Proceedings, International Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, June 1995.

Automatic location tracking of faces and facial features in video sequences

Arnaud Jacquin and Alexandros Eleftheriadis
AT&T Bell Laboratories
Signal Processing Research Department
600 Mountain Avenue
Murray Hill, NJ 07974, USA
E-mail: arnaud@research.att.com

Abstract

The work reported in this paper addresses the issue of automatically tracking the faces and facial features of persons in head-and-shoulders video sequences. We propose two totally automatic algorithms which respectively perform the detection of head outlines and identify rectangular "eyes-nose-mouth" regions, both from downsampled binary thresholded edge images. Unlike ones that have been proposed recently, *a priori* assumptions regarding the nature and content of the sequences to code are minimal for our techniques, and the algorithms operate accurately and robustly, even in cases of significant head rotation or partial occlusion by moving objects.

1 Introduction

The motivation for this work was to investigate the feasability of detecting and tracking specific moving objects known *a priori* to be present in a video sequence, and to enable a low bit rate video coding system to use this information in order to discriminatively encode different areas in "head-and-shoulders" video sequences—an idea which has been recently proposed in [11, 17, 13]. The encoder would, for example:

- Encode facial features (such as eyes, mouth, nose, etc.) very accurately.
- Encode less accurately the rest of the picture, be it moving or still.

This requires that the encoder first models and track face locations, then exploit this information to achieve *model-assisted coding* [5, 6]. The location detection algorithm should be of fairly low complexity; in addition, if transmission of the model parameters is required, the overhead bit rate should be minimized.

The detection of head outlines as well as outlines of persons (silhouettes) in still images has been the object of active and recent research in computer vision [7, 3, 8, 9]. The task of detecting and tracking head outlines in a sequence of images is facilitated by both the fact that people's head outlines are consistently roughly elliptical, including when the persons appear in a profile, and by the temporal correlation from frame to frame. Previous work [5, 6] proposed an automatic algorithm for head outline location tracking, as well as the design of a model-assisted dynamic bit allocation strategy with object-selective quantization, in the context of a 3D subband based video codec operating at the total bit rate of 128 kbps. In this paper, we propose automatic algorithms for the detection and tracking of both head outlines and rectangular "eyes-nose-mouth" regions. The former algorithm models face contours as ellipses. The latter exploits the symmetry with respect to a slanted facial axis, which is inherent to a human face appearing in a 2D projection provided that the rotation of the head is slight.

In Section 2, we describe the models adopted for the representation of face location information, and the generation of edge data used as input to the tracking algorithms. In Sections 3 and 4, we describe two automatic low–complexity algorithms respectively for the tracking of head outlines and for the tracking of eyes-nose-mouth regions. The algorithms operate under minimal assumptions regarding sequence content, and belong to a broad class of pattern-matching algorithms used for object detection [16, 15]. Sample detection results and statistics are presented in Section 5.

2 Location models and extraction of edge data

2.1 Face and feature location models

The model we adopted in order to represent the location of a face was simply that of an ellipse $\mathcal{E}$, as shown in Figure 1, characterized by a center (x_0, y_0), the lengths of its minor and major axes A, B, and a "tilt" angle θ_0. Although the upper (hair) and lower (chin) areas in actual face outlines can have quite different curvatures, ellipses provide a good trade-off between model accuracy and parametric simplicitly.

Equivalently, an ellipse of arbitrary size and "tilt" can also be represented by a quadratic, non-parametric equation (implicit form) [2] of the form:

$$ax^2+2bxy+cy^2+2dx+2ey+f=0, \quad b^2-ac<0, \quad (1)$$

where the negative value of the discriminant $D = b^2 - ac$ is a necessary condition, as other values are associated with different quadratic curves.

Since an elliptical head outline can in some cases provide only a rough estimate of the face location, we have chosen to refine this elliptical location model by identifying a rectangular region $\mathcal{W}$ inside the ellipse, which tightly captures the eyes, nose, and mouth of the person in the scene. This location model of an "eyes-nose-mouth" region is similar to the one proposed by Lavagetto et al. [10, 4], with a difference introduced by allowing a slant of its vertical axis with respect to the image vertical, as shown in Figure 2. This additional degree of freedom ensures that the detection will be robust in the (very frequent) case of slight head motion (see Sections 4, 5). The upper third of the window, denoted $\mathcal{W}_u$, is further identified to contain the eyes and eyebrows—two most reliably symmetric features in a human face. The window $\mathcal{W}$ is entirely characterized by a center (x_1, y_1), width w, height h, and slant angle θ_1.

