Automatic face region detection in MPEG video sequences

Hualu Wang and Shih-Fu Chang

Department of Electrical Engineering & Center for Image Technology for New Media
Columbia University, New York, NY 10027, USA

ABSTRACT

In this paper, we present a fast algorithm that automatically detects human face regions in MPEG video sequences. The algorithm takes the DCT coefficients of MPEG video frames as the input, and output the positions of the bounding rectangles of the detected face regions. The algorithm is divided into three stages, where chrominance, shape, and DCT frequency information are used respectively. By detecting faces directly in the DCT domain, there is no need to carry out the inverse DCT transform, so that the algorithm runs in real time. In our experiments, the algorithm detected 92% of the faces in one hundred MPEG I-frames with few false alarms. The algorithm can be applied to JPEG still images or motion JPEG videos as well.

Keywords: face detection, MPEG video, DCT coefficients, chrominance

1. INTRODUCTION

With advances in computing and telecommunications technologies, digital images and video are playing key roles in our information era. The huge amount of visual information is handled by image and video databases, which require effective and efficient mechanisms to index and search these imagery data. In recent years, techniques have been proposed allowing users to search images by visual features, such as texture, color, shape, and sketch, besides traditional textual keywords[3, 5].

Human face is an important subject in image and video databases, because face is a unique feature of human beings, and is ubiquitous in photos, news video, documentaries, etc. Faces can be used to index and search the image/video databases, classify video scenes (e.g., anchorperson v.s. news story), and segment human objects from the background. Therefore, research on face detection is critical in image and video database applications.

Although face detection is related with face recognition, the problem we are facing here is a little different from those in traditional face recognition scenarios. Past work on face recognition has been focused on digital images taken in environments with many constraints. For example, there is usually only a single front-view face in the central part of the image; the head is upright; the background is clean; no occlusion of the face exists; no glasses are worn; and so on[13]. The existence and locations of human faces in these images are known *a priori*, so there is little need to detect and locate faces. There are two major classes of techniques in face recognition, namely geometric-feature-based matching, and template matching[1]. The former techniques extract and normalize geometric descriptors of faces and face components for recognition. The latter ones are variations of template matching techniques, including deformable template[17] and eigenfaces[14].

In general image and video databases, however, there is little or no constraint on the number, location, size, and orientation of human faces in the scenes. The backgrounds of these images and video scenes are usually cluttered. Thus, successful face detection becomes important and challenging before the indexing, search, and recognition of the faces could be done.

Recently, work has begun in face detection and location in unconstrained images. One approach involves model-based algorithms that treat human faces as a whole unit consisting of inter-connected arcs that represent chins and hairline[7]. Another approach detects face components such as eyes, noses, nostrils at first, then uses a ranking technique to get possible face locations[2]. Other methods are based on ellipse fitting[4], multiresolution domain knowledge[16], neural networks[11], and zero-crossings of a wavelet transform[15]. Limitations of current techniques are as follows: computation is carried out in uncompressed pixel domain and usually expensive; color information is seldom used; and, there are still strong constraints in some of these algorithms.

In this project, our objective is to detect and locate face regions in MPEG video streams, with little constraints on the content of the video scenes. Although our work is complementary to traditional face recognition research, its focus is somewhat different. For the purpose of indexing and search in video databases, our algorithm has to be automatic and fast, so that it can be applied to real systems with hugh amount of data. Also, the algorithm is performed in the standard MPEG compressed domain, with as little decoding of the video streams as possible. Thus, the data amount for storage and processing can be greatly reduced, and useful visual features can be obtained more easily. These advantages will be demonstrated in the following sections.

The paper is organized as follows. Section 2 describes our automatic algorithm that detects face regions in MPEG video sequences. The algorithm is broken into three stages. Section 3 gives the experimental results of our algorithm, with discussions on its advantages and limitations. Conclusion and future work are presented in Section 4.

2. AUTOMATIC FACE DETECTION IN MPEG VIDEO SEQUENCES

In this project, we present a fast algorithm that automatically detects face regions in MPEG video sequences. Our algorithm takes as input the DCT coefficients of the macroblocks of MPEG frames[10], and generates positions of the bounding rectangles of the detected face regions. By detecting faces using the DCT coefficients, only minimal decoding of the compressed video sequence is necessary. The costly inverse DCT transform is avoided. Thus, the algorithm can achieve very high speed. The DCT coefficients can be obtained easily from I frames of MPEG videos. For B and P frames, transform-domain inverse compensation can be applied to obtain the corresponding DCT coefficients.

