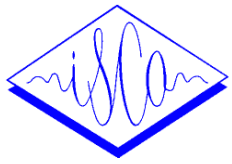
Auditory-Visual Speech Processing
(AVSP'98)
Terrigal – Sydney, Australia
December 4-6, 1998

Real-time Face and Facial Feature Tracking and Applications



Jie Yang, Rainer Stiefelhagen, Uwe Meier, Alex Waibel
Interactive Systems Laboratory
Carnegie Mellon University
Pittsburgh, PA 15213 USA
{yang+, stiefel, uwem, waibel}@cs.cmu.edu

Abstract

A human face provides a variety of different communicative functions. In this paper, we present approaches for real-time face/facial feature tracking and their applications. First, we present techniques of tracking human faces. It is revealed that human skin-color can be used as a major feature for tracking human faces. An adaptive stochastic model has been developed to characterize the skin-color distributions. Based on the maximum likelihood method, the model parameters can be adapted for different people and different lighting conditions. The feasibility of the model has been demonstrated by the development of a real-time face tracker. We then present a top-down approach for tracking facial features such as eyes, nostrils, and lip corners. These real-time tracking techniques have been successfully applied to many applications such as eye-gaze monitoring, head pose tracking, and lip-reading.

1. Introduction

Many applications in human computer interaction require tracking a human face and facial features. Locating and tracking human faces is a prerequisite for face recognition and/or facial expressions analysis, although it is often assumed that a normalized face image is available. In order to locate a human face, the system needs to capture an image using a camera and a framegrabber, to process the image, to search the image for important features, and then to use these features to determine the location of the face. In order to track a human face, the system not only needs to locate a face, but also needs to find the same face in a sequence of images. For example, in a teleconference, it is desirable to allow the participants to move freely while a face tracker tracks the current speaker. Locating and tracking features are essential for eye/gaze tracking. Human gaze indicates where a person is looking, and what he/she is paying attention to. Such information can be obtained from tracking the orientation of the person's head and the orientation of the person's eye. Many current speech recognition systems perform well on clean speech signals but perform poorly on noisy signals. Integration of acoustic and visual information (automatic lipreading) can improve overall recognition rate especially in noisy environments. A reliable face and lip tracker make lipreading possible.

We present approaches for real-time face/facial feature tracking and their applications in this paper. First, we present techniques of tracking human faces. It is revealed that human skin-colors can be used as a major feature for tracking human faces. An adaptive stochastic model has been developed to characterize the skin-color distributions. Based on the maximum likelihood method, the model parameters can be adapted for different people and different lighting conditions. The feasibility of the model has been demonstrated by the development of a real-time face tracker. The system has achieved a rate of 30+ frames/second using a low-end work station (e.g., HP9000) with a framegrabber and a camera. Once a face is located, it is much easier to locate the facial features such as eyes, nostrils, and lips. This top-down approach works very well for many applications such as gaze tracking, and lip-reading. The facial features are tracked in real-time and the head pose is estimated based on a full perspective 3D model. The eye gaze is monitored by a neural network based system. We describe some applications of the visual tracking techniques to multimodal human computer interaction. The gaze tracker has been combined with a speech recognizer in a multimodal interface to control a panorama image viewer.

2. Real-time Face Tracking

Human face perception is currently an active research area in the computer vision community. Facial features, such as eyes, nose and mouth, are natural candidates for locating human faces. These features, however, may change from time to time. Occlusion and non-rigidity are basic problems.

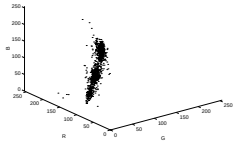
2.1. Skin Color Modeling

Color is another feature on human faces. A lot of research has been directed to understanding and making use of color information. Color has long been used for recognition and segmentation and recently

has been successfully used face locating and tracking [1, 2, 3, 4, 5]. However, color is not a physical phenomenon. It is a perceptual phenomenon that is related to the spectral characteristics of electromagnetic radiation in the visible wavelengths striking the retina [6]. There are several problems for using color as a feature to track human faces. First, the color representation of a face obtained by a camera is influenced by many factors such as ambient light, object movement, etc. Second, different cameras produce significantly different color values even for the same person under the same lighting condition. Finally, human skin colors differ from person to person. In order to use color as a feature for face tracking, we have to solve these problems.

