MIT. Media Laboratory Affective Computing Technical Report #550
Appears in Proceedings from 5th International Conference on Automatic Face and Gesture Recognition, May 2002

Real-Time, Fully Automatic Upper Facial Feature Tracking

Ashish Kapoor and Rosalind W. Picard
MIT Media Laboratory
20 Ames Street,
Cambridge, MA 02139
{ash, picard}@media.mit.edu

Abstract

Robust, real-time, fully automatic tracking of facial features is required for many computer vision and graphics applications. In this paper, we describe a fully automatic system that tracks eyes and eyebrows in real time. The pupils are tracked using the red eye effect by an infrared sensitive camera equipped with infrared LEDs. Templates are used to parameterize the facial features. For each new frame, the pupil coordinates are used to extract cropped images of eyes and eyebrows. The template parameters are recovered by PCA analysis on these extracted images using a PCA basis, which was constructed during the training phase with some example images. The system runs at 30 fps and requires no manual initialization or calibration. The system is shown to work well on sequences with considerable head motions and occlusions.

1. Introduction

Facial feature tracking has attracted a lot of attention due to its applications in fields of computer vision and graphics and a lot of research has been done to capture the communicative ability of the face. Applications like facial expression analysis, animation and coding need to track the facial features robustly and efficiently. Also, rather then just tracking the positions of facial features, we need to get the shape information as well. The variability in appearance of facial features changes due to pose, lighting, facial expressions etc making the task difficult and complex.

Even harder is the task of tracking the facial features robustly in real time, without any manual alignment or calibration. Many previous approaches have focused just on tracking the location of the facial features or require some manual initialization/intervention. In this paper, we describe a fully automatic system that requires no manual intervention and robustly tracks upper facial features in detail using templates in real time at 30 fps.

2. Previous Work

There is much prior work on detecting and tracking the facial features. Many feature extraction methods are based on deformable templates [10], which are difficult to use for real-time tracking and have to be initialized properly to achieve a good performance. Tian et al [8, 9] use multiple-state templates to track the facial features. Feature point tracking together with masked edge filtering is used to track the upper facial features. The system requires that templates be manually initialized in the first frame of the sequence, which prevents it from being automatic. Essa et al [4] analyze the facial expressions using optical flow in an estimation and control framework coupled with a physical model describing the skin and muscle structure of face. The shortcoming of their system is that it does not recover the detailed description of the facial features.

The tracking of facial features in detail requires recovering the parameters that drive the shape of the feature. A lot of research has been directed towards recovering shape using image matching techniques. Jones and Poggio [6] used a stochastic gradient-descent based technique to recover the shape and the texture parameters. Cootes et al [1] use active appearance models, which are statistical appearance models generated by combining a model of shape variation with a model of texture variation, for the purpose of image matching. The iterative nature of both these approaches and performance dependence on the initialization makes it difficult to use them in real time. Covell et al [2, 3] exploit the coupling between the shape and the texture parameters using example images labeled with control points. The shape is recovered from the image appearance in an non-iterative way using eigen analysis. Given an image of the face, this approach first needs to estimate the location of features of interest, which is again very difficult to do robustly in real time.

Morimoto et al [7] have described a system to detect and track pupils using the red-eye effect. Haro et al [5] have extended this system to detect and track the pupils using a Kalman filter and probabilistic PCA. Our infrared camera equipped with infrared LEDs, which is used to highlight and track pupils, is an in-house built version of the IBM Blue Eyes camera (http://www.almaden.ibm.com/cs/blueeyes).

Our system aims to combine the best of these methods, with several new improvements. We start with a simplified version of Covell's approach of estimating both shape and texture [2, 3]. Her method has a problem with requiring a good first estimate of the feature points; we solve this problem by using an infrared camera equipped with infrared LEDs, an in-house built version of the IBM Blue Eyes camera (http://www.almaden.ibm.com/cs/blueeyes), together with a combined detection/tracking procedure we developed. Thus, the part of our approach that locates the pupils is perhaps most similar to that of Morimoto et al [7] and Haro et al [5], while the feature tracking is most similar to that of Covell. However, the combination is new, and involves some simplifications that allow it to work in real time without any initialization. The method presented here is the first we know of that combines the strengths of robust IR pupil detection with flexible learning of estimators for shape and texture for feature point tracking.

The next section describes the overall architecture of the system, including our approach for detection and tracking of facial features. Followed by that, we demonstrate tracking results and discuss some experimental evaluations.

