Facial Feature Location with Delaunay Triangulation/Voronoi Diagram Calculation

Yi Xiao[1] and Hong Yan[1,2]

[1]School of Electrical & Information Engineering, University of Sydney, NSW 2006, Australia

[2]Department of Electronic Engineering, City University of Hong Kong, Kowlon, Hong Kong

yix@ee.usyd.edu.au

Abstract

Facial features determination is essential in many applications such as personal identification, 3D face modeling and model based video coding. Fast and accurate facial feature extraction is still a filed to be explored. In this paper, an automatic extracting algorithm is developed to locate "key points" of facial features. The Delaunay Triangulation/Voronoi Diagram technique well known in computational geometric is applied on the edge enhanced binarized facial image. Facial features are classified and extracted in terms of various types of Delaunay triangles and the dual of a subset of the Delaunay triangles, Voronoi edges form the skeleton of facial skin. That is, facial feature's shape is described by Delaunay Triangulation/Voronoi Diagram. Furthermore, the facial features can be identified. The method succeeds in locating facial features in the facial region exactly and is insensitive to face deformation. The method is executable in a reasonably short time.

Keywords: Facial feature extraction, eyes/mouth corners, Voronoi Diagram/Delaunay Triangulation, point set cluster, skeleton

1 Introduction

Facial feature extraction plays an important role in many visualiz applications. In automatic 3D facial model fitting and in facial expression analysis, positions of an individual's face features have to be located in order to express quantitatively the facial features' shape. For an automatic face recognition system, the features or representation of a face are extracted from an input facial image and then compared in a matching process. In a low bit-rate video, location of facial features is the prerequisite for model based image coding. Extracting facial features accurately and efficiently is still a complex issue. The main problems are that facial features vary in shape between individuals and are even highly deformable for one individual; lighting conditions vary for different input images as well.

Researchers have developed a number of methods to extract facial features based on; geometrical face modeling (Jeng et al 1998), feature template matching (Lin and Wu 1999), deformable template matching (Chow and Li 1993), eigenface modeling, elastic graph matching (Liao and Li 2000) and Gabor based complex filtering (Liao and Li 2000, Smeraldi, Carmona and Bigün 2000). Some methods locate feature only roughly by marking the centers of the facial features (Jeng et al 1998 and Smeraldi, Carmona and Bigün 2000). The other methods are devoted to more exacting facial feature extraction (Lin and Wu 1999, Chow and Li 1993 and Liao and Li 2000).

The precise iris and eye or mouth corner locations are given. Among those precise facial feature extraction methods, Chow and Li (2000) propose morphologic filtering and 8-connected blob coloring to find the intensity valley regions for the matching of face template in low resolution. The fine features such as the locations and shapes of eyes or mouth are analyzed in high resolution based on a Hough transform and the deformable template methods. The method can extract the eye boundary with the deformable template. However, the determination of the cutoff (threshold) value for each feature in the template energy cost is a difficult task. Lam and Yan (1996 and 1998) improve the method by detecting the eye and mouth with a deformable template and corner location. After the head boundary is located using a snake model, the eye windows are obtained by means of average anthropometrics. Corners inside the eye windows are then detected and classified based on cost functions. The preprocessing of corner locations avoid incorrect detection of eyebrows as eyes. However, the assumption that all eye corners are in a straight line, is inaccurate when detecting rotated facial features and facial expressions. Liao and Li (2000) describe a facial feature detection based on a Gabor-based complex vector and elastic graph matching. A facial feature is represented by a Gabor-based complex vector, which is obtained by a set of convolutions with a family of complex Gabor filters in order to localize facial features automatically. For a new face image, a data structure,

called a Face Bunch Graph (FBG), is utilized. Collections of face images with different points are marked at correct facial feature points manually. Then a bunch of complex vectors, each derived from a different face image for the same facial feature, is stored stack-like for the corresponding facial feature. The complex vector extracted from a facial feature in the new image will be compared to all complex vectors in the corresponding bunch of the FBG, and the best fitting one will be selected. Most facial features are correctly and accurately detected but if the starting point of a facial feature is not within the correct estimation range, it will be located at a wrong point. An extra adjusting process is needed. Also the whole extraction process is time consuming.

