Glasses Detection and Extraction by Deformable Contour

Zhong Jing & Robert Mariani
RWCP [*], KRDL Multi-Modal Function Laboratory
Kent Ridge Digital Laboratory 21, Heng Mui Keng Terrace Singapore 119613
Email : rmariani@krdl.org.sg, Tel: (+65) 874-8810, Fax: (+65) 774-4990

Abstract

To achieve a face recognition system robust to the presence of glasses, we have developped a glasses detection and extraction algorithm. Detection is realized using edge information within a small area defined between the eyes. Extraction is achieved with a deformable contour, combining edge features such as strength and orientation and geometrical features such as convexity, symmetry, smoothness and continuity. The final position of the deformable contour is obtained using dynamic programming. Experimental results demonstrate the effectiveness of the method, and the robustness to the variation of glasses' shapes and colours.

1. Introduction

Robust face recognition system requires recognising faces correctly under different environment, such as face scale, lighting, hairstyle and wearing glasses. However, the presence of glasses affects the performances of face recognition system. In this paper, we propose a method to detect and to extract the glasses. Detection is similar to the method proposed in [1] and consists in detecting the presence of the bridge of the glasses between the two eyes. Extraction is achieved with a deformable contour [3], combining edge features such as strength and orientation and geometrical features such as convexity, symmetry, smoothness and continuity. The final position of the deformable contour is obtained using dynamic programming [4].

Prior to the glasses detection and extraction, the eyes are automatically detected and the faces have been normalized under scale, rotation and translation, fixing the left and the right eyes at fixed position. The interest area where we research the glasses is defined according to this normalized image. Then, we consider a region R1, shown in Figure 1, in which the glasses are most likely to be located. R1 is further separated into three regions R2, R3, R4. The two regions that cover eyeglass frames, R2 and R3 (the two grey regions including the eyes) are symmetrical by the middle axis of the image. The region R4 (the grey region between the eyes) usually contains the bridge connecting the two glasses parts.

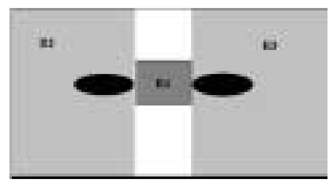


Figure 1. Region of Interest

In the second section, we describe the glasse detection algorithm, and in the third section, we detail the glasses extraction algorithm, before to present the results and to conclude on the perspective of this work.

2. Glasses Detection

There are many kinds of glasses, with different materials, different colours. Furthermore, sometimes there is no frame around the eyeglasses. However, as mentionned in [1], the nosepiece is one of the most common features existing on all of the glasses. It is indispensable and very obvious in any kind of glasses. Using this important feature, face photos with glasses can be detected from facial database.

In [1], the authors introduced six measures in different regions for detecting the presence of glasses. Experiments showed that measure from region R4 alone is the most powerful criterion for glasses detection. The nosepiece can be found in the region R4, where the presence of edges is an important cue. Using a 3x3

*Real World Computing Partnership

sobel edge detector, we obtain at each point (x, y) the magnitude and the orientation of the gradient:

$$G = \sqrt{G_x^2 + G_y^2}\ ,\ \theta = \arctan(G_y/G_x)$$

where G_x and G_y are the horizontal and vertical derivatives obtained at the pixel (x, y). Then, if the number of pixels having a vertical direction exceeds a fixed threshold, the presence of glasses is determined.

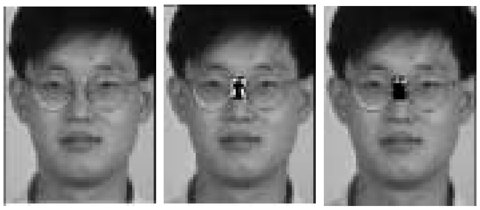


Figure 2. Glasses detection

3. Glasses Extraction

Glasses extraction is realized in three steps. First, we produce an edge map using the Canny edge detector [2] and we filter it by removing the unnecessary edge points. Then, according to the distribution of contour points, we initialize the deformable contour, and finally we optimize its shape and position, by combining image features and geometrical constraints, and, for this specific problem, we find the optimal solution by dynamic programming realizing an exhaustive search over the different possibilities [4].

