

Fast Face Detection Using QuadTree based Color Analysis and Support Vector Verification

Shu-Fai WONG and Kwan-Yee Kenneth WONG

Department of Computer Science and Information Systems
The University of Hong Kong, Hong Kong
{sfwong, kykwong}@ccsis.hku.hk

Key words: Object Detection, Pattern Recognition, Color Analysis

Abstract. Face detection has potential applications in a wide range of commercial products such as automatic face recognition system. Commonly used face detection algorithms can extract faces from images accurately and reliably, but they often take a long time to finish the detection process. Recently, there is an increasing demand of real time face detection algorithm in applications like video surveillance system. This paper aims at proposing a multi-scale face detection scheme using Quadtree so that the time complexity of the face detection process can be reduced. By performing analysis from coarse to fine scales, the proposed scheme uses skin color as a heuristic feature, and support vector machine as a verification tool to detect face. Experimental results show that the proposed scheme can detect faces from images reliably and quickly.

1 Introduction

Face detection is one of the hot topics in pattern recognition. It receives much attention mainly because of its wide range of applications. Intelligent video surveillance system, reliable human-computer interface, and automatic formation of face database are some examples of commercial and industrial applications using face detection as a major component.

Commonly used face detection algorithms can detect face accurately. A comprehensive survey on face detection can be found in [1], [2]. Most of these detection algorithms perform a brute force search of face patterns in the image. In other words, pattern recognition has to be performed at different scales and positions within the image. Pattern recognition techniques such as neural networks [3], gabor filters [4], and Markov random field [5] have been used in face detection recently. The major problem of such kinds of algorithms is that it is time consuming and hence cannot be applied to a real time system.

To reduce the time complexity, researchers start investigating the use of visual cues to facilitate the searching process. Motion, color, and configuration of facial features have been used as visual cues. Among these cues, color is most widely used [6], [7], [8]. Skin color can be learnt from face database. By performing appropriate color-space transformation, the projection of skin colors on such a color-space will form a cluster. Potential face region can be identified

by comparing the distance between the projection of the pixel's intensity and the center of the cluster in that color space. However, the time complexity of this approach is still high. This is because time consuming morphological analysis is usually done before face verification. There is an alternative that replacing the whole pattern recognition step by the visual cues detection so that detection process can be fastened a lot. However, the accuracy will drop dramatically, especially when there is a complex and distracting background.

As described above, detecting faces in image reliably and quickly is a difficult problem. This paper aims at proposing a multi-scale analysis scheme for face detection. Our idea is similar to that presented by Sahbi et. al. [9]. However, they haven't explained explicitly how multi-scale analysis is done and only illustrated a simple and unreliable verification scheme based on histogram. In our work, an explicit implementation details will be explored and in addition, speed up methods and reliable verification scheme are proposed. Under our proposed scheme, multi-scale analysis using Quadtree and wavelet decomposition will be performed on the input image. Skin color will be detected starting from low resolution images. Face verification using support vector machine (SVM) and wavelet coefficient as input will be done once skin color region is reported. Experimental results show that the scheme can detect face efficiently and reliably.

The overview of the proposed scheme will be covered in Section 2. The implementation details will be explored in Section 3, 4 and 5. Experimental results will be shown in Section 6.

2 System Overview - QuadTree based searching

The proposed system consists of three major components, namely the skin color detection module, the wavelet transformation module and the support vector verification module. The skin color detection module will extract skin color region from the image. The wavelet transformation module will transform the image into an image pyramid that consists of images at different resolutions and then form a primitive Quadtree for efficient skin color searching. The support vector verification module will determine whether the input pattern is a face pattern. The work flow of the system is shown in Algorithm 1. In general, by starting analysis from lowest resolution and by limiting the range of resolution of the image to be analysed, the total number of pixels to be analysed is small and the algorithm is thus fast. In addition, no further morphological analysis is needed because coarse-to-fine analysis using Quadtree structure is indeed performing image smoothing and noise filtering implicitly. Although it seems that the wavelet decomposition step causes serious computational load compare with simple blurring in order to generate image pyramid, the wavelet coefficients calculated can be used in the verification step which increases the reliability of the system. The system is supposed to detect face reliably in reasonable time.

