Robust Pose Invariant Facial Feature Detection and Tracking in Real-Time

Zhiwei Zhu
Sarnoff Corporation
zzhu@sarnoff.com

Qiang Ji
Department of ECSE, RPI
qji@ecse.rpi.edu

Abstract

In this paper, a robust technique is proposed to detect and track a set of twenty-eight prominent facial features under various facial expressions and face orientations in real-time. Specifically, after the face image is captured from the camera, a trained face mesh is first employed to estimate a rough position for each facial feature based on the located eye positions. Subsequently, an accurate position is obtained for each facial feature by searching around its roughly estimated position. Once the facial features are located, by using the appearance information of each facial feature together with the geometry information among the facial features, a shape-constrained correction-based tracking mechanism is activated to track them in the subsequent image frames. Finally, the performance of the proposed technique is demonstrated through building a real-time facial feature tracking system that can detect and track a set of twenty-eight facial features automatically as soon as a person is sitting in front of the camera.

1 Introduction

Facial features, such as eyes, eyebrows, nose and mouth, as well as their spatial arrangement, are important for the facial interpretation tasks including face recognition [1], facial expression analysis [2] and face animation [3]. Therefore, accurately locating these facial features in a face image is crucial for these tasks to perform well. Various techniques [4], [5], [6], [1], [7] have been proposed to detect and track facial features in face images. In general, two types of information are commonly utilized by these techniques. One is the image appearance of the facial features, which is referred as texture information; and the other is the spatial relationship among different facial features, which is referred as shape information.

In [6], a neural network is trained individually as a feature detector for each facial feature. Facial features are then located by searching the face image via the trained facial feature detectors. Similarly, Gabor wavelet networks are trained to locate the facial features in [5]. Since the shape information of the facial features is not modelled explicitly in both techniques, they are prone to image noise. Therefore, in [4], a statistical shape model is built to capture the spatial relationships among facial features, and multi-scale and multi-orientation Gaussian derivative filters are employed to model the texture of each facial feature. However, only the shape information is used when comparing two possible feature point configurations, ignoring the local measurement for each facial feature. Such a method may not be robust in the presence of image noise. In [1], facial features are represented with the Gabor Jets and the spatial distributions of facial features are captured with a graph structure implicitly. Via the graph structure, only the simple spatial information among the facial features is imposed, whose variation is not modelled directly. Since most of these proposed techniques either assume frontal facial views, or without significant facial expressions, or under constant illuminations, good performance has been reported. However, in reality, the image appearance of the facial features varies significantly among different individuals. Even for a specific person, the appearance of the facial features is easily affected by the lighting conditions, face orientations and facial expressions. Therefore, robust facial feature tracking still remains a very challenging task, especially under variable illumination, face orientation and facial expression.

In this paper, a novel technique is proposed to improve the robustness and accuracy of the existing facial feature trackers such that facial features can be detected and tracked under the above challenging situations. First, Kalman filtering [8] is utilized to predict the position for each facial feature in a new image so that a smooth constraint can be imposed on the motion of each facial feature. Next, given the predicted feature positions, the multi-scale and multi-orientation Gabor wavelet matching method [9], [1], [2] is used to detect each facial feature in the vicinity of the predicted locations. In addition, the robust matching in the Gabor space also provides an accurate and fast solution for tracking multiple facial features simultaneously.

During tracking, in order to adaptively compensate for the facial feature appearance changes, the Gabor wavelet coefficients are updated dynamically at each frame to serve as the tracking template for the subsequent image frame. This updating approach works very well when no significant appearance change happens for each facial feature. However, under the large face orientations in which an arbitrary profile is assigned to the occluded feature during self-occlusions, as well as the significant facial expressions in which the facial feature appearance varies significantly, the tracker often fails. Therefore, a shape-constrained correction mechanism is developed to tackle the above problems and to refine the tracking results. As a result, via our proposed technique, a set of twenty-eight facial features can be detected and tracked robustly in real-time under significant appearance changes in various facial expressions and face orientations.

Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

2 Facial Feature Representation

As shown in Figure 1, twenty-eight prominent facial features around eyes, eyebrows, nose and mouth are selected to detect and track in the face images. A set of multi-scale and multi-orientation Gabor wavelets are applied to each facial feature. Specifically, the set of utilized Gabor wavelets consist of 60 Gabor kernels, including 10 spatial frequencies and 6 distinct orientations. Therefore, for each facial feature $\vec{\mathbf{x}}$, a Gabor coefficient vector $J(\vec{\mathbf{x}})$ is derived as a set of 60 convolution results $\{J_j(\vec{\mathbf{x}})\}$ with the above 60 Gabor kernels. This Gabor coefficient vector $J(\vec{\mathbf{x}})$ can be used to represent each facial feature $\vec{\mathbf{x}}$ and its vicinity [9] efficiently. In our proposed algorithm, the Gabor coefficient vector $J(\vec{\mathbf{x}})$ is not only used to detect each facial feature at the initial frame, but also used during tracking.

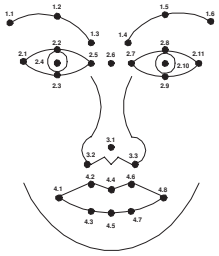


Fig. 1. A face mesh with facial features

2.1 Phase-Based Displacement Estimation

Given a facial feature at two consecutive image frames, a Gabor coefficient vector $J(\vec{x})$ is extracted at the position $\vec{\mathbf{x}}$ in the first image, while another Gabor coefficient vector $J(\vec{y})$ is extracted at a different position $\vec{y} = \vec{x} + \vec{\mathbf{d}}$ in the subsequent image frame, with the displacement $\vec{\mathbf{d}} = (dx, dy)^T$. Since the position $\vec{x}$ is in a small vicinity of the position $\vec{y}$, the phase shift between Gabor coefficient vectors $J(\vec{x})$ and $J(\vec{y})$ can approximately be compensated for by the term $\vec{\mathbf{d}} \cdot \vec{\mathbf{k_j}}$, where $\vec{\mathbf{k_j}}$ is the wave vector of each Gabor kernel. So the phase-sensitive similarity function between these two Gabor coefficient vectors can be expressed as:

$$\mathbf{S}(\mathbf{J}(\vec{\mathbf{x}}), \mathbf{J}(\vec{\mathbf{y}})) = \frac{\sum_{\mathbf{j}} \mathbf{m_j m_j^{'}} \cos(\phi_{\mathbf{j}} - \phi_{\mathbf{j}}^{'} - \vec{\mathbf{d}} \cdot \vec{\mathbf{k_j}})}{\sqrt{\sum_{\mathbf{j}} \mathbf{m_j^2} \sum_{\mathbf{j}} \mathbf{m_j^{'2}}}} \quad (1)$$

where each component in the Gabor coefficient vector $J(\vec{x})$ is represented as $J_i(\vec{x}) = m_j e^{i\phi_j}$, and each component in the Gabor coefficient vector $J(\vec{y})$ is represented as $J_i(\vec{y}) = m_j^{'} e^{i\phi_j^{'}}$. To compute it, the displacement $\vec{d}$ must be estimated. This can be done by maximizing the similarity in its second Taylor expansion:

$$\mathbf{S}(\mathbf{J}(\vec{\mathbf{x}}), \mathbf{J}(\vec{\mathbf{y}})) \approx \frac{\sum_{\mathbf{j}} \mathbf{m_j m_j^{'}} [\mathbf{1} - \mathbf{0.5}(\phi_{\mathbf{j}} - \phi_{\mathbf{j}}^{'} - \vec{\mathbf{d}} \cdot \vec{\mathbf{k_j}})^{\mathbf{2}}]}{\sqrt{\sum_{\mathbf{j}} \mathbf{m_j^2} \sum_{\mathbf{j}} \mathbf{m_j^{'2}}}} \quad (2)$$