Our algorithm consists of three stages, where chrominance, shape, and DCT frequency information are used respectively. MPEG macroblock (16x16 pixels) is our processing unit, so that the bounding rectangles of the detected face regions have a resolution limited by the boundaries of the macroblocks.

In the first stage, average chrominance of each macroblock is used to determine if it is a possible face macroblock, based on our statistics of skin-tone colors in the chrominance plane. In the second stage, we apply shape constraints to the candidate macroblocks in the first stage, to eliminate false alarms of face detection. In the third stage, energy distribution in the DCT frequency domain of the candidate face regions is used as the final verification. We use the fastest stage at the beginning of the algorithm, so that more costly stages will have to process a small subset of data only.

2.1. Macroblock Classification Based on Average Chrominance

Human skin tones form a special category of colors, distinctive from the colors of most other natural objects. Although skin colors differ from person to person, they are distributed over a very small area in the chrominance plane. This has been noticed and used by researchers in consumer electronics to design TV circuits that automatically detect and correct human skin-tone colors[8, 12]. Their objective is to minimize the distortion of skin-tone colors, which are obvious to human eyes. I and Q chrominance components of the NTSC TV signal are used to estimate the hue and saturation of a color. Colors with hue and saturation in certain ranges are classified to skin-tone colors[12], based on extensive observation. By removing the luminance component of colors, the differences among skin colors of different races and the effect of lighting conditions are greatly reduced.

In our project, we make use of the skin-tone characteristics as well, but take a different approach. First, we take Cb, Cr as the chrominance components, instead of I and Q. Second, we classify the colors based on statistical decision rules.

We use Cb and Cr, because they are usually the chrominance components used in MPEG video sequences. The linear conversion of colors from the RGB space to the YCbCr space is avoided. Figure 1(a) shows that all colors in the RGB color cube are mapped to a hexagon in the Cr-Cb chrominance plane. The letters B, M, R, Y, G, C, W, correspond to the hues of blue, magenta, red, yellow, green, cyan, and white, respectively. Note that W is the origin of the Cr-Cb plane, and it actually corresponds to all gray-level colors from black to white. (The intensity variation in Figure 1(a) is because we printed the original color image on a black-and-white printer.) Figure 1(b) shows the distribution of skin-tone colors in Cr-Cb plan, based on over forty sample face patches of different races that we cut from a news video sequence. The white area corresponds to skin-tone colors. As we mentioned, skin-tone colors are distributed over a very small area in the chrominance plane.

We use *Bayes decision rule for minimum cost* to classify a color into skin-tone class or non-skin-tone class. We believe that for the purpose of face detection in digital video, statistics of the skin-tone colors is needed. Also, because we carry out this classification in the first stage of our algorithm, we allow some false alarms, which can be eliminated in subsequent stages. On the other hand, we should avoid false dismissals, because they can not be recovered in the following steps of the algorithm. Hence, we should put different costs, i.e., penalties, on these false classifications. The Bayes decision rule for minimum cost[6] is as follows (for two classes):

$$R(X_0) = C_{00} \cdot p(X|\omega_0) \cdot p(\omega_0) + C_{10} \cdot p(X|\omega_1) \cdot p(\omega_1) \quad (1)$$

$$R(X_1) = C_{01} \cdot p(X|\omega_0) \cdot p(\omega_0) + C_{11} \cdot p(X|\omega_1) \cdot p(\omega_1) \quad (2)$$

$$R(X_0) < R(X_1) \Rightarrow X \in \omega_0 \quad (3)$$

$$R(X_0) > R(X_1) \Rightarrow X \in \omega_1 \quad (4)$$

where $p(\omega_0)$ and $p(\omega_1)$ are the *a priori* probabilities of non-skin-tone color class (ω_0) and skin-tone color class (ω_1), respectively; $p(X|\omega_0)$ and $p(X|\omega_1)$ are the conditional probability density functions (likelihoods) of these two classes of colors in the chrominance plane. In our experiments, all these parameters are estimated using sampled frames, face image patches, and non-face image patches from MPEG-compressed news video sequences. Cost parameters, i.e., C_{00}, C_{01}, C_{10}, and C_{11}, are set to 0.0, 0.1, 0.8, and 0.0, respectively. Namely, there is no cost for correct detections or correct rejections, a small cost for false alarms, and a big one for false dismissals.