Algorithm 1 Proposed Quadtree based face detection algorithm

```

Form a pyramidal image structure by performing discrete wavelet transform.
Form a Quadtree with depth 1 using the lowest resolution image as root.
Detect skin color in the image at depth level 0 in the Quadtree.
repeat
  Expand the current Quadtree leaf nodes where no skin color detected. Expansion is done by
  forming new leaf nodes at current node using higher resolution image patches from the pyra-
  midal image.
  Detect skin color in the image at newly expanded nodes.
until (The current depth of Quadtree  $\geq$  The depth level of Quadtree where skin color is firstly found
+ finite bound on search level) or (the image at current node is at highest resolution)
Form binary image by reprojecting the skin color nodes onto a single image.
for all skin regions indicated in the binary image do
  Compare pattern within the region with facial patterns using SVM
  if face pattern confirmed then
    Report and remove the face region
  end if
end for

```

3 Color Analysis

According to recent research, skin color tends to form cluster in different color space [10]. In the proposed system, color will be used as the only heuristic feature. Skin color pixel can be identified by comparing the given pixel with the skin color distribution or model.

Skin color model is learnt in offline. Face images taken from office environment under different lighting condition are used in skin color learning. Skin pixels are separated manually. Those skin color pixels extracted will be converted from RGB color space to normalized rg-color space, through $r = R/(R + G + B)$ and $g = G/(R + G + B)$. A histogram is then formed from those manually selected skin pixel in rg-color space. The histograms obtained in this step is illustrated in figure 1. As indicated in the figure, the distribution of skin color is quite compact and thus it can be approximated by Gaussian distribution ($M \sim N(\mu_M, \Sigma_M)$), where M is skin color model, μ_M and Σ_M are the mean and covariance of the skin color distribution. Thus, $P(\text{skin}|r, g, M)$ can be approximated by the mahalanobis distance of certain rg-value from the distribution of the model M , where r, g is the normalized rg-value of a given testing pixel. Detected region can be reported by thresholding those pixel with high $P(\text{skin}|r, g, M)$ value.

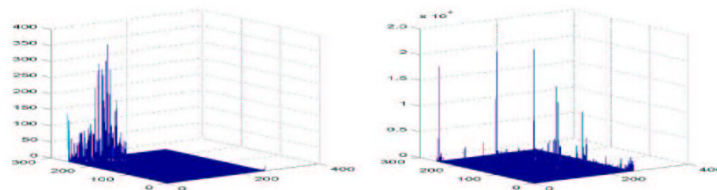


Fig. 1. The histogram on the left shows the distribution of normalized rg-value of skin pixels while the histogram on the right shows those distribution of the non-skin pixels.

4 Multi-Scale Wavelet Analysis

Image can be broken down into constituent images with different resolution through wavelet transform and an image pyramid can then be formed. Under Quadtree based searching scheme, color analysis can start from the tip of the pyramid (images with lowest resolution) downwards to the bottom of the pyramid (images with highest resolution). Information from analysis at a lower resolution can be used in analysis at a higher resolution. Analysis at a lower resolution can thus determine whether it is necessary to explore certain set of pixels at a higher resolution. Time can hence be saved by avoiding unnecessary traversal of pixels at higher resolution. Besides, by assuming faces detected are of similar size, we can limit the searching depth to finite number of levels after skin color is first detected and hence increase the efficiency. The result of discrete wavelet transform is shown in figure 2. Mathematical details of wavelet theory can be found in [11].



Fig. 2. Image pyramid formed from discrete wavelet transform. At each transform step, 4 components are extracted. The component at the top left corner of the result image is $W_\psi^V(x, y)$, that on the top right corner is $W_\psi^D(x, y)$, that on the bottom right corner is $W_\psi^H(x, y)$ and that on the bottom left corner is $W_\phi(x, y)$. Transformation is done recursively on $W_\phi(x, y)$ according to the wavelet theory.