3.1. Edge Map

The frame of glasses is not always obvious due to the approximation between the grey level of frame and skin in normalised images. So a sensitive detector is expected. For this, we use the Canny edge detector [2] as an optimal operator. There will be broken points in the detected edges, and eyes, nose, and eyebrow will generate disturbing edges.

We can distinguish edges of glasses from edges of eyes and nose simply by different geometrical location. All the edges present in the eyes and nose region are discarded. The edges corresponding to the eyebrows may have conflicting position with the edges glasses and it may be difficult to distinguish them and to eliminate them. The model-based glasses extraction has to deal with this ambiguity.

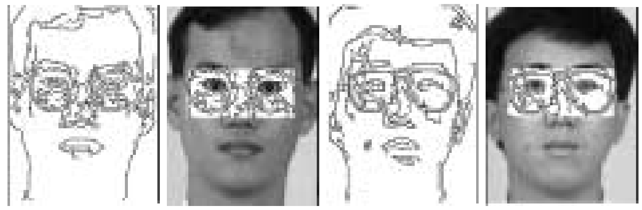


Figure 3. Edge Filtering

3.2. Symmetrical Contour Initialization

The contour of the glasses can be defined as an ordered set of n points $V = [v_1, v_2, ..., v_n]$, where each point v_i is defined on the image grid. As the optimization process is based on dynamic programming, realizing an exhaustive search of all the possible solutions, the location of the initial contour will affect the speed and the computational complexity of the algorithm. Therefore, we need an algorithm to approach the pixels of the glasses contour coarsely.

In order to take into account the natural symmetry of the glasses, and knowing the axis of symmetry of the frontal face, in this case the middle vertical line of the image, we initialize symmetrically the two parts of the contour' glasses, considering only one side, for example the left side, and integrating the edge information present in both sides. In this way, the obtained initialized contour will be symmetric, and later, we will allow only transformation of the contour which keeps the symmetry. The final contour will therefore be symmetric by construction.

Concretely, let (x_l, y_l) the coordinate of the left eye center, and the ith angular sector S_i defined by $[\theta_i, \theta_i + \Delta\theta]$ and centered in (x_l, y_l). We are interested in those edges points which are defined within an acceptable distance interval from the eye's center $[\rho_{min}, \rho_{max}]$. We obtain the sample point $v_i = (u_i, v_i)$ associated to this sector S_i as follow

$$\begin{array}{rcl} \theta_i & = & \frac{2\pi i}{\Delta\theta} \\ u_i & = & \frac{1}{2N}\sum_{j=1}^{N} x_j * (E(x_j, y_j) + E(2c - x_j, y_j)) \\ v_i & = & \frac{1}{2N}\sum_{j=1}^{N} y_j * (E(x_j, y_j) + E(2c - x_j, y_j)) \end{array}$$

where $E(x, y) = 1$ if the point (x,y) is an edge pixel, and 0 otherwise, $X = c$ is the middle vertical line of the image and used as symmetrical axis, and N is the number of edges pixels considered $(N = |(x, y) \in S_i : E(x, y) = 1| + |(x, y) \in S_i : E(2c - x, y) = 1|$. If no edge pixels are present within this sector, the initial point is taken as the middle of the sector. $\Delta\theta$ fixes the size of one sector and we then obtain $n = \frac{2\pi}{\Delta\theta}$ points. This computation is realized for each sector.