3. The Overall System

The overall system is shown in figure 1. An infrared sensitive camera synchronized with infrared LEDs is used as a sensor and produces an image with highlighted pupils. The image obtained from the sensor is used to extract the position of the pupils. The location of pupils are used to locate other facial features and recover their the shape information using a PCA based technique. The whole system is very efficient and runs in real time at 30 fps.

3.1. The Sensor

The pupil tracking system is shown in figure 2. The whole unit is placed under the monitor pointing towards the users face. The system has an infrared sensitive camera coupled with two concentric rings of infrared LEDs. One set of LEDs is on the optical axis and produces the red-eye effect. The other set of LEDs, which are off axis, keeps the scene at about the same illumination. The two sets of LEDs are synchronized with the camera and are switched

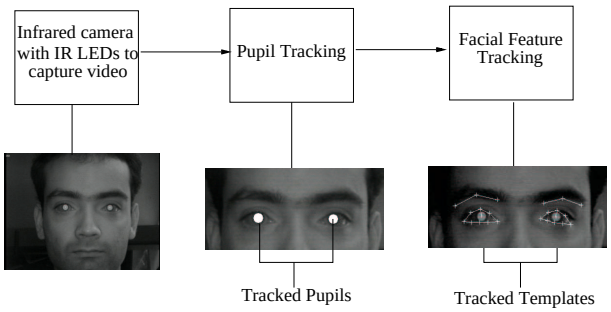

Figure 1. The Overall System.

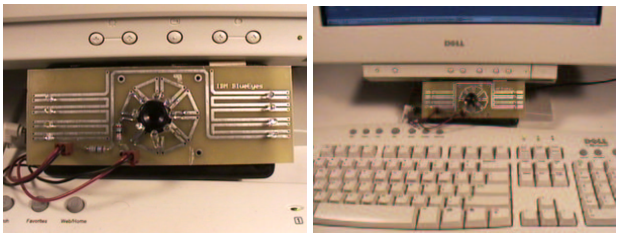

Figure 2. Camera to track pupils, placed under the monitor.

on and off to generate two interlaced images for a single frame. The image where the on-axis LEDs are on has white pupils whereas the image where the off-axis LEDs are on has black pupils. These two images are subtracted to get a difference image, which is used to track the pupils. Figure 3 shows a sample image, the de-interlaced images and the difference image obtained using the system.

3.2. Pupil Tracking

The pupils are detected and tracked using the difference image, which is noisy due to the interlacing and motion artifacts. Also, objects like glasses and earrings can show up as bright spots in the difference image due to their specularity. To remove this noise we first threshold the difference image using an adaptive thresholding algorithm [5]. First, the algorithm computes the histogram and then thresholds the image keeping only 0.1 % of the brightest pixels. All the non-zero pixels in the resulting image are set to 255 (maxval). The thresholded image is used to detect and to track the pupil. The pupil tracker is either in a detection mode or a tracking mode. Whenever there is information about the pupils in the previous frame the tracker is in tracking mode and whenever the previous frame has no information about the pupils the tracker switches to the detection mode. The pupil tracking algorithm is shown in figure 4.

During the tracking mode the tracker maintains a state

vector, comprised of the spatial information about the pupils. Specifically, the average distance between the pupils during the current tracking phase and their x, y coordinates in the previous frames is maintained. To obtain the new positions of pupils a search for the largest connected component is limited to a bounding box centered on previous pupils. The new connected components are accepted as valid pupils when they satisfy a number of spatial constraints. If the area is greater and the displacement of their centers from previous pupil position lies below a certain threshold, the connected components are considered valid. Also if a connected component is found for both the eyes then the distance between these pupils is also compared with the average distance maintained in the state space to rule out false detections. Once the connected components are identified as valid pupil regions, the state vector is updated.

The tracker switches to the detection mode whenever there is no information about the pupils. In this mode the tracker simply selects the two largest connected components that have an area greater then a certain threshold. Again, to validate the regions, we apply some spatial constraints. This approach allows us to track the pupils efficiently. Head movements during head nods and head shakes do produce motion artifacts but due to the nature of our algorithm to spatially constrain the search space, it tracks the pupils well. In extreme cases when head movements are too fast, the pupils are lost as motion artifacts overpower the red-eye effect and the pupils are absent from the difference image altogether. We found this tracking algorithm to be fairly reliable.