According to the wavelet theory, the image signal can be broken down into wavelets:

$$f(x, y) = \frac{1}{\sqrt{MN}} \sum_m \sum_n W_\phi(j_0, m, n) \phi_{j_0, m, n}(x, y) + \frac{1}{\sqrt{MN}} \sum_{i=H, V, D} \sum_{j=j_0}^{\infty} \sum_m \sum_n W_\psi^i(j, m, n) \psi_{j, m, n}^i(x, y) \quad (1)$$

$$\begin{aligned} \phi(x, y) &= \phi(x)\phi(y), \quad \psi^H(x, y) = \psi(x)\phi(y), \\ \psi^V(x, y) &= \phi(x)\psi(y), \quad \psi^D(x, y) = \psi(x)\psi(y) \end{aligned} \quad (2)$$

where ϕ is the scaling function and ψ is the wavelet function. The corresponding scaling and wavelet coefficient are W_ϕ and W_ψ respectively. The scaling coefficients form the images at different resolution while the wavelet coefficients form the feature vector in face verification step.

The discrete wavelet transform can be done by:

$$W_\phi(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \phi_{j_0, m, n}(x, y) \quad (3)$$

$$W_\psi^i(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \psi_{j, m, n}^i(x, y) \quad (4)$$

In the system, Daubechies wavelet is used because it has associated speed up algorithms in wavelet decomposition. Efficiency in wavelet decomposition is thus increased.

5 Verification by Support Vector Machine

Support vector machine have been widely used in face detection and recognition recently due to its non-linear classification power [12], [13]. Given data set: $(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l) \in X \times \{+1, -1\}$, support vector machine can learn to find out the association between x_i and y_i . In the proposed system, the x_i will be the normalized wavelet coefficient set $\{W_\psi^D, W_\psi^H, W_\psi^V\}$ after discrete wavelet transform of the image in the Quadtree at the level where skin color is detected and $\{+1, -1\}$ refers to face and non-face classes. Wavelet coefficients are used because it is illumination insensitive and thus much robust in detection than scaling coefficients. During pre-learning phase, discrete wavelet transform will be performed on the face images (inside the face database) and corresponding wavelet coefficients of a certain face image will be extracted as a feature vector. The ORL Database of Faces (<http://www.uk.research.att.com/facedatabase.html>) was used to train the support vector machine. The feature extraction result is shown in figure 3. During learning phase, the support vector machine will be trained to learn the face pattern. During testing phase, the wavelet coefficients correspond to skin region reported by skin color module will be converted to a feature vector and is classified by the support vector machine. Face pattern can then be verified. Details of support vector machine can be found in [14].

In order to use support vector machine, kernel function should be defined. In the proposed system, gaussian RBF kernel, $k(x, x') = e^{\left(\frac{-\|x-x'\|^2}{2\sigma^2}\right)}$, is used.

The determinant function can be written as:

$$f(x) = \text{sgn}\left(\sum_{i=1}^l \alpha_i y_i k(x, x_i) + b\right) \quad (5)$$

During learning phase, α_i are learnt from data set $\{x_i, y_i\}$ under the following criteria function:

$$\max_{\alpha} \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j k(x_i, x_j) \quad (6)$$

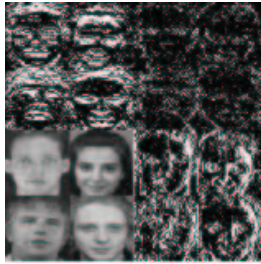


Fig. 3. The wavelet coefficients are extracted from the face images (extracted from ORL Database of Faces). These coefficients will be used to train the support vector machine.

In the system, $\{\alpha_i\}$ are learnt through gradient ascent:

$$\alpha_i^{t+1} = \alpha_i^t + \eta \left(1 - y_i \sum_{j=1}^l \alpha_j y_j k(x_i, x_j) \right) \quad (7)$$

where η is the learning rate.

During face verification or testing phase, equation (5) can be used to determine whether the input is a face pattern.

6 Experiments and Results

The proposed face detection algorithm was implemented using Visual C++ under Microsoft Windows. The experiments were done on a P4 2.26 G Hz computer with 512M ram running Microsoft Windows.