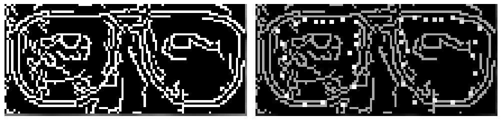


Figure 4. Initial Contour

3.3. Directed Graph Construction

Having the initial contour, we construct the directed graph $G = (H, T)$ of hypothesis by considering all the possible positions of the points and the edges linking them, and by attributing quality costs to each hypothesis and to the geometrical likelihood of a transition between two consecutive hypothesis. At the end, we obtain the optimal solution by computing the shortest path linking the two extremity of this graph. Here, H denotes the set of all the hypothesis and T denotes the set of the transitions between two hypothesis. Again, the symmetry of the contour is embedded in the model, and only one real contour is modified, the other one being obtained by vertical symmetry about the middle line $X = c$. To do so, we have incorporated into the hypothesis quality cost, a measure which is a combination of two image observations.

Let $P = \{P_1, P_2, ..., P_n\}$ the set of possible position for each of the n points of the initial contour, where $P_i = \{p_{-k}, p_{-k-1}, ..., p_0, ..., p_{k-1}, p_k\}$, and where $p_0 = (\rho_0, \theta_0)$ corresponds to initial ith point and $p_m = (m\Delta\rho + \rho_0, \theta_0)$ is the initial point placed at the mth position where k and $\Delta\rho$ are two fixed values.

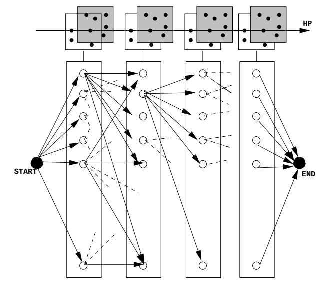
From this set of possible positions of points, we derive the set of possible segments, which constitutes the hypothesis set H, and the set of transitions between two hypothesis T

$$H = \{start, H_1, H_2, ..., H_n, end\} \ , \ H_i = P_i \times P_{(i+1)\%n}$$
$$T = \{T_{start}, T_1, ..., T_{n-1}, T_{end}\} \ , \ T_i = H_i \times H_{i+1}$$
$$T_{start} = \{start\} \times H_1 \ , \ T_{end} = H_n \times \{end\}$$

where $start$ and end are two virtual vertices denoting respectively the first and the last node of the graph G, $\times$ is the cartesian product and $\%$ is the modulo operator.

3.4. Cost Function of a Hypothesis

The quality of a hypothesis $h_i \in H_i$, where $h_i = [p, q]$ and p and q are two possible positions of the points



(d)

Figure 5. Dynamic programming

of the deformable contour, is defined using a similarity between the observed direction of contour and the expected one, and makes use directly of the symmetry constraint, by combining the observations on $[pq]$ and $[rs]$, the symmetric segment of $[pq]$.

The cost function from the edge information is produced according to the magnitude and the direction of the gradient. A comparison of the expected gradient direction with the observed gradient provides this cost. With θ, the direction normal to the segment $[pq]$, and $\theta(u)$ the observed gradient at the pixel u, the edge cost is obtained as follow:

$$Cost([pq]) = 1 - \frac{|\{u \in [pq] : ||\theta(u)| - |\theta|| \leq \epsilon\}|}{\|\overrightarrow{pq}\|}$$

where $|X|$ denotes the cardinal of the ensemble X, u is a pixel situated on the segment $[pq]$ and ϵ is an angular tolerance. When the matching is perfect, this cost is equal to zero and in the worst case it is equal to one. Then, the hypothesis cost is defined as

$$C(h_i) = \frac{1}{2}(Cost([pq]) + Cost([rs]))$$

where $[rs]$ is the symmetric segment of $[pq]$ about the vertical middle line $X = c$. This cost is defined between 0 and 1.

3.5. Transition Cost between Two Hypotheses

Let $h_i = [p, q]$ and $h_{i+1} = [q, r]$, two consecutive hypothesis, linked at the point q. The cost associated to the transition $t_i = (h_i, h_{i+1})$ is defined using a smoothness measure associated to the chain $[p, q, r]$, and which comes from the fact that the shape of the glass is generally convex and with a piecewise smooth contour, the ideal example of glass being the ellipsis. Moreover, a transition is declared impossible when it creates a non-convex contour or when it is discontinuous, ie, we do not consider transitions between a hypothesis $[pq]$ and $[rs]$ when $q \neq r$.

