DETECTION OF SIDE-VIEW FACES IN COLOR IMAGES

Gang Wei¹, Dongge Li¹ and Ishwar K. Sethi²
¹Department of Computer Science
Wayne State University, Detroit, MI 48202
²Intelligent Information Engineering Laboratory
Department of Computer Science and Engineering
Oakland University, Rochester, MI 48309-4478

Abstract

A coarse-to-fine scheme for the detection of side-view faces in color images is proposed in this paper, which extends the current state of the art of face detection research. The input image can be of complex scene, containing cluttered background and confusing objects like frontal-view faces. The system consists of four stages, each of which is a refinement of the previous one, namely: 1) skin-tone detection by color, 2) region and edge preprocessing with morphological operations and length filtering, 3) face candidate region selection based on normalized similarity value and 4) final verification using Hidden Markov Models. Encouraging experimental results have been obtained, due to the utilization of multiple features of the input image and the conjunction of employment of various image processing and pattern recognition techniques. The paper has several original designs besides being able to detect faces other than frontal-view, including the Normalized Similarity Value (NSV) to detect the presence of a given curve pattern, the Iterative Partition Process to segment the object from confusing extraneous regions for higher detection accuracy and the exploration of the use of HMM to recognize objects in images.

1. Introduction

The problem of face detection in images - locating image areas corresponding to human faces - has received a considerable amount of attention in recent years due to numerous possible applications in practice, including those in person identification, surveillance, and image/video annotation. A variety of face detection methods has been reported in the literature and a comprehensive review can be found in [2]. While many of the earlier methods were developed to detect faces in gray scale images, most of the recent methods, however, work on color images. Despite the large number of methods proposed in the literature for face detection, the performance of these methods remains unsatisfactory for images containing side-view faces. This probably stems from the fact that people usually focus on frontal-view faces in interpreting video or image contents and the detection of side-view faces are neglected in most research work. While in applications such as access control via face identification, it is reasonable to look for a front-view face only, applications such as surveillance, criminal identification in a mug shot database, and image/video annotation would greatly benefit by being able to detect side-view faces.

In this paper, we propose a coarse-to-fine scheme to detect side-view faces in color images to extend the state of the art of face detection research. The proposed scheme is designed to complement our frontal-face detection system [11]. For a given color image of arbitrarily complex scene, the purpose of the system is to locate only side-view faces as in Fig. 1. An omni-face detection system capable of finding human faces from various views can be obtained by integrating the frontal-view and side-view face detection systems.

Unlike front-view faces, which have an oval shape, it is hard to mathematically model the shapes of side-view faces. Some characteristics of side-view faces have been analyzed in previous research on person identification such as [5] and [10]. It has been observed that the most distinguishable feature of side-view faces is the protrusion of the nose in the face region contour. This naturally leads to the attempt to model the side-view face profiles using curve representation methods like B-spline as in [1] or [8] and use it as the filter to select matching curves in the edge map of the image. In practice, however, the face profiles can not be perfectly extracted due to the limitation of existing edge detection techniques. As illustrated in Fig. 1 (d), the face profile could be broken into many segments or connected to extraneous edges because of environmental factors like noises,
shading and occlusion, ending up that no single edge segment satisfies the selection criteria. To circumvent this difficulty, we developed the Normalized Similarity Value (NSV) to find groups of edge segments that resemble side-view face profiles when considered together. It does not require explicit correspondence between pixels in the template and the image and is especially suitable for this problem where the scene could be highly cluttered.

Working on multiple image features, our system is stable and accurate owing to the employment of the novel components besides the NSV, including the Iterative Partition Process and the exploration of Hidden Markov Model for recognizing objects in 2D image. The remainder of this paper is organized as follows. Section 2 presents the detailed description of the four stages of the system. In Section 3 we give examples of the experimental results and evaluate the performance of the system. Conclusion and possible future extension of the system are discussed in Section 4.

2. System description

The system follows a coarse-to-fine strategy in that each of the four stages is a refinement of the previous one. A stage makes a hypothesis about the position and size of side-view faces based on certain image features. The hypotheses are either verified or rejected in the next stage until the faces are located in the final stage. Those four stages are 1) skin-tone region extraction by color, 2) region and edge preprocessing with morphological operations and length filtering, 3) face candidate region selection based curve pattern detection with Normalized Similarity Value and 4) final verification using Hidden Markov Models.