The system was tested by detecting faces in image. The qualitative results are shown in figure 4 and 5 which show that faces can be detected even under illumination variation and existence of distractor (skin color of limbs). The quantitative results are shown in Table 1.

Comparison with face detection algorithm using color analysis alone was made. The qualitative results of using color-based algorithm are shown in figure 6. The figure shows that the algorithm does not work reliably and easily affected by distractors. In addition, the run time is around 1.5 seconds which is not faster than the proposed algorithm significantly. If face verification module is added without using multi-scale analysis. The accuracy does improves, but the run time raises to over 60s on average. This is mainly due to the time consuming morphological operations and brute force verification within skin color region.

7 Conclusion

Face detection is useful in various industrial applications like human-computer interface. However, commonly used face detection algorithms are time consuming



Fig. 4. Each column in this figure shows the face detection result of the input image at first row. The input images show variation in illumination. Second row shows the resultant binary image of the skin detection. Third row shows skin regions detected (in white color) and the face detected (in green box).



Fig. 5. This figure shows the face detection result on the image with multiple faces. The left image is the input image. The middle one shows the resultant binary image of the skin detection. The right image shows skin regions detected (in white color) and faces detected (in green box).



Fig. 6. This figure shows the face detection result on images using common color-based detection algorithm. The left column shows the result on single face detection while the second column shows the result on multiple faces detection. The input image is the same as those used in previous experiments. The first row shows skin regions detected (in white color) and the second row shows the resultant binary image of skin detection.

sample size	image size	hit %	false positive%	run time (s)
100	512x512	80%	0%	3

Table 1. *The table shows the quantitative result of the experiments.*

even if visual cues are used. This paper proposed a multi-scale analysis scheme using Quadtree that start searching for visual cue (skin color) from coarse to fine scale. Searching time is reduced because possible regions will be explored at a lower resolution and the searching is limited to appropriate range of resolution. In addition, face verification ensure high accuracy of detection. Experimental results show that the proposed algorithm can detect faces efficiently and reliably. Note that, the proposed algorithm assumes faces in the image are of similar size and are of frontal view. Detection of face from different views, depth and size will be investigated in future.

References

1. Hjelmas, E., Low, B.K.: Face detection: A survey. *CVIU* **83** (2001) 236–274
2. Yang, M., Kriegman, D., Ahuja, N.: Detecting faces in images: A survey. *PAMI* **24** (2002) 34–58
3. Rowley, H.A., Baluja, S., Kanade, T.: Neural network-based face detection. *PAMI* **20** (1998) 23–38
4. Wu, H., Yoshida, Y., Shioyama, T.: Optimal gabor filters for high speed face identification. In: *ICPR02. (2002) I*: 107–110
5. Dass, S., Jain, A., Lu, X.: Face detection and synthesis using markov random field models. In: *ICPR02. (2002) IV*: 201–204
6. Wang, H., Chang, S.F.: A highly efficient system for automatic face region detection in mpeg video. *IEEE Transactions on circuits and system for video technology* **7** (1997) 615–628
7. Hsu, R.L., Abdel-Mottaleb, M., Jain, A.K.: Face detection in color images. In: *ICIP01. (2001) I*: 1046–1049
8. Phung, S.L., Bouzerdoun, A., Chai, D.: A novel skin color model in ycbcr color space and its application to human face detection. In: *ICIP02. (2002) I*: 289–292
9. Sahbi, H., Boujemaa, N.: Coarse-to-fine face detection based on skin color adaption. In: *ECCV's 2002 Workshop on Biometric Authentication. (2002)* 112–120
10. Swain, M., Ballard, D.: Color indexing. *IJCV* **7** (1991) 11–32
11. Daubechies, I.: *Ten Lectures on Wavelets*. SIAM, Philadelphia (1992)
12. Ai, H., Liang, L., Xu, G.: Face detection based on template matching and support vector machines. In: *ICIP01. (2001) I*: 1006–1009
13. Fransens, R., DePrins, J., Gool, L.J.V.: Svm-based nonparametric discriminant analysis, an application to face detection. In: *ICCV03. (2003)* 1289–1296
14. Scholkopf, B., Burges, C.J.C., Smola, A.J.: *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge, MA (1999)