3.3. Facial Feature Tracking

We use templates to represent the detailed shape information of the facial features. Eye and eyebrow templates, used in our system, are shown in figure 5. A set of 8 points placed on the eye contour describes the shape of an eye. Two of these points correspond to the eye corners and the rest are equidistant on the contour. Similarly 3 points are used to describe the shape of an eyebrow. A total of 22 points (8 for each eye and 3 for each eyebrow) are used to describe the positions and shapes of upper facial features. Tian et al [8, 9] use a template that consists of the iris location and two parameterized parabolas, which would fit the lower and the upper eyelid. They use the corners of eyes and center points on the eyelids to extract the template parameters. As our system tracks more points just than eye-corners and center points on the eyelid, it can represent more detailed and accurate information about the shape.

The pupil positions obtained are used to crop out *regions of interest*: two 140 x 80 pixel images of the eyes and two 170 x 80 pixel images of eyebrows. The template parameters for the features are recovered by PCA analysis on these extracted images using a PCA basis, which was constructed during the training phase with some example images. We describe the method to recover these parameters below in detail.

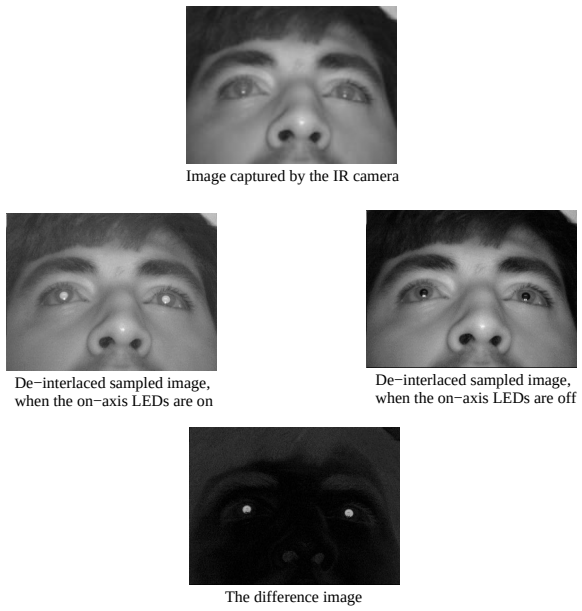

Figure 3. Pupil tracking using the infrared camera.

3.3.1 Recovering Template Parameters

Our system exploits the fact that it can estimate the location of facial features very robustly in the image. Once the facial features are located, *regions of interest* corresponding to eyes and eyebrows are cropped out and analyzed to recover the shape description. Given some example images, as training data, with hand marked control points that describe the shapes, our approach tries to express the new feature image as a linear combination of examples. We apply the same linear combination to the marked control points to recover the control points for the new image. Since the number of example images is large, we use principal component analysis (PCA) to recover the template parameters. Given n example images $\mathbf{I_i}$, let $\mathbf{p_i}$ ($i = 1..n$) be vectors corresponding to the marked control points on each image. If $\overline{\mathbf{I}}$ is the mean image, then the covariance matrix of the training images can be expressed as:

$$\Lambda = A \cdot A^T \text{ where } A = [\mathbf{I_1} - \overline{\mathbf{I}}, \mathbf{I_2} - \overline{\mathbf{I}}, ..., \mathbf{I_n} - \overline{\mathbf{I}}]$$

The eigenvectors of Λ can be computed by first computing the eigenvectors for $A^T \cdot A$. If $V = [\mathbf{v_1}, \mathbf{v_2}, ..\mathbf{v_n}]$ where $\mathbf{v_i}$