2.2 Generation of binary edge data

The binary input data to the automatic face location algorithm of the next section are obtained at the encoder through a preprocessing stage depicted in Figure 3, consisting of the following cascade of operations:

1. Temporal downsampling of the input luminance video signal, consistent with the frame rate of the input signal to the video codec.
2. Low–pass filtering of input video frames of size 360×240 with a separable filter with cut-off frequency at $\pi/8$, followed by decimation by a factor 8 in both horizontal and vertical dimensions, producing low-pass images of size 45×30.
3. Edge detection on these images. The Sobel operator [14], represented in matrix form by horizontal and vertical operators as:

$$\delta_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad (2)$$

$$\delta_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \quad (3)$$

   is used to compute the two components of an image gradient. A gradient magnitude image is then obtained by computing the magnitude of the gradient at each pixel.
4. Tresholding of the gradient magnitude images to generate binary edge data.

The binary input data to the automatic location algorithm for the eyes–nose–mouth region is generated in a similar way, albeit on images which are only downsampled by a factor two, in order not to lose the features of interest in the downsampling, namely eye, nose, and mouth edge data.

3 Automatic detection and tracking of head outlines

The algorithm detects and traces the outline of a face location geometrically modeled as an ellipse, using as preprocessed input data binary thresholded gradient magnitude images of size 45×30. Our face location detection algorithm was designed to locate both oval shapes (i.e. "filled") as well as oval contours partially occluded by data. The algorithm is organized in a hierarchical three-step procedure: coarse scanning, fine scanning, and ellipse fitting. A final step consists of selecting the most likely among multiple candidates. This decomposition of the recognition and detection task in three steps, along with the small input image size, make the algorithm attractive for its low computational complexity; exhaustive searches of large pools of candidates were thereby avoided. The different steps are described below, and are illustrated in Figure 4.

Step 1: Coarse Scanning
The input signal is segmented into blocks of size 5×5. Each block is *marked* if at least one of the pixels it contains is non-zero. The block array is then scanned in a left-to-right, top-to-bottom fashion, searching for contiguous runs of marked blocks. One such run is shown in the small circle, in Figure 4.a. For each such run the following two steps are performed.

Step 2: Fine Scanning
Figure 4.b shows the two circled blocks of the run of Figure 4.a, appropriately magnified. The algorithm scans the pixels contained in the blocks of a run, again in a left-to-right, top-to-bottom fashion. Here, however, the algorithm is not interested in contiguous runs of pixels, but rather in the first line that contains non-zero pixels. The first and last non–zero pixels of that line, with coordinates (X_{start}, Y), (X_{end}, Y), define a *horizontal scanning region*.

The first two steps of the algorithm act as a horizontal edge-merging filter. The size of the block directly relates to the maximum allowable distance between merged edges. It also has a direct effect on the speed of the algorithm, which is favored by large block sizes. The purpose of these two steps is to identify candidate positions for the top of the head, where the edge data corresponding to the head outline is generally unencumbered by data corresponding to other objects. At the end of the second step, the algorithm has identified a horizontal segment which potentially contains the top of the head.

Step 3: Ellipse Fitting/Data Reduction
In this third step, illustrated in Figure 4.c, the algorithm scans the line segment defined by (X_{start}, Y), (X_{end}, Y). At each point of the segment ellipses of various sizes and aspect ratios are tried-out for fitness, with the top–most point of the ellipse always located on the horizontal scanning segment. Good matches are entered as entries in a list. After the search is completed on the segment, the algorithm continues at the point where it left off in Step 1. Only ellipses with "zero tilt" ($\theta_0 = 0$) were considered here. The primary reason for imposing this restriction is that we could trade-off an extra degree of freedom (and hence algorithm simplicity) by extending the search range for the aspect ratio[1]. Another reason is that the orientation of the facial axis (corresponding to the major axis of the ellipse model) can be much more reliably obtained by exploitng considerations of facial symmetry, as described in Section 4.