2.1 Skin-tone region segmentation

The purpose of the first stage is to reduce the pixel search space. Human faces have skin-tone color and thus by extracting those regions we can focus our attention only to them instead of the whole image. YIQ color coordinate is used to perform the segmentation at pixel level. By manually marking the skin-tone regions in a large collection of image as the training set, we observed that skin-tone pixels reside in a half ellipse in the Y-I plane as shown in Fig. 2. The distribution of skin-tone pixels is modeled by the half ellipse, which is applied to the input color image to classify each pixel as being skin-tone or non-skin-tone. The output of this operation is a binary image where the white pixels have skin-tone in the original image, which is illustrated in Fig. 1 (a) and (b). This stage provides the coarsest estimation of side-view faces because only color information is considered and many false alarms may be generated such as the frontal-view face and non-face objects in Fig. 1 (b).

2.2 Region and edge preprocessing

The candidate regions of side-view faces are selected based on an analysis of the shapes and edges of the regions. Due to various factors such as noise, shading and illumination variations, skin-tone regions or edges can not be perfectly extracted. Therefore it is necessary to eliminate the effect of above factors before proceeding to the next stage. Small and isolated region are usually unlikely to be faces and it is desirable to remove them in this early stage instead of later stages, which require far more computation overhead. Morphological operations are used to accomplish this. We applied the Open operation to the skin-tone binary image to remove small isolated regions. It can also break weak connections as well as erase thin protrusion [4]. This feature is important because face regions may be connected to other objects having skin-tone, which introduces complexity in later analysis, and breaking them apart in advance can minimize the adverse effect. To get optimal results, the size of the structuring element of the open operation is adaptive to the size of the image while constrained by an upper and a lower limit. Fig. 1 (c) shows the result of this process.

Edge detection is applied to the image and the result is shown in Fig. 1 (d). Since the contours of face also are in skin-tone, it is sufficient to just inspect skin-tone edges. To extract those edges, the pixel-wise intersection operation is performed between the edge map like in Fig. 1 (d) and Fig. 1 (c), the preprocessed skin-tone regions and Fig. 1 (e) shows the result. Very short edge segments are usually due to noise and minor variations and we use an edge-length filter to remove short edge segments. The result in a cleaner edge map is shown in Fig. 1 (f).

2.3 Face candidate selection

In this stage face candidates are selected based on the analysis of each region. Two side-view
face profile templates are constructed as in Fig. 3, for left-oriented and right-oriented, respectively. If a region contains edge curves similar to one of the templates, then that region is considered as a candidate. Due to extraneous factors, the contour of a face is often broken into several edge segments and thus no single segment resembles the templates. Instead, groups of edge segments need to be checked to see if they satisfy the criteria. However, exhaustively considering all possible combinations of edge segments is impractical. In our system, this difficulty is circumvented by the Normalized Similarity Value (NSV) measure, which addresses the problem of finding matches for a designated pattern (template) in the image. NSV is relatively insensitive to noise and does not require explicit correspondence between the points on the pattern and the image. These two characteristics make it especially suitable for our application environment, where the scene could be very complex and the curve pattern to be detected can not be defined mathematically.

To determine if a region should be accepted as a candidate, its NSV’s to the templates are computed. If the highest NSV is higher than a threshold, the region is accepted as a face candidate, otherwise the Iterative Partition Process is applied to see if it can be divided into smaller candidates until there are no more potential face candidates.

2.3.1 Normalized Similarity Value

NSV is essentially a variation of Directed Hausdorff Distance discussed in [6] and [7]. Thus before introducing our approach, it is useful to take a brief review of Hausdorff Distance, a measure for the similarity between two finite point sets. Given sets A and B, Hausdorff distance is defined as

\[ H(A, B) = \max(h(A, B), h(B, A)) \]  

(1)

where \( h(A, B) \) is called directed distance from A to B. For all points in A, calculate the distances to their respective nearest neighbors in B, and \( h(A, B) \) is the largest value. The definition of \( h(A, B) \) is given below:

\[ h(A, B) = \max_{a \in A} \min_{b \in B} ||a - b|| \]  

(2)

\( ||.|| \) is some norm in the plane. \( h(B, A) \) is the directed distance from B to A, which is defined similarly. The undirected \( H(A, B) \) is small when both \( h(A, B) \) and \( h(B, A) \) are small. If \( H(A, B)=d \), then every point in A is within distance \( d \) of some (note the term some is used since no explicit pairing between points in A and B is required) point in B, and vice versa. Thus Hausdorff distance can be used to evaluate the mismatch (and similarity) between A and B. However, this measure is vulnerable to noise since a single point far away may cause \( H(A, B) \) very large. A modification called partial Hausdorff distance is used to solve this problem, which is defined as

\[ H_{hk}(A, B) = \max(h_k(A, B), h_k(B, A)) \]  

(3)

where the \( h_k(A, B) \) and \( h_k(B, A) \) are the partial directed distances. By ranking all points in A by their distance to the nearest match in B, \( h_k(A, B) \) is the \( L_k \) largest distance (\( 1 \leq L_k \leq q \), where \( q=|A| \)). The definition of \( h_k(A, B) \) is:

\[ h_k(A, B) = \min_{a \in A} \left( \min_{b \in B} ||a - b|| \right) \]  

(4)

and \( h_k(B, A) \) is similarly defined. Thus if \( H_{hk}(A, B)=d \), then there are \( L \) points in A that are within distance \( d \) of some point in B and vice versa. By truncating the contributions of maximum values, the influence of the noise is effectively reduced.