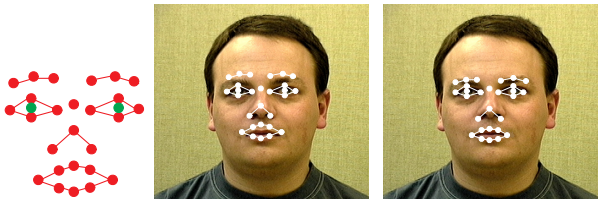
Therefore, by setting $\frac{\partial}{\partial dx}\mathbf{S} = \mathbf{0}$ and $\frac{\partial}{\partial dy}\mathbf{S} = \mathbf{0}$, the optimal displacement vector $\vec{d}$ can be estimated efficiently. As a result, via the above displacement estimation technique, the optimal position of each facial feature in a new image can be estimated automatically, which dramatically speeds up the displacement estimation processing and makes the real-time implementation for multiple-feature tracking possible.

3 Facial Feature Detection

Our proposed algorithm starts with automatically detecting these twenty-eight facial features in the initial image frames. In essence, the proposed facial feature detection technique consists of two steps: facial feature approximation and facial feature refinement. The first step provides an approximated location for each facial feature based on two detected eye positions, which is then fine-tuned in the second step.

3.1 Facial Feature Approximation

After the face image is captured from a camera, a frontal face is first localized via a trained Adaboost face detector [10]. Subsequently, based on the detected face region, the eye positions can be detected via a trained Adaboost eye detector [10]. Once the eye positions are known, based on them, a trained face mesh F as shown in Figure 2 (a) is resized and imposed on the face image to estimate a rough position for each facial feature. The face mesh F is obtained by taking the mean face from a set of frontal faces that covers a variety of different people. Since the deviation from the actual position is usually small for each facial feature as shown in Figure 2 (b), its position can be further refined by searching around its estimated position in the subsequent refinement step.



(a) (b) (c)

Fig. 2. (a) face mesh, (b) face image with approximated facial features, (c) face image with refined facial features

3.2 Facial Feature Refinement

Given an approximated position $\vec{x}_e$ for a facial feature in the face image, a Gabor coefficient vector $J(\vec{x}_e)$ is first computed. Then, the nearest neighbor searching approach is utilized to seek a most similar Gabor coefficient vector $J^{'}$ from a training set by matching with $J(\vec{x}_e)$ in the Gabor space. In order to obtain an effective training set for each facial feature, a large number of frontal face images that include different individuals, different lighting conditions and different facial expressions were collected.

Once the most similar $J^{'}$ is selected from the training set for each facial feature, it is utilized as a model to estimate a new position $\vec{x}^{'}$ starting from the approximated position $\vec{x_e}$ via the fast phase-based displacement estimation technique proposed in Section 2.1. The above procedure repeats until

the final estimated position $\vec{x}'$ converges or the pre-defined number of iterations exceeds. Finally, as shown in Figure 2 (c), the facial features can be located successfully via the proposed facial feature detection technique.

4 Facial Feature Tracking

Once the facial features are located, they are tracked in the subsequent image frames. The facial feature tracking, especially tracking a set of facial features simultaneously is a crucial and difficult task. The appearance of the facial features and their spatial relationship can vary significantly under changes in facial expression and face orientation. Therefore, tracking these facial features robustly is a tough issue. However, we propose a novel facial feature tracking scheme that can track them robustly under large head movements and significant facial expressions. Specifically, it consists of three stages, namely facial feature prediction, facial feature measurement and facial feature correction.

During the facial feature prediction stage, Kalman filtering is used to predict the position of each facial feature in a new image frame from its previous locations. Subsequently, during the facial feature measurement stage, a searching process within an area centered at the predicted position is often used to detect its optimal position. But when tracking a number of facial features simultaneously, exhaustively searching for each facial feature is very time-consuming. Instead, it is done automatically via the proposed fast phase-based displacement estimation technique described in Section 2.1. Hence, the displacement $\overrightarrow{d}$ for each facial feature can be estimated to obtain its position efficiently.

Once the facial features are located, a Gabor wavelet coefficient vector will be extracted and serve as the tracking model in the subsequent image frame for each facial feature. It is equivalent to updating the tracking model dynamically in each image frame; hence, the appearance changes of each facial feature in the image frames can be adapted. However, since it does not have the ability to correct the possible errors during tracking, errors may be accumulated over the image frames such that the tracker will drift away eventually. Therefore, in the next stage, a correction step is proposed to eliminate the errors associated with each facial feature to avoid drifting.