A color histogram is a distribution of colors in the color space. It has long been used by the computer vision community in image understanding. For example, analysis of color histograms has been a key tool in applying physics-based models to computer vision. It has been shown that color histograms are stable object representations unaffected by occlusion and changes in view, and that they can be used to differentiate among a large number of objects. In the mid-1980s, it was recognized that the color histogram for a single inhomogeneous surface with highlights will have a planar distribution in color space [7]. It has since been shown that the colors do not fall randomly in a plane, but form clusters at specific points [8]. The Figure 1 shows a face image, the skin-color occurrences in the RGB color space (256x256x256), and the skin color distribution in the normalized color space. It has been observed that (1) human skin colors cluster in a small region in a color space; (2) human skin colors differ more in intensity than in colors, and (3) under a certain lighting condition, a skin-color distribution can be characterized by a multivariate normal distribution in the normalized color space [3]. These observations have been further justified by quantitative analysis goodness-of-fit techniques [9].

(a) Face image (color!)



(b) Skin-color occurrences

Figure 1. An example of a human face and the skin-color cluster

2.2 Skin Color Adaptation

Although under a certain lighting condition, the skin-color distribution of each individual is a multivariate normal distribution, the parameters of the distribution for different people and different lighting conditions are different. There are two schools of philosophy to handle environment changes: tolerating and adapting. Color constancy refers to the ability to identify a surface as having the same color under considerably different viewing conditions. Although human beings have such ability, the underlying mechanism is still unclear. The adaptive approach, on the other hand, is to transform the previous developed color model into the new environment. Since the Gaussian model has only a few parameters, it is possible to update them in real-time. Due to the fact that the linear combination of Gaussian distributions is still a Gaussian distribution, we can consider the current Gaussian distribution is a combination of the previous distributions. One way of adaptation is to use the linear combination of known parameters to predict, or, approximate the new parameters, i.e.,

$$\hat{\mu} = \sum_{k=1}^{r} \alpha_k \mathbf{m_k}, \quad \hat{\Sigma} = \sum_{k=1}^{r} \beta_k S_k, \tag{1}$$

where $\hat{\mu}$ is the estimated mean vector; $\hat{\Sigma}$ is the estimated covariance matrix; $\alpha_i \leq 1$ and $\beta_k \leq 1$ $k = 1, \ldots, r$, are weighting factors; $\mathbf{m_k}$ and S_k, $k = 1, \ldots, r$, are the previous mean vectors and covariance matrices.

We can use the maximum likelihood criterion to find the best set of coefficients for the prediction [9]. We discuss two cases: adapting mean vector only and adapting both mean vector and covariance matrix.

2.2.1 Mean Adaptation

In this case, the covariance matrix is assumed to be a constant and the mean vector μ is assumed to be a linear combination of the previous mean vectors. By setting the derivatives of the likelihood function with respect to $\alpha_k, k = 1, \ldots, r$, to 0, the equations for the maximum likelihood estimates are

$$\sum_{k=1}^{r} \mathbf{m_j}' \Sigma^{-1} \mathbf{m_k} \hat{\alpha_k} = \mathbf{m_j}' \Sigma^{-1} \bar{\mathbf{x}}, \quad j = 1, \ldots, r \tag{2}$$

We can obtain α_k by solving the equation (2).

2.2.2 Mean and Covariance Adaptation

In this case, both mean vector and covariance matrix are assumed to be a linear combination of the previous parameters. In general, explicit solutions for this problem do not exist and estimates must be performed by iterative numerical techniques.

In fact, because the two sets of estimates are asymptotically independent, each set of parameters can be estimated as if the other set of parameters is known. In the following we present an EM algorithm based on the estimate procedure proposed by Anderson [10].