The transition cost is then given by

$$T(h_i, h_{i+1}) = \begin{cases} \infty \ \textit{if not convex} \\ \frac{1}{2}(1 - \cos\theta), \ , \ \theta = \widehat{\overrightarrow{pq}, \overrightarrow{qr}} \end{cases}$$

where θ is the bending angle. Convexity of the chain $[pqr]$ is established using the center of the eye (x_l, y_l).

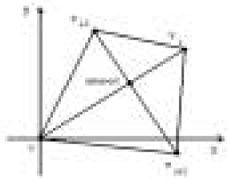


Figure 6. Convexity

3.6. Optimization

Once the elementary costs of hypothesis and transitions have been established, we normalize the graph by associating to each transition, a final cost G, linear combination of the elementary transition cost and the following hypothesis cost, ie

$$G(h_i, h_{i+1}) = \lambda C(h_i) + (1 - \lambda) T(h_i, h_{i+1})$$

where λ has been fixed experimentally. Then, we compute the shortest path linking the vertex *start* to the vertex *end*, and this path defines the final and optimal contour.

4. Results

We have tested our glasses detection algorithm with a set of facial images. This set contains photos of 419 people, among which 151 people wear glasses and these images are taken under different imaging conditions. All the face images are frontal view, however, some of them are taken with slight face slant and orientation. The glasses that appeared on the face images have various shapes and colours, so that the robustness to the different variations of the method can be evaluated.

The correct detection rate is 99.52%. We obtained two false alarms in this test, by falsely detecting the presence of glasses in facial images. 50% glasses are accurately extracted from the database, 30% of glasses are extracted with satisfactory results, and the remaining 20% are obtained with fair results. In failed face images, the gray scale of face skin is very similar to the grey-scale of glasses; furthermore, eyebrows cross edges of glasses, creating substantial ambiguities.

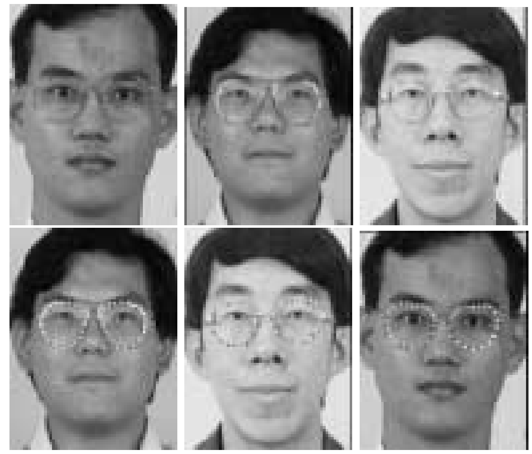


Figure 7. Results

5. Conclusion and Future Work

We have proposed a glasses detection and extraction algorithm based on deformable contours. The optimization of the final position of the contour was established by dynamic programming, realizing an exhaustive but limited search of the optimal solution, around an initial contour, which is close to final solution. Here, the main features we used were the symmetry, the convexity and the continuity of the contour image, combined with edge orientation information. Results obtained on facial frontal images are satisfactory, and now we are investigating the extraction of glasses in the presence of pose, using a perspective symmetry.

References

[1] X. Jiang, M. Binkert, B. Achermann, H. Bunke, "Towards Detection of Glasses in Facial Images", *In Proc. 14th International Conference on Pattern Recognition*, Brisbane, 1071-1073, 1998.

[2] J. Canny, "A Computational Approach to Edge Detection", *IEEE Trans. on Patt. Anal. and Mach. Intell. 8, 6*, 1986.

[3] Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models", *Proc. Int'l Conf. Computer Vision*, pp. 259-268, London, England, June 1987.

[4] D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos, "Dynamic Programming for Detecting, Tracking, and Matching Deformable Contour", *IEEE Trans. on Patt. Anal and Mach. Intell. 17, 3*, pp. 294-302, 1995.