4.1 Facial Feature Correction

The proposed facial feature correction strategy consists of two components: refining facial features and imposing shape constraint. In the facial feature refinement component, the technique discussed in Section 3.2 is utilized. Differently, for each facial feature, a different training set is obtained by extracting a set of Gabor wavelet coefficient vectors from a large number of face images collected under various face orientations, instead of frontal faces only. Then, the tracked position of each facial feature is refined by matching with the new training set in the Gabor space to eliminate the possible errors caused by the appearance changes. By this way, the accumulated tracking error can be eliminated in time during tracking so that the drifting can be avoided.

However, the above procedure only works when the tracked position does not deviate far from the actual position for each facial feature, otherwise it may fail. Hence, a second component is subsequently activated to impose the shape constraints among the facial features to correct those obvious geometry-violated ones that deviate far from their actual positions. It will be discussed briefly in the following section.

4.1.1 Imposing Geometry Constraints

So far, during tracking, the geometrical relationship among the facial features has not been considered. In order to correct those geometrically violated facial features that deviate far from their actual positions, the geometry constraint among the detected facial features is imposed. However, in practice, the geometry variations among all the twenty-eight facial features under changes in individuals, facial expressions and face orientations are too complicated to be modelled. Therefore, we propose a simple but effective technique to deal with this issue. Basically, the proposed technique is divided into two steps: face pose estimation and geometry constraint imposing.

In the first step, the face pose information is estimated from a set of tracked facial features. Specifically, in order to minimize the facial expression effect, only a set of rigid facial features that do not move significantly under facial expressions are selected, which include four eye corners and three nose points as shown in Figure 1. In addition, the 3D face model composed of these seven facial features is first initialized from a generic 3D face model, which is subsequently adapted to a new individual automatically via the use of a frontal face image. Finally, based on the personalized 3D face model and these seven tracked facial features in a given face image, the face pose can be estimated efficiently under the assumption of weak-perspective projection model.

Given the estimated face pose information, the face pose effect can be eliminated from the tracked twenty-eight 2D facial features. Subsequently, the geometry constraint is imposed on the pose-eliminated 2D facial features to correct the geometrically violated ones. By this way, only the geometry variation under the frontal view needs to be learned, which can be done effectively using a statistical shape model via Principal Component Analysis (PCA) technique.

Specifically, a frontal face shape model is built from a set of face shape samples Q_i extracted from a large number of frontal faces with various facial expressions using the PCA analysis technique, obtaining a mean face shape vector Q_{mean} and a set of k basis face shape vectors $\{Q^j, 1 \leq j \leq k\}$. Usually, the number k is much smaller than the dimension of the face shape vector Q. As a result, given a face shape vector Q_i composed of the twenty-eight tracked facial features from a face image, the global geometry constraint among the facial features can be imposed by $Q_i - Q_{mean} \approx \Phi\mathbf{b}$, with $\Phi = (Q^1, ..., Q^k)$ and $\mathbf{b}$ is a coefficient vector given

by

$$\mathbf{b} = \mathbf{\Phi}^{+}(\mathbf{Q_i} - \mathbf{Q_{mean}}) \quad (3)$$

where Φ^{+} is the pseudo-inverse of the matrix Φ. After the coefficient vector $\mathbf{b}$ is obtained, the geometry-constrained face shape vector $Q_i^{'}$ is represented as $Q_i^{'} = Q_{mean} + \Phi\mathbf{b}$.

Via the proposed method, normally, the obvious geometrically violated facial features that deviate far from their actual positions can be corrected efficiently.