The basic idea of the algorithm is to iteratively estimate two sets of parameters independently. In order to iteratively estimate $\hat{\alpha}_k^{(i)}$ and $\hat{\beta}_k^{(i)}$, where the superscript (i) denotes the ith iteration.

Algorithm

1. Initialization

$$\sum_{k=1}^{r} \mathbf{m_j}'\mathbf{m_k}\hat{\alpha_k}^{(0)} = \mathbf{m_j}'\bar{\mathbf{x}}, \quad j = 1, \ldots, r,$$

$$\hat{\mu}^{(0)} = \sum_{k=1}^{r} \hat{\alpha}_k^{(0)}\mathbf{m_k}, \quad j = 1, \ldots, r,$$

$$C^{(0)} = \frac{1}{N}\sum_{k=1}^{N}(\mathbf{x_k} - \bar{\mathbf{x}})(\mathbf{x_k} - \bar{\mathbf{x}})' + (\mathbf{x_k} - \hat{\mu}^{(\mathbf{0})})(\mathbf{x_k} - \hat{\mu}^{(\mathbf{0})})'$$

$$\sum_{k=1}^{r} tr\ S_j S_k \hat{\beta_k}^{(0)} = tr\ S_j C^{(0)}, \quad j = 1, \ldots, r,$$

$$\hat{\Sigma}^{(0)} = \sum_{k=1}^{r} \hat{\beta}_k^{(0)} S_k,$$

2. Iteration

$$\sum_{k=1}^{r} \mathbf{m_j}'\Sigma^{-1}\mathbf{m_k}\hat{\alpha_k}^{(i)} = \mathbf{m_j}'\Sigma^{-1}\bar{\mathbf{x}}, \quad j = 1, \ldots, r,$$

$$\hat{\mu}^{(i)} = \sum_{k=1}^{r} \hat{\alpha}_k^{(i)}\mathbf{m_k}, \quad j = 1, \ldots, r,$$

$$C^{(i)} = \frac{1}{N}\sum_{k=1}^{N}(\mathbf{x_k} - \bar{\mathbf{x}})(\mathbf{x_k} - \bar{\mathbf{x}})' + (\mathbf{x_k} - \hat{\mu}^{(\mathbf{i})})(\mathbf{x_k} - \hat{\mu}^{(\mathbf{i})})'$$

$$\sum_{k=1}^{r} tr\ (\hat{\Sigma}^{(i-1)})^{-1} S_j (\hat{\Sigma}^{(i-1)})^{-1} S_k \hat{\beta_k}^{(i)}$$
$$= tr\ (\hat{\Sigma}^{(i-1)})^{-1} S_j (\hat{\Sigma}^{(i-1)})^{-1} C^{(i)}, \quad j = 1, \ldots, r,$$

$$\hat{\Sigma}^{(i)} = \sum_{k=1}^{r} \hat{\beta}_k^{(i)} S_k,$$

3. If $\max(|\beta_j^{(i)} - \beta_j^{(i-1)}|,\ j = 1, \ldots, r) \leq \epsilon$ for a small number $\epsilon > 0$, stop; otherwise goto step 2.

It has been shown that the solution of these estimation equations is asymptotically efficient provided that the estimate of Σ is consistent [10].

2.3 A Real-time Face Tracker

We have developed a real-time face tracker [3]. The system has achieved a rate of 30+ frames/second using a low-end workstation (e.g., HP9000) with a framegrabber and a camera. Three types of models have been employed to track human faces. In addition to the skin-color model used to register the face, a motion model is used to estimate image motion and to predict the location of the search window. Finally a camera model predicts and compensates for camera motion (panning, tilting, and zooming). The system can track a person's face while the person moves freely (e.g., walks, jumps, sits down and stands up). The QuickTime movies of demo sequences in different situations and on different subjects can be found on our web site http://www.is.cs.cmu.edu/.

Figure 2. Iterative thresholding of the search window

3. Real-time Tracking Facial Features

Inside the found facial area, the facial features are searched and tracked inside the found facial area. In this section we describe our approaches to search and track eyes, lip-corners and nostrils.