The fitness of any given ellipse to the data is determined by computing normalized weighted average intensities I_i and I_e of the binary pixel data on

[1]Typical face outlines have been found to have aspect ratios in the range of $(1.4, 1.6)$ [9]. Moreover, the face tilt has been found to be in the range $(-30^o, +30^o)$; a significant constraint due to the human anatomy. Within these ranges for θ and r, a tilted ellipse can be reasonably covered by a non–tilted one, albeit with a smaller aspect ratio (in the range $(1.0, 1.4)$).

the ellipse *contour* and *border* respectively. Although the contour of an ellipse is well-defined by its non-parametric form, the rasterization (spatial sampling) of image data necessitates the mapping of the continuous curve to actual image pixels. This is also true for the ellipse border. These discretized curves are defined as follows. Let $I_{\mathcal{E}}(i,j)$ be the index function for the set of points that are inside or on the ellipse $\mathcal{E}$. In other words,

$$I_{\mathcal{E}}(i,j) = \begin{cases} 1 & \text{if } (i,j) \text{ is inside or on } \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

A pixel is classified as being on the ellipse *contour* if it is inside (or on) the ellipse, and at least one of the pixels in its $(2L+1)\times(2L+1)$ neighborhood is not, i.e.:

$$\begin{aligned} (i,j) \in \mathcal{C}_i \Longleftrightarrow I_{\mathcal{E}}(i,j) &= 1, \quad \text{and} \\ \sum_{k=i-L}^{i+L} \sum_{l=j-L}^{j+L} I_{\mathcal{E}}(k,l) &< (2L+1)^2. \end{aligned} \tag{5}$$

Similarly, a pixel is classified as being on the ellipse *border* if it is outside the ellipse, and at least one of the pixels in its $(2L+1)\times(2L+1)$ neighborhood is inside the ellipse. The parameter L defines the desired *thickness* of the ellipse contour and border, and is a tunable design parameter.

Given the above definitions for contour and border pixels, the normalized weighted average intensities I_e and I_i are defined as follows:

$$I_i = \frac{1}{|\mathcal{C}_i|} \sum_{(m,n)\in\mathcal{C}_i} w_{m,n}\; p(m,n), \tag{6}$$

where $p(m,n)$ are the (binary) image data, $|\mathcal{C}_i|$ is the cardinality of $\mathcal{C}_i$, and $w_{m,n}$ are weighting factors introduced to enhance the contribution of the data in the upper quarter of the ellipse (see Fig. 2)—the most reliable region for fitting, i.e.

$$w_{m,n} = \begin{cases} w > 1 & \text{if } (i,j) \in \mathcal{Q}_u \\ 1 & \text{if } (i,j) \text{ not in } \mathcal{Q}_u, \end{cases}$$

In our experiments, a weight $w = 1.5$ was used, for which consistenly reliable results were obtained. Similarly, we define:

$$I_e = \frac{1}{|\mathcal{C}_e|} \sum_{(m,n)\in\mathcal{C}_e} p(m,n). \tag{7}$$

The normalization with respect to the "length" of the ellipse contour and border is necessary, in order to accomodate ellipses of different sizes.

An ellipse will fit ellipse–shaped data well whenever the value of I_i is high (close to the maximum value $I_{max} = \frac{3+w}{4}$), and that of I_e is low (close to zero). In order to translate this joint maximization-minimization problem to the maximization of a single quantity, we define a model-fitting ratio R as:

$$R = \frac{1+I_i}{1+I_e}. \tag{8}$$

The higher the value of R, the better the fit of the candidate ellipse to the head outline[2].

[2] In the hypothetical situation of perfectly ellipse-shaped data, the best-fitting ellipse aligned with the data would correspond to $I_i = 1$, $I_e = 0$, and $R = 2$.