We can see from the above that most of the methods are approached in a bottom-up way, that is, a face is detected by a coarse location of facial features at the first step and fine features are extracted next. The accuracy of the result is affected mainly by two factors; facial features segmentation and flexible facial features classification. Little success has been made so far on the precise location of facial features from a face image. It is difficult to determine the boundary value to extract facial features from the background, and the facial feature identification is affected by rotation, perspective deformation and minor scaling.

This paper focuses on the improvement in facial feature segmentation with Gabor filtering and specific adaptive thresholding method in order to retain precise facial features in a form in which identification is easy. Another contribution to the problem is using Delaunay triangulation/Voronoi diagram computation to classify the feature candidatures, which is suitable for variation of the facial template. In this study, eyes and mouth are selected as the facial features to be recognized because of their generality and salience. Their respective two corners locate these features, as corners are important feature points in shape description and their positions are less susceptible to changes in facial expression and deformation.

The rest of this paper is organized as follows. Section 2 analyses the intensity distribution of typical grayscale facial images, then designs a thresholding scheme to extract eyes, mouths (nostrils in some cases) and separate them from facial skin. For neat segmentation, Gabor-based filtering is adopted to enhance the features edges. Section 3 describes facial features clustering, presented and identified by Delaunay triangulation and the Vorinoi Diagram. Section 4 contains the experiment results for a set of facial images obtained from the ORL database. A conclusion is presented in Section 5.

2 Facial features segmentation

2.1 Intensity distribution of a facial image

Generally, a facial image is evenly illuminated. Intensity within facial skin regions varies slightly. Due to the different color of eyes and facial skin and the shadows cast by eyes, nostrils and mouth over a face, these facial features have a distinct intensity compared to skin. If we take a look at the intensity illustration of a negative grayscale facial image, we will see the ridge like distribution. The peaks are situated in the position of the eye sockets, mouth and nostrils. The ridges are well separated. Their edges reflect the sudden intensity change between eye sockets and skin around them and the shadow between upper lip and lower lip contrasting one from the other, while the facial skin is distributed smoothly and flatly.

Figure 1 shows the intensity distribution of two samples of facial images. The intensity of facial skin dominates the facial region. The pixels in the position of eyes, mouth and nostrils have generally higher intensity values than that for facial skin pixels.

2.2 Local adaptive thresholding

Bernsen's method is an edge sensing thresholding (Bersen 1986 and Verbeek, Vrooman, and Van Vliet 1988) approach, which will highlight regions of the image having high spatial gradients. In his method, a threshold is calculated by statistically examining the intensity values of the window for each pixel. The threshold value *H(i, j)* takes the mean of the minimum and maximum values of the local intensity distribution in the window.

$$H(i,j) = 0.5 * \{ \underset{m=-w}{\overset{w}{MAX}} \underset{n=-l}{\overset{l}{MAX}} I(i+m, j+n) + \underset{m=-w}{\overset{w}{MIN}} \underset{n=-l}{\overset{l}{MIN}} I(i+m, j+n) \}$$

$I(i, j)$ is the intensity for pixel $P(i, j)$. Subtracting the thresholding value from the input, it yields: $d(i, j) = I(i, j) - H(i, j)$. As $d(i, j)$ has a second order derivative behavior, thresholded at zero (Bersen 1986), it will highlight the edge of the image.

There is a relationship between the window size used in a locally adaptive thresholding and the size of the objects of interest in the image (Bersen 1986 and Trier and Jain 1995). Too small a window may cause true foreground pixel be falsely labeled as background. On the other hand, the larger window yields redundancy because of an adverse affect from the illumination gradient. The large window is also more computationally intensive than thresholding using a smaller window. Figure 2 shows the method with different values for *w*. When $w = 4$, most pixels in the

mouth area are misjudged as background pixels, causing the loss of mouth feature. When $w = 20$, the facial features are wrongly enlarged. The method gives a satisfactory result with $w = 6$.