3.1 Searching Pupils

Assuming a frontal view of the face initially, we can search the pupils by looking for two dark regions that satisfy certain anthropometric constraints and lie within a certain area of the face. For a given situation, these dark regions can be located by applying a fixed threshold to the gray-scale image. However, the threshold value may change for different people and lighting conditions. To use the thresholding method under changing lighting conditions, we developed an iterative thresholding algorithm. The algorithm iteratively thresholds the image until a pair of regions that satisfies the geometric constraints can be found. Figure 2 shows the search window for the eyes for different thresholds k_i. After three iterations, both pupils are found.

3.2 Searching Lip Corners

First, the approximate positions of the lip corners are predicted, using the positions of the eyes, the face-model and the assumption, that we have a near-frontal view. A generously big area around those points is extracted and used for further search.
Finding the vertical position of the line between the lips is done by using a horizontal integral projection P_h of the grey-scale-image in the search-region. Since the lip line is the darkest horizontally extended structure in the search area, its vertical position can be located where P_h has its global minimum.

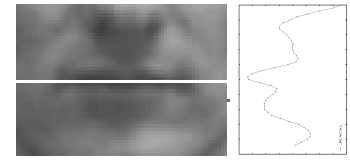


Figure 3. Integral projection of the search-window

The horizontal boundaries of the lips can be found by applying a horizontal edge detector to the refined

search area and regarding the vertical integral projection of this horizontal edge image. The positions of the lip corners can be found by looking for the darkest pixel along the two columns in the search area located at the horizontal boundaries.

horizontal edge-image

vertical projection

Figure 4. Finding horizontal borders of the lips, using a vertical projection of the horizontal edge-image of the lips

3.3 Searching Nostrils

Similar to searching the eyes, the nostrils can be found by searching for two dark regions, that satisfy certain geometric constraints. Here the search-region is restricted to an area below the eyes and above the lips. Again, iterative thresholding is used to find a pair of legal dark regions, that are considered as the nostrils.

3.4 Tracking the Eyes

For tracking the eyes, simple darkest pixel finding in the predicted search-windows around the last positions is used.

3.5 Tracking Lip Corners

Our approach to track the lip-corners consists of the following steps:

1. Search the darkest pixel in a search-region right of the predicted position of the left corner and left of the predicted position of the right corner. The found points will lie on the line between the lips

2. Search the darkest path along the lip-line for a certain distance d to the left and right respectively, and choose positions with maximum contrast along the search-path as lip-corners

Because the shadow between upper and lower lip is the darkest region in the lip-area, the search for the darkest pixel in the search windows near the predicted lip corners ensures that even with a bad prediction of the lip corners, a point on the line between the lips is found. Then the true positions of the lip corners can be found in the next step. Figure 5 shows the two search windows for the points on the line between the lips. The two white lines mark the search paths along the darkest paths, starting from where the darkest pixel in the search windows have been found. The found corners are marked with small boxes.

Figure 5. Search along the line between the lips

3.6 Tracking Nostrils

Tracking the nostrils is also done by iteratively thresholding the search-region and looking for 'legal' blobs. But whereas we have to search a relatively big area in the initial search, during tracking, the search-window can be positioned around the previous positions of the nostrils, and can be chosen much smaller. Furthermore, the initial threshold can be initialized with a value that is a little lower than the intensity of the nostrils in the previous frame. This limits the number of necessary iterations noticeably.
However, not always both nostrils are visible in the image. For example, when the head is rotated strongly to the right, the right nostril will disappear, and only the left one will remain visible. To deal with this problem, the search for two nostrils is done only for a certain number of iterations. If no nostril-pair is found, then only one nostril is searched by looking for the darkest pixel in the search window for the nostrils. To decide which of the two nostrils was found, we choose the nostril, that leads to the pose which implies smoother motion of the head compared to the pose obtained choosing the other nostril.

4. Applications

These real-time tracking techniques can be used to build non-intrusive vision-based user interfaces. We have used the described tracking techniques to build a system that estimates a user's head pose, and to obtain the visual evidence for an eye-gaze tracker and a lip-reading system. These applications will be described below.