Experiments show that our method based on the statistical decision rule gives more accurate result, compared with the method suggested in [12]. An example of comparison is given in Figure 2. Figure 2(a) is one frame in a complex scene from a news video sequence (the original is in color). Each pixel in this frame is classified as skin-tone colors or non-skin-tone colors. Figures 2(b) and 2(c) are the classification results using the method in [12] and our method, respectively. Gray pixels correspond to skin-tone colors; black pixels the opposite.

We apply the above minimum cost decision rule to MPEG video streams, and classify each MPEG macroblock as a candidate face macroblock or a non-face one. We use only the DCT DC coefficients of the corresponding Cr and Cb blocks, which are equivalent to the average values (up to a scale) of the chrominance blocks in the pixel domain. This means that we are working on a lower-resolution version of the video frames, so that the inverse DCT transform is avoided, and the following stages of the algorithm can be sped up. Higher resolution in skin-tone detection can be achieved by taking more DCT coefficients (e.g., a few low-order coefficients) as input in the above classification process.

After the classification, we get a binary mask image for each video frame. Each value in the mask indicates the classification results of the corresponding macroblock. Then, a 3x3 (macroblocks) cross median filter is applied to the binary mask image, to remove noise and smooth the image. The filtering is helpful, because faces are connected regions, and are homogenous in chrominance. Figure 3(a) is one frame from a news video sequence. Figure 3(b) shows the binary mask image after the classification by average chrominance. Figure 3(c) is the result after applying the median filter to Figure 3(b). It is clear that the result in Figure 3(c) is more desirable.

2.2. Shape Constraints on Candidate Face Macroblocks

Clearly, chrominance alone is not enough to detect face regions. In a video sequence with complex scenes, besides human faces, there are other exposed parts of the body with skin tones, and there are natural scenes with colors quite similar to skin tones (e.g., a desert scene). Therefore, in the second stage of our algorithm, we apply shape constraints on the binary mask images generated by the first stage.

The shape of human faces can be approximated by connected arcs[7], or more simply, by an ellipse[4]. Typical face outlines have been found to have aspect ratios in a narrow range consistently[7]. Since our face detection algorithm works on a macroblock basis, it is difficult to use ellipse or arcs to describe face outlines without the inverse DCT transform. Therefore, we use rectangles to approximate face regions, and use locations of rectangles as the boundaries of faces.

These rectangles, however, are not arbitrary. They have certain aspect ratios, and are bounded in size. In our algorithm, we set the range of aspect ratios for these rectangles to [0.9, 1.7], based on experiments. The size of the faces are obviously

upper bounded by the size of the video frames. It is lower bounded as well, because it is generally believed in face recognition field that 32x32 pixels is the lower limit for face detection[13]. Since we are working in the compressed domain, we set the lower limit of our face detection algorithm to 48x48 pixels, or, 3x3 macroblocks.

In summary, the shape constraints we put on binary mask images are: (1) faces are contiguous regions that fit well in their bounding rectangles, whether the face is front view or side view, or whether the head is upright or a little tilted; (2) the size of the bounding rectangles is bounded by the lower limit of face detection and the size of the video frames; (3) the aspect ratios of the bounding rectangles should be in a certain range. Thus, at this stage, face detection becomes the task to search for face-bounding rectangles that satisfy the above constraints.

To accomplish this task, we use an intuitive method of low complexity. The idea is to use rectangles of possible sizes and aspect ratios as face templates to match against the binary mask images. A face template is shown in Figure 4(a), whose size is MxN macroblocks (the shaded rectangle). The macroblocks adjacent to the sides and top of the rectangle are considered as the background of face regions. The macroblocks adjacent to the bottom of the rectangle are not considered as a part of the background, because they can be either the exposed neck or clothes and have no definitive color characteristics. Because the background is usually distinctive from faces, there should be few face-macroblocks in the background region. Our matching criterion is as follows. Count the number of face-macroblocks inside the face (shaded) rectangle, as well as the numbers of face-macroblocks in the top, left, and right parts of the background region. Denote the numbers as N_0, N_1, N_2, and N_3, respectively. Only if N_0 is above a threshold, and N_1, N_2, N_3 are below certain thresholds, we declare a match. This is illustrated in Figure 4. Figure 4(b) is a match, because the face region is almost covered by face-macroblocks, and there are few face-macroblocks in the background region. Figure 4(c) is not a match, because the face rectangle is not covered enough by face-macroblocks. Figure 4(d) is not a match either, because there are too many face-macroblocks in the background region.

