also Figure 6).

- Finally, the animated eye model can be texture-mapped by the original image (*i.e. first frame*) to synthesize the real eye expression completely.

4. Experimental results

To evaluate the algorithms developed, experiments with various color images of different eyes and eye sequences were made. The preliminary experiments have shown that the Hough transform, which is the first and therefore one of the most important steps, is very robust against noise as well as edges which are not produced by the contour of the iris. The results on tracking and animating of the iris and the eye lids in one image sequence are shown in Figure 6.

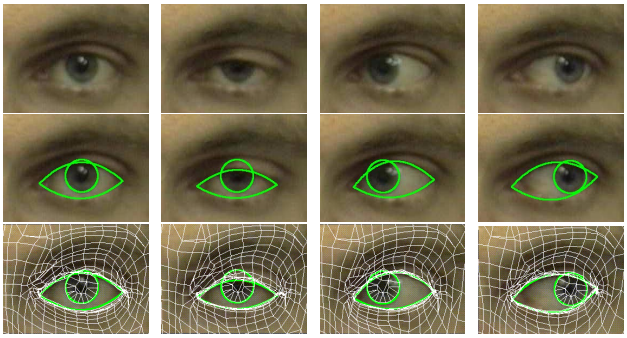

Figure 6. Sequence (Frame 1, 24, 68, 169) showing eye movement. Original images (top); extracted eye template (middle); matched and animated 3D model (bottom).

Eye movements are synthesized as follows: First, eye movements are modeled as deformations of the individualized wire frame model (see Figure 7)(top). Then, the wire frame models in the successive frames are texture-mapped by using the first frame of the sequence (see Figure 7)(bottom).

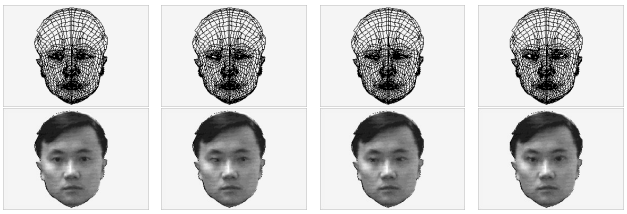

Figure 7. Animated wire frame model (top) and texture-mapped model (bottom) for different eye expressions. From left to right: individual model and original image (first frame), deformed model and synthesized images looking left, right, and straight respectively.

From the preliminary experimental results, we can see that the algorithms proposed here behaves well for synthesizing eye movements. For the experiments in this paper, a high resolution facial model with 3118 patches is used instead of the low resolution model used in [11]. However, the time complexity of modeling, analysis and synthesis increase significantly as a result.

5. Conclusion

In this paper we discussed robust methods for detecting and tracking eye movements, a strategy for eye movement animation, and resulting applications in model-based low bit-rate coding.

In future reports we will discuss extensions of the method to robust detection of lip movements and network strategies for real-time demonstration of the system.

References

[1] R. Brunelli and T. Poggio. Face Recognition: Features versus Templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, Oct. 1993.

[2] T. Chang and T. Huang. Facial Feature Extraction from Color Images. In *Proc. 12^{th} IAPR Int. Conf. on Pattern Recognition, Jerusalem, Israel*, volume II, pages 39–43. IEEE Computer Society Press, October 9-13 1994.

[3] C. Choi and T. Takebe. Analysis and Synthesis of Facial Image Sequences in Model-Based Image Coding. *IEEE Transactions on Video Technology*, 4:257–275, June 1994.

[4] G. Chow and X. Li. Towards a System for Automatic Facial Feature Detection. *Pattern Recognition*, 26(12):1739–1755, Dec. 1993.

[5] E. Davies. A Modified Hough Scheme for General Circle Location. *Pattern Recognition*, 7:37–43, Jan. 1988.

[6] J. Deng and F. Lai. Region-Based Template Deformation and Masking for Eye-Feature Extraction and Description. *Pattern Recognition*, 30(3):403–419, Mar. 1997.

[7] P. Ekman and W. Friesen. *Facial Action Coding System*. New York: Consulting Psychologists Press, 1977.

[8] P. Kierkegaard. A Method for Detection of Circular Arcs Based on the Hough Transform. *Machine Vision and Applications*, 5:249–263, 1992.

[9] D. Pearson. Developments in Model-Based Video Coding. *Proc. of the IEEE*, 83(6):892–906, June 1995.

[10] S. Platt and N. Badler. Animating Facial Expressions. *Computer Graphics*, 13(3):245–252, 1981.

[11] L. Tang and T. Huang. Quantifying Facial Expressions: Smiles. In *Proc. of the Int. Workshop on Coding Techniques for Very Low Bit-rate Video*, pages 345–350, 1994.

[12] 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):569–579, June 1993.

[13] H. Wu, T. Yokoyama, D. Pramadihanto, and M. Yachida. Face and Facial Feature Extraction from Color Image. In *Proc. 2^{nd} IEEE Int. Conf. on Automatic Face and Gesture Recognition, Killington, Vermont, USA*, pages 345–350, October 14-16 1996.

[14] A. Yuille, P. Hallinan, and D. Cohen. Feature Extraction from Faces Using Deformable Templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.