2.3 Ridge enhancement

A simple thresholding technique is often not sufficient to extract all the facial feature pixels as foreground pixels due to the intensity osmosis of the features and the background. To neatly segment the facial features from the facial skin, a restricted version of a Gabor filter is used to enhance ridges that represent facial features.

The filter is specifically designed to give maximal response to ridges at a specific orientation (Jain, Halici, Hayashi, Lee, and Tsutsui 1999). The product of a Gaussian and a consine plane wave gives the impulse response of the filter,

$$f(x,y) = e^{\frac{-(x^2+y^2)}{2\sigma^2}} \cos(k_x x + k_y y)$$

Where σ is the variance of the Gaussian, $\mathbf{k} = [k_x, k_y]^T$ is the wave vector of the plane wave. The main features used in the filter construction are local ridge spacing *s* and local ridge orientation angle θ. The ridge spacing determines both the magnitude of the wave vector and the variance of the Gaussian. The magnitude of the wave vector corresponds to the frequency of the cosine along the wave vector. Given θ and *s*, **k** is determined by the equations below:

$$s = \frac{2\pi}{\sqrt{k_x^2 + k_y^2}} \quad , \quad \tan(\theta) = \frac{-k_x}{k_y}$$

The Gabor filter responses are computed only on a number of orientations in this study, thus imposing a light computational load. As the orientation of facial features is roughly at the horizontal direction, we choose $\theta = 0^0$, $\theta = 45^0$ and $\theta = 135^0$. The choice is sufficient to enhance the edges in eyes, mouth and in some cases, nostril. Figure 3 shows an experimental filtering result in the three orientations. In Figure 3(a), the largest portion of the eye in the horizontal is captured. Figure 3 (b) and (c) extract the curved edge around the left and right eye and eyebrow corners respectively. Combining Figure 3 (a), (b) and (c), Figure 3(d) is obtained. Comparing Figure 3 (d) to Figure 2 (b), We can see that each feature is extracted precisely without fragments and the corners of each feature remain.

With the proposed thresholding after edge enhancement, the facial features are seperated without fragments, but some dark portions like hair, clothing and dimples/pimples on the face might also be included. The unwanted parts are removed by measuring the region's size and position. (See section 4 for detail). Figure 4 is the result of facial features extraction of Figure 3.

The segmentation of facial features from a grayscale image by Gabor filtering and local thresholding can keep exactly the eyes/mouth corner points. However, the result might include a number of other blobs that have a similar size and are close to the facial features. These blobs cause difficulty in locating the eyes/mouth corner points directly from the segmented results using simple local window searching or geometric calculation. Instead, Voronoi diagram/Delaunay triangulation provides an efficient way for analyzing the regions with the help of the nearest neighborhood detection of Delaunay triangulation and the symmetry of the Voronoi diagram.

3 Facial feature detection

Theoretical graph techniques have been extensively used in clustering a planar point set (Urquhart 1982 and Tamura 1982). For clustering, the density of the point set is measured by Delaunay triangle's edge. It reflects the nearest packing neighborhood of any point in the set. Among these clustering methods, the minimum spanning tree, the Gabriel graph and the relative neighborhood graph represent the clusters of the point set by a graph with vertices as the essential points. They are the sub-graph of Delaunay triangulation. Delaunay triangulation describes the clusters of a point set by the connected Delaunay triangles with the points in the set as the triangles' vertices. The clusters are a subset of the Delaunay triangulation. They form planar areas with any pair of points within a cluster close to each other, while any pair of points between two different clusters have a larger distance than the ones within a cluster.

3.1 Delaunay Traingulation for facial feature clustering

In a point set P, its Delaunay triangulation is defined as,

$$DT = \{T(p_i, p_j, p_k) \mid p_i \in P, p_j \in P, p_k \in P, \\ \mathrm{C}(p_i, p_j, p_k) \cap P \setminus (p_i, p_j, p_k) = \phi\}$$

Where $C(p_i, p_j, p_k)$ is the circle circumscribed by the three vertices p_i, p_j, p_k, which form a Delaunay triangle $T(p_i, p_j, p_k)$.