To limit the search area for matching, we detect non-overlapping rectangular regions that cover contiguous face macroblocks. As an example, Figure 5(a) is the binary mask image corresponding to a video frame with two faces in it. We first project the binary mask image onto the x and y axes. Based on the zero-runs and non-zero-runs of the projections, we segment the binary mask image into rectangular regions that contain either contiguous face-macroblocks, or only non-face macroblocks (Figure 5(b)). Those regions having only non-face macroblocks or too few face-macroblocks will be discarded, and not considered for matching. Thus the areas to search for face regions will be greatly reduced, as shown in Figure 5(c).

Then, in each of these non-overlapping regions, we apply the face template matching mentioned above. Because the size of the face regions is unknown, we start from the largest possible rectangle for each region, then gradually reduce the template size. Therefore, all sizes of face regions can be detected. Finally, overlapping face rectangles are resolved. If only a small area is overlapped, we keep both regions as valid face regions. If one of the region is small and the overlapping area is large compared with its size, we discard the small rectangle.

An example is shown in Figure 6. The original video frame is in Figure 6(a). The binary mask image is in Figure 6(b), along with search regions (bounded by white rectangular frames). Figure 6(c) shows the detected face regions before stage three. The final result after stage three is overlaid on Figure 6(a). In some cases, there might be non-face regions that have similar colors to skin tones, and have a roughly rectangular shape. This will cause false alarms, as seen from the rectangle in the lower-left corner of Figure 6(c). This problem can be solved in stage three of our algorithm (see Figure 6(a)), as will be described in the following subsection.

2.3. Verification Based on Energy Distribution of Luminance DCT Coefficients

In the final stage of our algorithm, we verify the rectangular face regions detected in the second stage, by testing the energy distribution of the luminance (Y-component in MPEG sequences) DCT coefficients. Since the existence of eyes, nose base, and lips in the face region causes luminance variation in the vertical direction, we expect some energy in the luminance DCT coefficients corresponding to high frequencies in the vertical direction. Usually, the intensity variation in the horizontal direction is less significant than that of the vertical one.

We follow the DCT coefficient grouping scheme proposed in [9]. In this scheme, the DCT coefficients in a 8x8 transform block are partitioned into groups corresponding to the directional features (horizontal, vertical, and diagonal edges) of the block in the spatial domain, along with the DC coefficient. Here we compute the percentage of energy in the luminance DCT

DC coefficient, and denote it as E_{DC}. We also compute the percentage of energy in those luminance DCT coefficients corresponding to high frequencies in the vertical direction, and denote it as E_V. These two parameters are tested against thresholds T_{DC} and T_V, respectively. If E_{DC} is larger than T_{DC}, then the region is too smooth to be a face. If E_V is smaller than T_V, the region is unlikely to be a face, either. This helps to remove some false alarms from stage two. For example, the false detection in Figure 6(c) is removed after the verification, and does not appear in the final detection result (Figure 6(a)).

3. EXPERIMENTAL RESULTS AND DISCUSSIONS

We have tested our algorithm on 100 I-frames from an MPEG-compressed CNN news video. The I-frames include anchorperson scenes, news stories, interviews, and commercials. The size of each frame is 352x240 pixels. The 4:2:0 macroblock format is used, which means that chrominance signals are subsampled by two, both horizontally and vertically. There are totally 91 faces in these frames, including frontal views, semi-frontal views, side views, and tilted faces. These faces are usually contained in complex scenes. Some of the frames have multiple faces. Our algorithm detected 84 of the faces (92%), including faces of different sizes, frontal and side-view faces, etc. Detected face regions are marked by white rectangular frames overlaid on the original video frames. Frames without faces but with exposed arms and hands can be successfully rejected. There were 8 false alarms in our experiment.

The run time of our algorithm ranges from 1 to 14 milliseconds per frame on a SGI ONYX workstation, depending on the complexity of the scenes in the video frames. So this algorithm can be performed in real time.