4.1 Head Pose Tracking

A person's gaze direction is determined by two factors: the orientation of the head and the orientation of the eyes. But whereas the the eye orientations determine the exact direction of the user's gaze, the head orientation determines the overall gaze direction.
Since we know the geometry of a face, determining the orientation of the head is a pose estimation problem. In fact, the head pose can be estimated by finding correspondences between a number of head model points and their locations in the camera image. We have developed a system to estimate the head pose using a full perspective model [11]. The

method	R_x error	R_y error	R_z error
TC	5.5	7.6	2.2
SC	7.4	11.8	2.3
no pred	5.6	10.7	2.1

Table 1. Average rotation error in degrees for sequence 1.

method	T_x error	T_y error	T_z error
TC	7	4	63
SC	6	5	100
no pred	5	4	59

Table 2. Sequence 1: Average translation error in mm.

system tracks six non-coplanar facial features (eyes, lip-corners and nostrils) in real-time and estimates the head pose using an algorithm proposed by DeMenthon & Davis [13]. Table 1 and 2 show the accuracy of the head pose estimation on a test sequence that we recorded in our lab (see [11] for details. We have applied the system to a multimodal interface to control a panorama image viewer [12]. The interface allows a user to scroll through a 360 degree panorama image by his/her head pose and zoom in or zoom out by voice commands.

4.2 Eye-Gaze Monitoring

A user's eye gaze on a computer screen can be accurately estimated by a neural network using images of the eye of the user as input. Baluja and Pomerleau demonstrated such an approach by a system which used a flash light to acquire stable eye images [14]. The flash light, however, causes the problem of user acceptance. Using our real-time tracking techniques, we can obtain stable images of the eyes without the need of special lighting. In our system as described in [15] the user's eyes are automatically tracked, images of the eyes are extracted, preprocessed and are used as input to a neural net, which estimates the x- and y-coordinates of the user's focus on the screen.
The system consists of a three layer network with 400 input units (for the two eye images), 40 to 50 hidden units and 2 x 50 output units for Gaussian output representation of the x- and the y-coordinates of the focused point on the screen.
Figure 6 shows two sample pairs of extracted and histogram normalized eye images that are used as input to the neural nets.
We have trained and tested neural nets for each of four different user's and also for all four users. In the user dependent case we achieved mean error on

Figure 6. Sample input images for the neural net (20x10 pixel)

test sets of between 1.3 and 1.8 degrees with the best neural nets. For the multi-user neural net the mean error was 1.9 degrees using the best net. Table 3 shows these results.

user	mean error
A	1.5
B	1.3
C	1.6
D	1.8
A,B,C,D	1.9

Table 3. Average error of eye gaze estimation (in degrees)

4.3 Lipreading

It has been demonstrated that visual information can enhance accuracy of speech recognition for both a human and a computer. However, many other lipreading systems require a user keep still or put special marks on his/her face. Using the face and lip tracking techniques discussed above, we have developed a lipreading system that gives a speaker reasonable freedom of movement within a room [16, 17]. The system is based on a modular MS-TDNN structure. The visual and acoustic TDNNs are trained separately, and visual and acoustic information are combined at the phonetic level. We use gray-scale images of the lip-region as visual input. Adaptive gray-value modification is used to eliminate different lightning conditions [17]. The speech signal is preprocessed and 16 melscale coefficients are fed into the neural network. The current system is for German spelling task, mainly in the speaker-dependent mode. Letter sequences of arbitrary length and content are spelled without pauses. Words in our database are 8 letters long on average. The task is therefore equivalent to continuous recognition with a small but highly confusable vocabulary.We have trained a speaker dependent recognizer on 170 sequences of acoustic/visual data, and tested on 30 sequences. For testing we also added white noise to the test-set. The results are shown in Table 4 as performance measure word accuracy is used where a spelled letter is considered a word. The current system has achieved an error reduction of up to 55% compared to the system which only uses acoustic recognition.