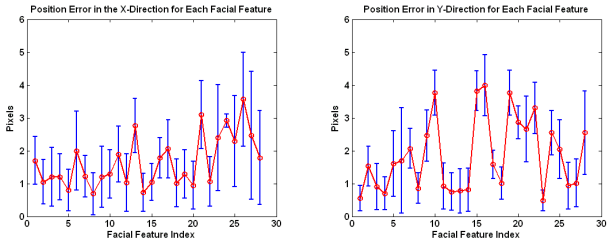
5 Experiment Results

The proposed facial feature detection and tracking technique is implemented using C++ on a PC with a Xeon (TM) 2.80GHz CPU and a 1.00GB RAM. The resolution of the captured images is 320×240 pixels, and the built facial feature tracking system runs at approximately 26 fps. When a person is sitting in front of the camera, it can detect and track twenty-eight facial features automatically. Figure 3 shows the facial feature tracking results on a typical face image sequence that contains significant changes in both facial expression and face orientation.

Fig. 3. The randomly selected face images with tracked facial features.

In order to evaluate the accuracy of the proposed facial feature tracking algorithm, ten face image sequences with significant changes in facial expression and face pose were collected. In each sequence, twenty-eight facial features were detected and tracked automatically by the proposed facial feature tracker. In addition, the positions of these facial features at each frame were also manually located for each sequence, serving as the ground truth during the error computation.

Figure 4 illustrates the computed absolute position errors for the facial features tracked by the proposed facial feature tracker. Specifically, Figure 4 (a) and (b) summarize the computed mean and standard deviation of the absolute position errors for each facial feature in the X-direction and Y-direction respectively, where the mean is represented by the middle value of each error bar and the standard deviation is indicated by the half length of the error bar. In the X-direction, the average mean and standard deviation of the computed position errors for all the twenty-eight facial features is 1.66 pixels and 0.89 pixel, respectively. While in the Y-direction, the average mean and standard deviation of the computed position errors for all the twenty-eight facial features is 1.85 pixels and 0.71 pixel, respectively. Since the original face image resolution is 320×240 pixels, it shows that the position errors of the tracked facial features are small enough for most applications including facial expression recognition and facial animation.



(a) (b)

Fig. 4. The computed position errors for each facial feature: (a) in the X-direction; (b) in the Y-direction

6 Conclusion

In this paper, we present a robust real-time technique to detect and track twenty-eight facial features in the video sequences with significant changes in both facial expression and face orientation. The improvements over the existing facial feature detection and tracking algorithms result from: (1) a dynamic and accurate model updating strategy for each facial feature to eliminate any tracking error accumulation; and (2) an efficient approach of imposing the global geometry constraints among the facial features to eliminate any geometrical violations. With the use of these combinations, the accuracy and robustness of the facial feature tracker reaches a practical acceptable level. Subsequently, based on the tracked facial features, numerous applications including facial expression analysis and face animation can be performed.

References

[1] L. Wiskott, J. M. Fellous, N. Kruger, and C. V. Malsburg, "Face recognition by elastic graph matching," *IEEE Transactions on PAMI*, vol. 19, no. 7, 1997.

[2] Z. Zhang, "Feature-based facial expression recognition: Experiments with a multi-layer perceptron," in *Technical Report INRIA 3354*, 1998.

[3] X. Wei, Z. Zhu, L. Yin, and Q. Ji, "A real-time face tracking and animation system," in *IEEE Workshop on FPIV*, 2004.

[4] M. Burl, T. Leung, and P. Perona, "Face localization via shape statistics," in *International Conference on AFGR*, 1995.

[5] K. Toyama, R. S. Feris, J. Gemmell, and V. Kruger, "Hierarchial wavelet networks for facial feature localization," in *International Conference on AFGR*, 2002.

[6] M. J. Reinders, R. W. Koch, and J. Gerbrands, "Locating facial features in image sequences using neural networks," in *International Conference on AFGR*, 1996.

[7] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," in *ECCV*, 1998.

[8] W. S. Cooper, "Use of optimal estimation theory, in particular the kalman filter, in data analysis and signal processing," *Rev. Sci. Instrument*, vol. 57, no. 11, pp. 2862–2869, 1986.

[9] T. S. Lee, "Image representation using 2D gabor wavelets," *IEEE Transactions on PAMI*, vol. 18, no. 10, pp. 959–971, 1996.

[10] P. Viola and M. Jones, "Robust real-time object detection," *IJCV*, vol. 57, no. 2, pp. 137–154, 2004.