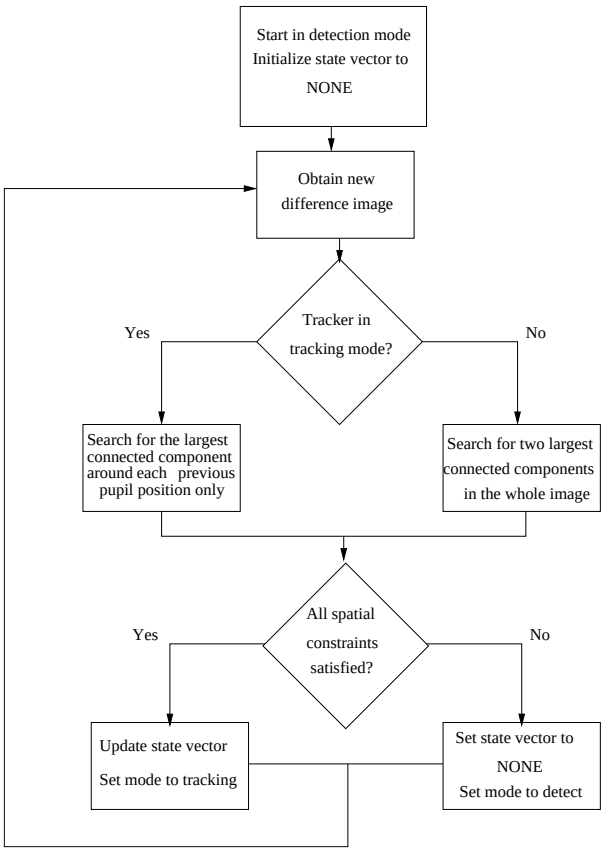

Figure 4. The Pupil tracking Algorithm.

represents the eigenvectors of $A^T \cdot A$, then the eigenvectors $\mathbf{u_i}$ of Λ can be computed as:

$$U = [\mathbf{u_1}, \mathbf{u_2}, ..., \mathbf{u_n}] = A \cdot V$$

As the eigenvectors are expressed as a linear combination of example images, we can place the control points on the eigen images using the same linear combination. Let $\overline{\mathbf{p}}$ be the mean of the vectors corresponding to the control points in example images and let $\mathbf{Ep_i}$ $(i = 1..n)$ be the vector corresponding to the control points on an eigenvector, then:

$$[\mathbf{Ep_1}, \mathbf{Ep_2}, ..., \mathbf{Ep_n}] = [\mathbf{p_1} - \overline{\mathbf{p}}, ..., \mathbf{p_n} - \overline{\mathbf{p}}] \cdot V$$

To recover the vector of control points on a new image, we first express the new image as a linear combination of the eigenvectors by projecting it onto the top few eigenvectors.

$$\mathbf{I_{new}} = \sum_i a_i \mathbf{u_i} + \overline{\mathbf{I}}$$

where $a_i = (\mathbf{I_{new}} - \overline{\mathbf{I}})^{\mathbf{T}} \cdot \mathbf{u_i}$ and $\mathbf{u_i}$ is i^{th} eigenvector.The same linear combination is applied to the vectors of control points on the eigenvectors to recover the new control points.

$$\mathbf{p_{new}} = \sum_i a_i \mathbf{Ep_i} + \overline{\mathbf{p}}$$

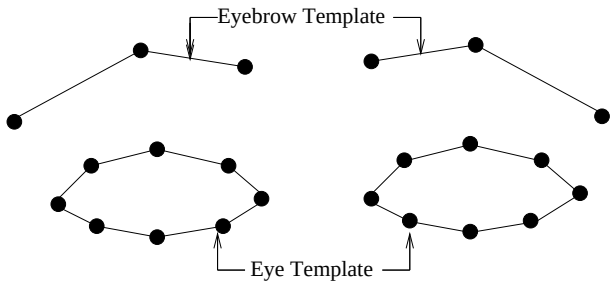

Figure 5. Eye and Eyebrow Templates.

This strategy is a simplification of the approach used by Covell et al[2, 3]. The non-iterative nature of the approach makes it ideal to be used in a real-time system.

In our system, 150 images of eyes and eyebrows from ten different individuals with different facial expressions and different lighting conditions were used in the training set. These images were hand marked to fit the feature templates. Eigenvectors and control points on those eigenvectors were computed as described. During the real-time tracking the extracted images of the features are projected on the first forty eigenvectors. These projections are used to recover the control points using the approach explained above.

4. Results

The system was implemented and worked in real time at 30 fps on a Pentium-III 933 MHz Linux machine. The system was tested on many different subjects in different lighting conditions. The system worked particularly well on the subjects who had their images in the training database.The system is very efficient and is robust to occlusions and recovers very quickly when the feature reappears.

Figure 6 and 7 show tracking results of some sequences. Both the subjects appearing in figure 6 were in the training database. The system is able to track the features very well. Note that in the first sequence of figure 6 the left eyebrow is not tracked in frames 67, 70 and 75 as it is not present in the image. Similarly all the templates are lost in the frame 29 in the second sequence of figure 6 when the pupils are absent, as the subject blinks. The templates are recovered as soon as the features reappear. To evaluate the system performance, eye and eyebrow corners were handmarked in the 93 frames of the first sequence appearing in figure 6. These points were compared with the points tracked automatically by the system. Figure 8 shows the frame by frame RMS difference per control point between the points manually marked and points tracked by the system for each frame. The mean RMS difference for the whole sequence is 0.65 pixels per control point location.