We decide if a region is a candidate by checking if there are any edges that resemble the side-view face templates well. Therefore, when evaluating the similarity, only the distance from the template to the image (forward distance) is considered. The following measure \( s \) for NSV is used to see how similar the points on an edge segment map A are to the template B:

\[ s = \text{card} \left( \{ b \in B \mid \left( \min_{a \in A} \|a - b\| < \Gamma_s \} \right) / N \right) \]  

(5)

The quantity \( N \) is the number of points in B and \( \text{card}(\cdot) \) is the cardinality of the set. The value of \( s \) ranges from 0 to 1, which indicates the fraction of points in B that are within distance \( \Gamma_s \) to some points in A. Thus the larger \( s \) is, the better match we can find in A for B. When \( s=1 \), it means that every point in B is within distance \( \Gamma_s \) to its nearest neighbor in A and a perfect match is found. In implementation, the value of \( \Gamma_s \) is adaptive to the size of the template and ranges from 2 to 6 pixels.
2.3.2 Comparison between a region and the templates

We compare each region against the two templates to decide if it is accepted as a candidate. The following processes are applied.

- Compute its similarity to the left and right template ($S_L$ and $S_R$, respectively)
- Take the larger one as the final similarity ($S$). That is, $S=\max(S_L, S_R)$. If $S$ is less than a threshold $\Gamma_s$, the region is put into the rejected list and subject to an iterative process.
- Otherwise the region is accepted. If $S_L$ is greater than $S_R$, the region is more like a left-oriented face than a right-oriented face and thus annotated as a left candidate. If $S_L$ is less than $S_R$, it is annotated as right candidate. If they are equal, an arbitrary decision is made (however this is very unlikely to happen)

To calculate the $S_R$ of a given region, the right face template is scaled to the same height as the region and aligned with the region vertically as in Fig. 4. Then the template is translated horizontally pixel by pixel from the leftmost point to the rightmost point of the region. At each position, the Normalize Similarity Value ($s_{ri}$) between the template and the cleaned edge map (like Fig. 2 (f)) in the region is calculated. Thus we have a series of NSV's $S_R$ is defined as

$$S_R = \max (s_{r1}, s_{r2}, \ldots, s_{rn})$$

And $S_L$ is computed in the same approach with the left face template. By comparing the values of $S_L$ and $S_R$, we can decide if the region should be accepted and further annotate the candidates as left or right oriented.

A straightforward implementation of the above process is inefficient because it involves complex and unnecessary computation. First, repeatedly finding the nearest neighbor in the image for a point on the template is time-consuming. We used distance transformation as discussed in [3] to solve this problem. Second, at most translation positions, the NSV’s are less than $\Gamma_s$, which will have no impact on the decision and annotation results and calculating the exact values on those positions is meaningless. Therefore, it is desirable to tell the NSV of current position is less than $\Gamma_s$ before fully calculating is finished. According to the definition of NSV, if the NSV at a position is greater than $\Gamma_s$, it follows that there are at least $\Gamma_s^* \Gamma_s$ points in $B$ within distance $\Gamma_d$ of some point in $A$. On the other hand, if there are more than $\Gamma_s^* (1-\Gamma_s)$ points in $B$ that are away from any point of $A$, NSV is less than $\Gamma_d$. Based on this analysis, we define a counter $C$, which is initialized to 0 and incremented by 1 each time a point in $B$ in found to be more than $\Gamma_d$ away from any point in $A$. When $C$ grows greater than $\Gamma_s^* (1-\Gamma_s)$, the computation at current position is aborted and NSV is assigned with any value less than $\Gamma_s$. Then the template is translated to the next position for the same process. By doing this, we can avoid a lot of unnecessary computation overhead and improve the efficiency significantly.