In order to filter out false candidates, only ellipses which satisfy:

$$I_i > I_{i_{\min}} \text{ and } I_e < I_{e_{\max}}, \tag{9}$$

are considered, where $I_{i_{\min}}$ and $I_{e_{\max}}$ are tunable design parameters. Their use is necessitated by the fact that R is mostly sensitive to the relative values of I_i and I_e, and much less to their absolute values.

The above three–step procedure will in general yield more than one ellipse with a good fit. If there is a need to select a *single* final one (e.g. when it is known that the sequence only includes one person), then an elimination process has to be performed. This process uses two "confidence thresholds" $\Delta R_{\min}$ and $\Delta I_{e_{\min}}$. If the value of R for the best–fitting ellipse is higher from the second best by more than $\Delta R_{\min}$, then the first ellipse is selected. If not, then if the border intensity difference between the two ellipses is higher than $\Delta I_{e_{\min}}$, the ellipse with the smallest I_e is selected. If the border intensity difference is smaller than that (which rarely occurs in practice), then the original best candidate (the one with the maximum R) is selected.

4 Detection of eyes–nose–mouth regions

The elliptical face location model described above can be refined by including a segmentation of the elliptical region into a rectangular window and its complement. We require that the window be positioned so that it tightly captures the region of the face corresponding to eyes and mouth; the features of interest. Our identification of eyes/mouth regions follows the procedure described by Lavagetto et al. in [10, 4], and first proposed by Badiqué [1], and extends it to include tracking in cases where: i) the subject does not directly face the camera, ii) the subject has facial hair and/or wears eyeglasses, and iii) the subject is not caucasian. The algorithm is based on exploiting the typical symmetry of facial features with respect to a longitudinal axis going through the nose and accross the mouth. Our algorithm allows this symmetry axis to be *slanted* with respect to the vertical axis of the image, thereby ensuring considerably more robustness in the detection of an eyes-nose-mouth region. In particular, detection of this region is still possible when the subject does not look directly at the camera; a very common occurence in a video teleconferencing situation. The algorithm operates in two steps:

Step 1: Definition of search region

The center (x_0, y_0) of the elliptical face location model is used to get estimates for the positioning of the eyes-nose-mouth window. The search region for the center of this window is defined as a square region of size $S \times S$ (S was equal to 12 in our experiments). The window itself was chosen to be of fixed size $w \times h$, defined relatively to the minor and major axes of the face location model.

Step 2: Scanning of search region

For each candidate position (x_k, y_k) of the window center in the search region, a *symmetry functional* with respect to the facial axis is computed, where this axis can be rotated by discrete angle values around the center of the window. In our experiments, the slant angle θ_k could take any of the discrete values $-10°$, $-5°$, $0°$, $5°$, $10°$. Let $S(m,n)$ denote the point which is symmetric to (m,n) with respect to the axis $\mathcal{D}((x_k,y_k),\theta_k)$. The symmetry functional is computed as follows:

$$S(x_k,y_k,\theta_k) = \frac{1}{A(\mathcal{R})}\left(\sum_{(m,n)\in\mathcal{R}\cap\mathcal{W}_u} w a_{m,n} + \sum_{(m,n)\in\mathcal{R}\setminus\mathcal{W}_u} a_{m,n} \right) \quad (10)$$

where $A(\mathcal{R})$ denotes the cardinality (area in pixels) of the trapezoid region $\mathcal{R}$ depicted in Figure 2, $\mathcal{R}\setminus\mathcal{W}_u$ denotes the set difference of $\mathcal{R}$ and $\mathcal{W}_u$, $a_{m,n}$ is the function defined by:

$$a_{m,n} = \begin{cases} 1 & \text{if } p(m,n) = p(S(m,n)) = 1 \\ \frac{1}{2} & \text{if } p(m,n) = p(S(m,n)) = 0 \\ 0 & \text{otherwise,} \end{cases}$$

and w is a weighting factor greater than one. This weighting factor w ensures that the edge data in $\mathcal{W}_u$ which is symmetric with respect to the axis, significantly contributes to the functional. The segmentation of the rectangular window into the regions $\mathcal{W}_u$ and $\mathcal{R}$ ensures that only data corresponding roughly to the eyes, nose and mouth be taken into consideration in the positioning of the window, and that this positioning is mostly enforced by "eye data", which we have found to be the most reliably quantitatively symmetric region.