In a well-separated point set, Delaunay triangles can be classified into two types. A *foreground triangle* (F-T) is a Delaunay triangle in a cluster. A *background triangle (B-T)* is a Delaunay triangle outside any cluster. Highly dense points form F-Ts, while sparse points constitute B-Ts. Clear boundaries can be found to differentiate the two type of triangles.

$$F = \{T(p_i, p_j, p_k) \mid \max d(p_i, p_j, p_k) < th\}$$

$$B = \{T(p_i, p_j, p_k) \mid \max d(p_i, p_j, p_k) \geq th\}$$

$d(p_i, p_j, p_k)$ is the distance between any two vertices in $T(p_i, p_j, p_k)$. *th* is the properly decided threshold which yields F-Ts representing patterns of the point set. We have,

$$DT = F \cup B \quad F \cap B = \phi$$

Constructing the Delaunay triangulation on the facial feature candidature, the triangles in different contours can be marked with different ID. Triangles are identified with a threshold that is selected as the distance greater than the minimum geometrical distance between facial features such as eyes, nose and mouth.

Figure 5 is an example of Delaunay triangles classification in a facial image. The facial features consist of a well-separated point pattern in a 2D plane. Edge length criteria in the connected Delaunay Triangulation allows us to identify disjoint regions with F-Ts lying in the feature regions and B-Ts located in the skin areas. Each region is bounded by the connected F-Ts. These connected F-Ts form the regions of two eyes/eyebrows and lip/nose. Facial skin region is an unseparated area; it disjoins each facial feature region. The Delaunay edges shared by B-Ts and F-Ts form the boundary of the facial skin (see Figure 5).

DT/VD analysis allows us to, not only partition the regions but also extract the skeleton of the background area, thereby identifying each facial feature.

3.2 Facial skeleton calculation with DT/VR

Blum and Nagel (1978) introduced the computational analysis of a skeleton. Constructing the DT/VD on the boundary of a shape, the skeleton of shape is the discrete symmetric axis defined by the collection of particular Voronoi diagram points (Fukushima 1997 and Zou. 2001). Each of these points is the center of a circumscribed circle of a Delaunay triangle. We call it a *Delaunay center*. Joining each pair of Delaunay centers corresponding to two adjacent B-Ts, the discrete symmetric axis (skeleton) of the shape is obtained. Each of these pairs is a Voronoi edge. The skeleton is a tree graph. In this discrete discription, each B-T is marked as a *Sleeve-triangle* (S-T), or a *Terminal-triangle* (T-T), or a *Junction-triangle* (J-T) in terms of the triangle having one, two and no edge(s) on the boundary, respectively. An *end point* is the Delaunay center of a T-T, and a *trunk point* is the Delaunay center of a J-T. A branch is a path of joined Delaunay centers starting from an end point and terminating at a trunk point. Pruning all the branches, we obtain the global symmetric axis of the shape. It is the joint of all paths between two trunk points.

Figure 6 shows the skeleton of the sample face.

We refer a facial skeleton as the discrete symmetric axis of facial skin. Connected B-Ts outline the facial skin. It is obtained by linking the Delaunay centers of S-Ts and J-Ts. The topologic structure is slightly different, based on two cases of facial features clustering (see figure 6). Three robust axes will be generated in both cases, 1) an axis (axis 1) between the two eyes, 2) an axis (axis 2) between the left eye and the lip, and 3) an axis (axis 3) between the right eye and the lip.

3.3 Location of feature corners

The axes of the face skin separate each region. The pair of vertices in S-Ts that are the duality of the axes touch the boundaries of two different features. They are called opposite points in Fukushima's paper (1997). Each axis has virtually the same orientation for various faces if they are viewed from the front. In a frontal view face, axis 1 is roughly vertical, while axis 2 and axis 3 are roughly 30^0 apart from the horizontal line.

Based on these analyses, the recognition of each region can be conducted by checking the orientation and size of the S-Ts which connect the eye/eyebrow and lip region. Three such S-Ts are on the boundary of B-T set. We define them as *left triangle (L-T), Right triangle (R-T)* and *Horizontal triangle (H-T).* Their off boundary edges are the symmetric pairs connecting different facial features.