Discarding the regions that are rejected in the template comparison can lead to low detection rates because the face region may be connected to another skin-tone region. For example, in Fig. 1 (c) the woman's face region was rejected in the first round of template comparison because the extension of her neck caused an improper scaling of the template, which is illustrated in Fig. 4. We designed an Iterative Partition Process to solve this problem. The rejected regions are divided into smaller sub-regions to see if new face candidates can be found. A similar algorithm is discussed in [11]. However, the objective in [11] is to find frontal-view face candidates and therefore a clustering partition is used to generate convex sub-regions. In this application, regions are split vertically into several (the empirical number is 3) parts of uniform width like Fig. 4 so that the frontal part of the side-view face can be separated from the rest part of the region. The divided smaller regions undergo the same template comparison as described above. This process is applied iteratively until no more face candidates can be found. Fig. 1 (g) illustrates that the frontal part of the woman's face was detached from the neck and picked up in the Iterative Partition Process. On the binary image there are also other two small regions that are falsely accepted as face candidates. These candidates are subject to further verification in the final stage.

2.3.3 Final verification with Hidden Markov Model

The final verification stage determines if a candidate region is indeed a face by inspecting the intensity patterns within the region. Facial features like eyes and eyebrows are usually
darker than the rest of the face, which motivated us to divide the candidate region into a number of horizontal stripes of uniform height and check which stripes have higher average intensity than others. However, manually identifying those patterns unique to faces is tedious and hard due to the variation of different person’s face and the complication caused by other factors like expression and hairstyle. By the use of Hidden Markov Model, we enabled the system to automatically capture patterns corresponding to faces, which solves the difficulty elegantly.

Hidden Markov Model (HMM) is a popular technique widely used in signal processing and is particularly powerful in recognizing time-series patterns. The essence of HMM is to construct a model to explain the occurrence of observations (symbols) and use it to identify other observations. [9] presents the fundamentals of HMM and its applications. We used a collection of face candidate regions containing both real-face regions and false positive regions as the training set. Each region is partitioned into a number of horizontal stripes (in implementation the number is 16) and the average intensity values of the stripes are used as the observation sequence to the HMM. Based on these observation sequences, we trained two HMM’s for face and non-face, respectively. Thus for a given candidate region, the same types of patterns are extracted and used as the observation sequence to the two HMM’s. If the face-HMM generates the larger response the region is labeled as a face, otherwise it is rejected. The final result of the system is shown in Fig.1 (h).

3. Experimental results

Our system has been tested with 130 images representing varying degrees of difficulty in terms of different factors of the problem, e.g., background skin-tone objects, partial occlusion and various scales. The images contain 60 frontal faces and 85 side-view faces. 60 of the side-view faces are correctly. 15 false positives are generated, 7 of which are caused by frontal-view faces. The performance is encouraging considering the complexity of the test images we used and that the recall and precision is comparable to the state of the art of frontal-view face detection systems. More results are presented in Fig. 5. We observed that false positive are often found in skin-tone regions rich in texture whereas false negatives are usually faces of which the nose protrusions are not obvious or the face is enclosed by other skin-tone objects.

4. Summary and future work

Our work was developed to extend the state of the art of face detection research by providing the ability to detect side-view faces, which is not supported by existing systems. The proposed system works in a top-down fashion on color images. Each of the four stages generates hypotheses and seeks evidences to verify them. The hypotheses correspond to face candidate regions and the evidences are essentially particular image features. Unsupported regions are rejected while accepted ones are subject to refinement in the next stage until the faces are finally located.

The system contains three original components. The first is the design of NSV to detect the presence of a given curve pattern, which is especially useful when the pattern cannot be model mathematically. It does not require explicit correspondence between pixels in the template and the image, which makes it very suitable in detecting patterns in complex scenes. The Iterative Partition Process is another novel part of our scheme to improve the detection. This process allows us to minimize falsely rejected regions by detaching candidate regions connected to extraneous regions. Finally we explored the use of HMM to recognize objects in image. HMM is a popular technique to identify time-series patterns and our experiment shows that it also works well in 2D images.

Further refinements on each stage will be made to improve the performance in terms of precision, recall and efficiency. We are in the process of developing a module for the final verification stage using time-wrapping technique and compare its performance with the current HMM method. The frontal-view face detection system in [11] will be incorporated to obtain an omni-face detector capable of detecting faces irrespective of pose. We have extended the frontal-view detection system for face tracking in video, and by adding the side-view face detection ability, the tracking power will be greatly reinforced.

Reference:

1. T.J. Cham and R. Cipolla (1999). Automated B-spline Curve Representation


---

**Fig. 1 Example of the detection process**

(a) Original Image  (b) Skin-tone regions  (c) Preprocessed regions

(d) Edge Map  (e) Skin-tone edges  (f) Cleaned skin-tone edges

(g) Face Candidates  (h) Detection result
Figure 2. The distribution of skin-tone pixels in Y-I plane.

Figure 3. Side-view face templates

Figure 4. Illustration of iterative region splitting for the side-view face candidate selection

Fig. 5: More examples