As in the ellipse detection case, false candidates defined as windows for which the density of data points in the upper third rectangle is below a minimum density D_{min}, are filtered out.

5 Experimental detection results

The output of sample test runs of the automatic face location detection algorithm on sequences referred to as "jelena," "roberto," and "jim," is shown in Figures 5, and 6. In these figures, the images on the left show binary edge images, magnified by a factor two in both dimensions, with best-fitting eyes-nose-mouth regions overlayed in gray.

The head location tracking algorithm performs robustly, even in difficult situations such as partial occlusion of the person's head/face by a hand–held moving object. The tracking of eyes-nose-mouth regions is also performed robustly as illustrated by the examples given, where the two men in the images have facial hair and/or wear eyeglasses. The rate of "correct tracking" of the eyes-nose-mouth area—defined as the accurate capture of eyes-nose-mouth region *and* correct estimation of the facial nose-mouth axis—is approximately 95%, on average over more than 80 seconds of video data. It was of particular interest to observe that constraining the rectangular window to be aligned with the image frame resulted in an average correct tracking rate of only 80%.

5.1 Conclusion and future work

We proposed two algorithm for the tracking of a facial area in head-and-shoulders video sequences. Head location detection is based on a low-complexity hierarchical algorithm that models the head as an ellipse, and utilizes physical structure information to robustly identify head contours, even in cases of severe occlusion. Face location is then estimated starting from the head position information, and exploiting the natural symmetry that can be found in the facial features with respect to a vertical axis.

Unlike algorithms that have been proposed recently, ours do not assume *a priori* restrictions regarding the nature and content of the head-and-shoulders sequences to code. The two algorithms operate accurately and robustly respectively: i) in cases of significant head rotation and/or partial occlusion by moving objects, ii) in cases where the person in the image has facial hair and/or wears eyeglasses.

The tracking accuracy for eyes-nose-mouth regions is high—more than 95% on average—which gives hope for the future design of algorithms for the robust tracking of individual facial features, such as eyes and mouth corners. The tracking algorithms could also be tailored to different applications, i.e. to track any object with a simple geometric outline known *a priori* to be present in the scene, and also be extended to operate on multiple simultaneous objects.

References

[1] E. Badiqué, "Knowledge–based facial area recognition and improved coding in a CCITT–compatible low bit rate video codec," *Proc. PCS*, 1990.

[2] R. C. Beach, "An Introduction to the Curves and Surfaces of Computer-Aided Design," Van Nostrand Reinhold, New York, 1991.

[3] I. Craw, H. Ellis, J.R. Lishman, "Automatic extraction of face features," *Pattern Recognition Letters*, vol. 5, no. 2, February 1987.

[4] S. Curinga, A. Grattarola, and F. Lavagetto, "Synthesis and animation of human faces: artificial reality in interpersonal video communication," in *Modeling in Computer Graphics*, B. Falcidieno, T.L. Kunii (Eds), Springer Verlag, 1993.

[5] A. Eleftheriadis, A. Jacquin, "Model–assisted coding of video teleconferencing sequences at low bit rates," *Proc. ISCAS '94*, May–June 1994.

[6] A. Eleftheriadis, A. Jacquin, "Automatic face location detection and tracking for model–assisted coding of video teleconferencing sequences at low bit rates," *Image Communication*, To appear.

[7] M.A. Fischler, R.A. Elschlager, "The representation and matching of pictorial structures," *IEEE Trans. on Computers*, January 1973.

[8] V. Govindaraju, D.B. Sher, S.N. Srihari, "Locating human faces in newspaper photographs," *Proc. IEEE Computer Society Conference on*

Computer Vision and Pattern Recognition, June 1989.

[9] V. Govindaraju, S.N. Srihari, D.B. Sher, "A computational model for face location," *Proc. Third International Conference on Computer Vision*, December 1990.

[10] F. Lavagetto, S. Curinga, "Object–oriented scene modeling for interpersonal video communication at very low bit rate," to be published in *Signal Processing: Image Communication*.