TestSet	clean	16dB SNR	8 dB SNR
visual only	55%	55%	55%
acoustic only	98.4%	56.9%	36.2%
combined	99.5%	73.4%	66.5%

Table 4. Speaker-dependent results

5. Conclusion

We have presented techniques for real-time tracking of human faces and facial features. We have demonstrated that human skin color distributions can be characterized by an adaptive statistic model and a system can track a face in real-time using such a model. We have presented a top-down approach to track facial features. It has been shown that facial features such as eyes, lips and nostrils can be located and tracked with this approach. We have described applications of the real-time tracking techniques. We are currently working on applying these visual tracking techniques to multimodal human computer interfaces to improve human computer interaction.

References

[1] M. Hunke and A. Waibel. Face locating and tracking for human-computer interaction. In *Proc. Twenty-Eight Asilomar Conference on Signals, Systems & Computers*, Monterey, CA, USA, 1994.

[2] T.C. Chang, T.S. Huang, and C. Novak. Facial feature extraction from color images. In*Proc. the 12th IAPR International Conference on Pattern Recognition*, Vol. 2, pages 39-43, 1994.

[3] J. Yang and A. Waibel A real-time face tracker. In *Proceedings of the Third IEEE Workshop on Applications of Computer Vision*, pages 142-147, 1996 ("Tracking human faces in real-time," Technical Report CMU-CS-95-210, CS department, CMU, 1995).

[4] N. Oliver, A. Pentland, and F. Berard. LAFTER: lips and face realtime tracker. In *Proceedings of CVPR'97*, pages 123-129, 1997.

[5] J.L. Crowley and F. Berard. Multimodal tracking of faces for video communications In *Proceedings of CVPR'97*, pages 640-645, 1997.

[6] G. Wyszecki and W.S. Styles. Color Science: Concepts and Methods, Quantitative Data and Formulae (Second Edition). John Wiley & Sons, New York, 1982.

[7] S.A. Shafer. Optical phenomena in computer vision. In *Proc. Canadian Soc. Computational Studies of Intelligence*, pages 572-577, 1984.

[8] G.J. Klinker, S.A. Shafer, and T. Kanade. Using a color reflection model to separate highlights from object color. In *Proc. ICCV*, pages 145-150, 1987.

[9] J. Yang, W. Lu, and A. Waibel. Skin-color modeling and adaptation. In *Proceedings of ACCV'98* (Technical Report CMU-CS-97-146, CS department, CMU, 1997).

[10] T.W. Anderson. Asymptotically efficient estimation of covariance matrices with linear structure. The Annals of Statistics, Vol. 1, No. 1, pages 135-141, 1973.

[11] R. Stiefelhagen, J. Yang, and Alex Waibel. A model-based gaze tracking system. In *Proceedings of IEEE International Joint Symposia on Intelligence and Systems*, pages 304 – 310, 1996.

[12] R. Stiefelhagen and J. Yang. Gaze tracking for multimodal human-computer interaction. In *Proceedings of International Conf. on Acoustics, Speech, and Signal Processing*, April 1997.

[13] D. F. DeMenthon and L. S. Davis. Model based object pose in 25 lines of code. In *Proceedings of Second European Conference on Computer Vision*, pages 335 – 343. Springer Verlag, May 1992.

[14] S. Baluja and D. Pomerleau. Non-intrusive gaze tracking using artificial neural networks. Technical Report CMU-CS-94-102, Carnegie Mellon University, 1994.

[15] R. Stiefelhagen, J. Yang, and A. Waibel. Tracking eyes and monitoring eye gaze. In *Workshop on Perceptual User Interfaces*, Banff, Canada, October 1997.

[16] U. Meier, W. Hürst, and P. Duchnowski. Adaptive bimodal sensor fusion for automatic speechreading. In *Proceedings of International Conf. on Acoustics, Speech, and Signal Processing*, 1996.

[17] U. Meier, R. Stiefelhagen, and J. Yang. Preprocessing of visual speech under real world conditions. In *Proceedings of European Tutorial & Research Workshop on Audio-Visual Speech Processing*, 1997.