Algorithm for L-T, R-T and H-T searching is as follows:

Input: B-T set B, F-T set F

Output: L-T, R-T, H-T

Begin: L-T and R-T searching

Prune branches by setting the B-Ts associated to a branch as F-Ts;

H= the height of the image;

Search the B,

if (a B-T' non-shortest edge is on the boundary & the B-T is on the right hand side of the image)

if (the angle of the edge is greater than 45^0 & the length of the edge is greater than H/3)

then mark the B-T as R-T;

if (a B-T' non-shortest edge is on the boundary & the B-T is on the left hand side of the image)

if (the angle of the edge is less than 135^0 & the

Once the L-T, R-T and H-T are identified, we can work out the eye/lip contour by searching the regions that connect to

these triangles, thereby locating the feature corners that are in the left/right most of a contour.

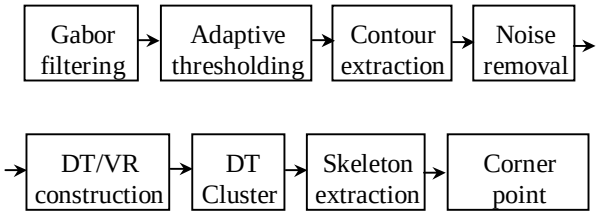
Figure 8 illustrates the L-T, R-T and H-T of Figure 7. The corner points connecting the triangles are marked.

4 Experiment Results and Discussion

The experiment is conducted on the database of the Olivetti Research Laboratory in Cambridge. There are 400 facial images corresponding to 40 persons in the database. Each image is 92×112 with 256 grayscale level. They were taken under slightly different lighting conditions, with different facial expressions, orientations and perspective variations. We are concerned with the front view image without glasses. 304 non-glasses images out of 400 total images were selected.

4.1 Procedure

The procedure for feature corner finding can be summarized as follows:



Contour extraction:

A contour extraction process is performed on the facial feature candidatures to describe the facial feature's shape. Instead of calculating the whole image, only the point in the contour of the facial feature candidatures are examined. Therefore the Delaunay triangulation computation load is reduced greatly.

Noise removal:

Filtering is conducted on the general assumption that in a front viewed, head-and-shoulder image, facial features are located in the middle of the facial image with a specific size. Only the contours having facial features' size around the center position of the face are retained.

The contours with size in the range of R = 0.05 ~ 0.2 and position in $0.2L \le x_c \le 0.8L, 0.7H \le y_c \le 0.1H$ are picked up. R is the ratio of the contour's width l over the whole image's width L or the contour's height h to the whole image's height H. The contour's position is estimated with its center point $c(x_c, y_c)$. Where $x_c = (x_l+x_r)/2$, $y_c = (y_u+y_d)/2$. x_l is the minimum horizontal coordinate of the contour; x_r is the maximum horizontal coordinate of the contour; y_u is the minimum vertical coordinate of the contour; x_d is the maximum vertical coordinate of the region. The width of the contour is $l=x_r-x_l$, and the height of the contour is $h=y_d-y_u$.

4.2 Results and discussion

Figure 9 shows the successful corner point location using the proposed method.

The only assumption of the feature location method is that the eyes come up in pairs and the mouth exists. The method is insensitive to minor scaling, perspective deformations and rotation of an image.

Figure 10 and Figure 11 compare the proposed method with the methods of Lam and Yan and Liao and Li. In figure 10(a), some corner points do not match the real corners because the face rotate, eye corners are not in a straight line. In Figure 11 (a), the left eyebrow point is wrongly located because its initial searching position is not in the estimated area. The proposed method is not affected by these conditions.

The method is not able to handle the following situations:

1. The eye touching the boundary of the face or the hair;
2. The eye joining the eyebrow;
3. The mouth and face being covered by the beard;
4. The face wearing glasses.

The facial feature segmentation affects significantly the correct location rate and computation time. We test correct location rate R using

$$R = \frac{\text{Number of images with correct facial feature location}}{\text{Total number of test image}}$$

.Table 1 tabulates the correct location rate with different image sizes for the same image.