The algorithm is restricted in several aspects. It can only be applied to color images and videos. False dismissals can not be totally avoided, especially in very cluttered scenes with small faces. There are still false alarms even after we apply the shape and energy constraints. Nevertheless, our compressed-domain approach is very efficient and can be applied to large video databases for indexing and recognition. It helps very much in focusing our attention to only a small portion of the entire video sequences and frames, so that the amount of decoding can be greatly reduced.

The algorithm does not give the exact outlines of the faces detected, because we avoid inverse DCT transform and work on the macroblock basis. The positions of the faces detected are sometimes not perfectly aligned, because the face rectangles we detect lie on the borders of 16x16 macroblocks. This can be improved if the compressed video sequence has a format with more chrominance information, e.g., the 4:2:2 format, thus we can work on 8x8 blocks so resolution of the result can be much improved. Also, we can use more DCT coefficients rather than just the DC coefficient of the Cb and Cr blocks, to get more accurate results.

4. CONCLUSION AND FUTURE WORK

We propose and demonstrate a fast algorithm to detect rectangular face regions directly in MPEG video streams. The run time of the algorithm ranges from 1 to 14 milliseconds per frame (352x240 pixels) on a SGI ONYX workstation. Experiments showed that it is effective for general video streams with little constraints on the number, size, and orientation of faces, such as a news video sequence.

In future work we will consider incorporating motion information of MPEG video with the current algorithm, as well as domain knowledge. Some decoding will be performed for more accurate face locations, or for the purpose of face recognition.

5. REFERENCES

[1] Brunelli, R. and Poggio, T., “Face Recognition: Features versus Templates,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 10, October 1993.

[2] Burl, M.C., Leung, T.K., and Perona, P., “Face Localization via Shape Statistics,” *International Workshop on Automatic Face and Gesture Recognition*, June 1995.

[3] Chang, S.-F. and Smith, J.R., “Extracting Multi-Dimensional Signal Features for Content-Based Visual Query,” *SPIE Symposium on Visual Communications and Signal Processing*, May 1995.

[4] Eleftheriadis, A. and Jacquin, A., "Model-Assisted Coding of Video Teleconferencing Sequences at Low Bit Rates," *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 3.177-3.180, May-June 1994.

[5] Niblack, W., Barber, R., Equitz, W., Flickner, M., Glasman, E., Petkovic, D., Yanker, P., Faloutsos, C., and Taubin, G., "The QBIC Project: Querying Images by Content Using Color, Texture and Shape", *SPIE 1993 International Symposium on Electronic Imaging: Science and Technology, Conference 1908, Storage and Retrieval for Image and Video databases*, February 1993.

[6] Fukunaga, K., *Introduction to Statistical Pattern Recognition*, 2nd Ed., Academic Press, Inc., 1990.

[7] Govindaraju, V., Srihari, R.K., and Sher, D.B., "A Computational Model for Face Location," *Proceedings of the Third International Conference on Computer Vision*, pp. 718-721, 1990.

[8] Harwood, L.A., "A Chrominance Demodulator IC with Dynamic Flesh Correction," *IEEE Transactions on Consumer Electronics*, Vol. CE-22, pp. 111-117, February 1976.

[9] Ho, Y.S. and Gersho, A., "Classified Transform Coding of Images using Vector Quantization," *IEEE International Conference on ASSP*, pp. 1890-1893, May 1989.

[10] ISO/IEC 13818 - 2 Committee Draft (MPEG-2).

[11] Rowley, H., Baluja, S., and Kanade, T., "Human Face Detection in Visual Scenes," *Carnegie Mellon University Computer Science Technical Report*, CMU-CS-95-158R.

[12] Rzeszewski, T. "A Novel Automatic Hue Control System," *IEEE Transactions on Consumer Electronics*, Vol. CE-21, pp. 155-162, May 1975.

[13] Samal, A. and Iyengar, P.A., "Automatic Recognition and Analysis of Human Faces and Facial Expressions: A Survey," *Pattern Recognition*, Vol. 25, No. 1, pp. 65-77, 1992.

[14] Turk, M. and Pentland, A., "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, pp.71-86, 1991.

[15] Venkatraman, M. and Govindaraju, V., "Zero-Crossings of a Nonorthogonal Wavelet Transform for Complex Object Location," *Proceedings of the IEEE International Conference on Image Processing*, October 1995.

[16] Yang, G. and Huang, T.S., "Human Face Detection in a Complex Background," *Pattern Recognition*, Vol. 27, Mo. 1, pp. 53-63, 1994.