Figure 7 shows the tracking results for the subjects not

Frame 57 Frame 67 Frame 70 Frame 75 Frame 88

Frame 27 Frame 28 Frame 29 Frame 30 Frame 31

Figure 6. Tracking Results for Subjects in Training Set.

Frame 59 Frame 60 Frame 61 Frame 68 Frame 87

Frame 25 Frame 28 Frame 38 Frame 45 Frame 55

Figure 7. Tracking Results for Subjects not in Training Set.

in the training set. Again, note that the second frame in the first sequence does not show any eyes or eyebrows, due to the fact that the subject blinked and hence no pupils were detected. The tracking is recovered in the very next frame when the pupils are visible again. The first sequence appearing in figure 7, which is 100 frames long, was hand marked to locate the positions of inner and outer corners of eyes and eyebrows . Figure 9 shows framewise RMS difference per control point between the hand marked and the tracked points. The mean RMS difference for the whole sequence is 0.78 pixels per control point.

The results show that the system is very efficient, runs in real time at 30 fps and is able to track upper facial features robustly in presence of large head motions and occlusions. One limitation of our implementation is that it is not invariant to large zooming in or out as our training set did not have samples with scale changes. Also in few cases with some new subjects, the system did not work well, as the training images were not able to span the whole range of variations in appearance of the individuals. A training set which captures the variations in appearance should be able to overcome these problems.

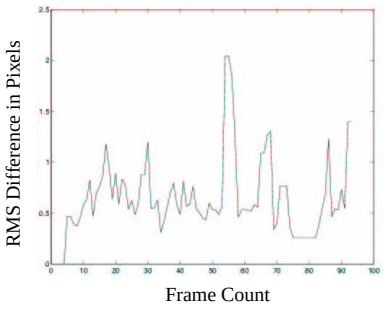

Figure 8. RMS Difference in Pixels for sequences with Subjects in Training Set.

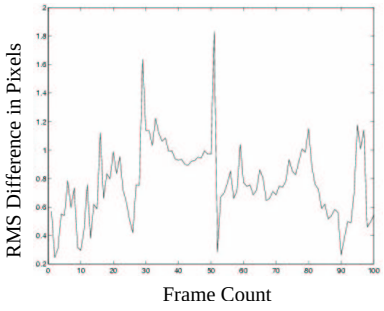

Figure 9. RMS Difference in Pixels for sequences with Subjects not in Training Set.

5. Conclusions and Future Work

We have described a real-time system that tracks upper facial features robustly without any manual alignment or calibration. An infrared camera equipped with infrared LEDs is used to track the pupils. The pupil positions are used to extract the images of eyes and eyebrows. Principal component analysis on these images is used to recover the template parameters. The system is shown to work well on sequences with considerable head motions and occlusions.

The work presented in this paper is work in progress to build a fully automatic facial expression analysis tool. Future work includes extending the system to track lower facial features and using it as an interface to build socially and emotionally intelligent systems.

6. Acknowledgements

We thank Dave Koons and Ted Selker for their help to build the IBM Blue Eyes camera. This research was supported by NSF ROLE grant 0087768.

References

[1] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *Pattern Analysis and Machine Intelligence*, 23(6), June 2001.

[2] M. Covell. Eigen-points. In *Proceedings of International Conference Image Processing*, September 1996.

[3] M. Covell. Eigen-points: control-point location using principal component analyses. In *Proceedings of Conference on Automatic Face and Gesture Recognition*, October 1996.

[4] I. Essa, S. Basu, T. Darrell, and A. Pentland. Modeling, tracking and interactive animation of faces and heads using input from video. In *Proceedings of Computer Animation Conference*, 1996.

[5] A. Haro, I. Essa, and M. Flickner. Detecting and tracking eyes by using their physiological properties. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, June 2000.

[6] M. J. Jones and T. Poggio. Multidimensional morphable models. In *Proceedings of International Conference on Computer Vision*, 1998.

[7] C. Morimoto, D. Koons, A. Amir, and M. Flickner. Pupil detection and tracking using multiple light sources. Technical report, IBM Almaden Research Center, 1998.

[8] Y. Tian, T. Kanade, and J. F. Cohn. Dual-state parametric eye tracking. In *Proceedings of Conference on Automatic Face and Gesture Recognition*, 2000.

[9] Y. Tian, T. Kanade, and J. F. Cohn. Recognizing upper face action units for facial expression analysis. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, June 2000.

[10] A. Yuille, P. Haallinan, and D. S. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 1992.