Image size (pixel)	Local ridge spacing S (pixel)	Correct location rate (R)
92×112	7	89%
120×146	9	90%

Table 1: Correct location rate with different resolutions

Table 2 is a comparison between the correct location rate of the proposed method and Lam and Yan's method for the same images.

Method	The proposed method	Lam and Yan's method
Image size (pixel)	92×112	92×112
Correct location rate (R)	89%	85%

Table 2: Correct location rate with different methods

The processing time depends mainly on the Gabor filtering computation time (85%). Figure 10 shows the processing time with different image sizes and s.

We can see from the numerical results that, for the same image, higher resolution gives a higher location rate, but a longer processing time than lower resolution; the proposed method provides more pricise corner location than Lam and Yan's method.

5 Conclusion

Facial images are generally illuminated evenly and facial features such as eyes, mouth and nostrils have higher intensity value compared to skin. They form ridges in the negative grayscale image. Gabor filtering in mainly horizontal orientation enhances the ridges to distinguish the facial features from the facial skin.

Efficient separation of the features from facial skin will make the feature location easy and accurate. Bernsen's method selects an individual threshold for each pixel, based on the maximum and minimum intensity values in their local neighborhood. This allows the neat segmentation of images with edge information.

In a Delaunay triangulation of facial features for a face, the Delaunay triangles have different sizes in different areas. By classifying the size of the Delaunay triangles, the different facial figures are separated into a number of regions. Using with the DT and its dual Voronoi diagram (edges) that represent the skeleton of the skin region, we can find the symmetric points linking the feature corners and therefore, locate each feature.

The proposed method is robust and precise. It is independent of shape, lighting and miner scaling variations. Also, the proposed method is executable within a reasonably short computation time.

6 References

BERNSEN, J. (1986): Dynamic thresholding of gray-level images, *Proc. 8th ICR*, Paris, 1251-1255

BLUM, H. AND NAGEL, R.N (1978): Shape description using weighted symmetric axis features, *Pattern Recognition*, **10**(1): 167-180

CHOW, G. AND LI, X. (1993): Towards a system for automatic facial feature detection. *Pattern Recognition*, **26**(12): 1739-1755

FUKUSHIMA, S. (1997): Division-based analysis of symmetry and its application, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(2):144-148

JAIN, L. C., HALICI, U., HAYASHI, I., LEE, S.B. AND TSUTSUI, S. (1999): *Intelligent biometric techniques in fingerprint and face recognition*, 236. The CRC press. International series on computational intelligence.

JENG, S.H. et al, (1998): Facial feature detection using geometrical face model: an efficient approach. *Pattern Recognition*, **30**(2): 273-282

LAM, K. M. AND YAN, H. (1996): Location and extracting the eye in human face images. *Pattern Recognition*, **29**(25): 771-779

LAM, K.M. AND YAN, H. (1998): An analytic-to-Holistic approach for face recognition based on single frontal view, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(7): 673-686

LIAO, R. AND LI, S.Z. (2000): Face Recognition Based on Multiple Features. Automatic face and gesture recognition, *Proceedings. Fourth IEEE International Conference on*, 34-39

LIN, C.H. AND WU, J.L. (1999): Automatic facial feature extraction by genetic algorithms. *IEEE Transactions On Image Processing*, **8**(6): 834-845

SMERALDI, F., CARMONA, O. AND BIGÜN, J. (2000): Saccadic search with Gabor features applied to eye detection and real-time head tracking, *Image and Vision Computing* **18:** 323-329

TAMURA, A. (1982): Clustering based on multiple paths, *Pattern Recognition*, **15**(6): 477-483

TRIER, O.D., JAIN, A.K., (1995): Goal-directed evaluation of binarization methods, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(12): 47-58

URQUHART, R. (1982): Graph theoretical clustering based on limited neighborhood sets. *Pattern Recognition*, **15**(3): 173-187

VERBEEK, P.W., VROOMAN, H.A. AND VAN VLIET, L.J. (1988): Low-level image processing by max-min filters. *Signal Processing*, **15**(3): 249-258

ZOU, J. J. (2001): Skeletonization of Ribbon-like shapes based on regularity and singularity analyses, *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, **31**(3): 401-407