[11] C. Lettera, L. Masera, "Foreground/background segmentation in videotelephony," *Signal Processing: Image Communication*, vol. 1, no. 2, pp. 181–189, October 1989.

[12] M. Liou, "Overview of the px64 kbit/s Video Coding Standard", *Communications of the ACM*, vol. 34, no. 4, April 1991.

[13] M.M. de Sequeira, F. Pereira, "Knowledge–based videotelephone sequence segmentation," VCIP '93, vol. 2094, part 2, November 1993.

[14] V.S. Nalwa, *A guided tour of computer vision*, Addison–Wesley, 1993.

[15] *Automatic object recognition*, Edited by Hatem Nasr, SPIE Milestone Series, vol. MS 41, 1991.

[16] T. Pavlidis, *Structural pattern recognition*, Springer–Verlag, 1977.

[17] H. Ueno, K. Dachiku, K. Ohzeki, F. Sugiyama, "A study on facial region detection in the standard video coding method," *Proc. 3rd international workshop on 64 kbit/s coding of moving video*, 1990.

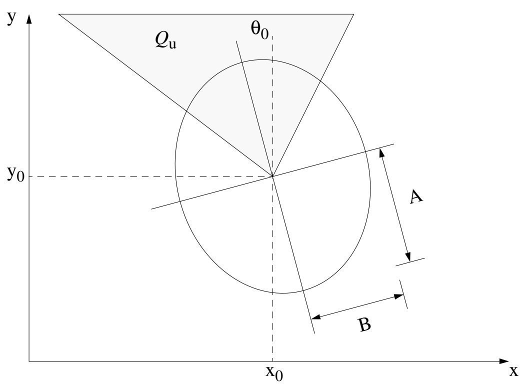

Figure 1: Elliptical face location model

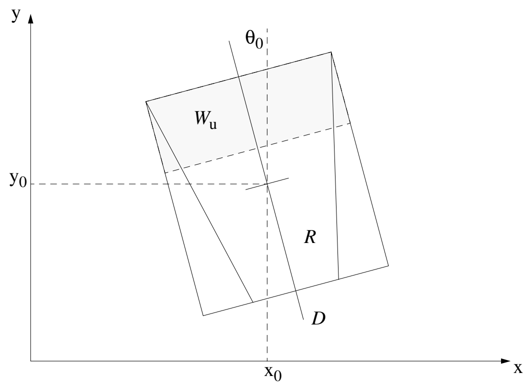

Figure 2: Rectangular window as eyes-nose-mouth location model

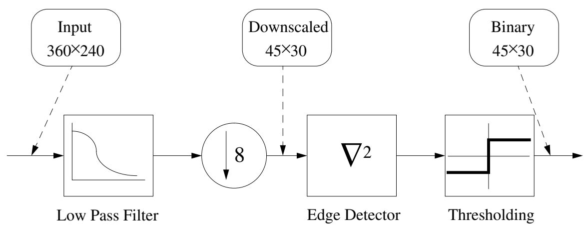

Figure 3: Block diagram of edge extraction system for face location detection.

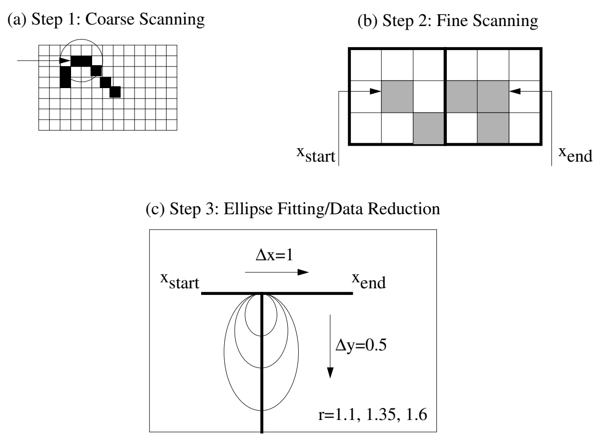

Figure 4: Algorithm for automatic face detection and tracking in video sequences.

Figure 5: Automatically detected eyes–nose–mouth locations in sequences "jelena," and "roberto."

Figure 6: Automatically detected eyes–nose–mouth locations in sequence “jim.”