[17] Yuille, A.L., "Deformable Templates for Face Recognition," *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, pp. 59-70, 1991.

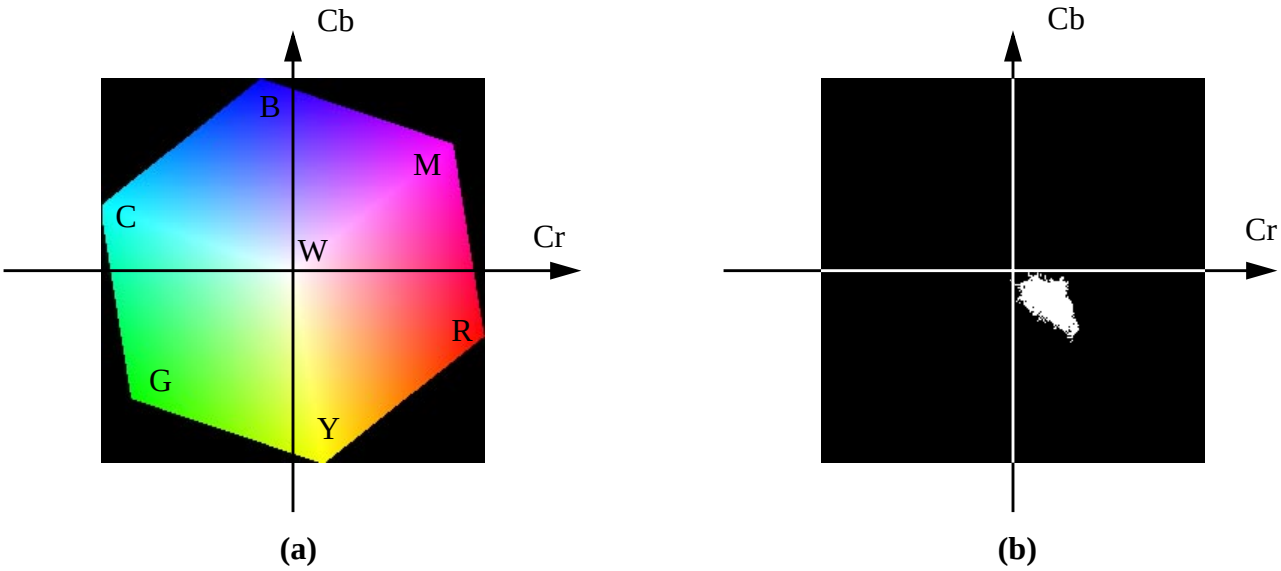
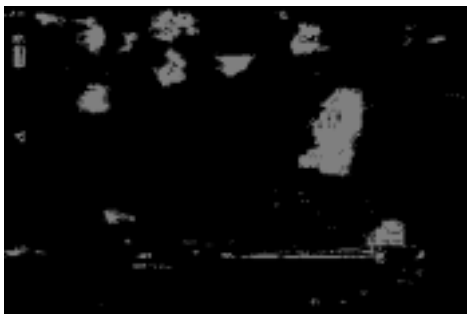


FIGURE 1. Color distributions on Cr-Cb chrominance plan: (a) all RGB colors; (b) skin-tone colors.

(a)

(b)



(c)

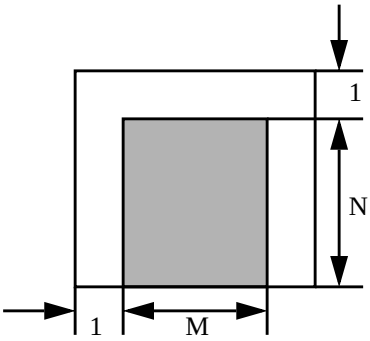
FIGURE 2. Comparison of skin-tone classification methods: (a) the original video frame; (b) the result using the method suggested in [12]; (c) the result using our method.

(a)

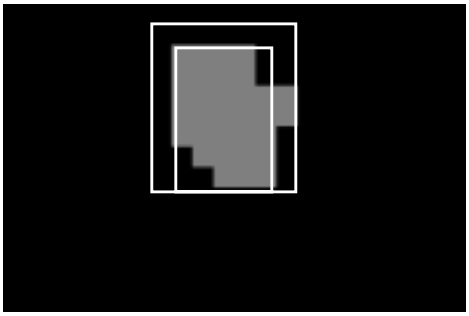
(b)

(c)

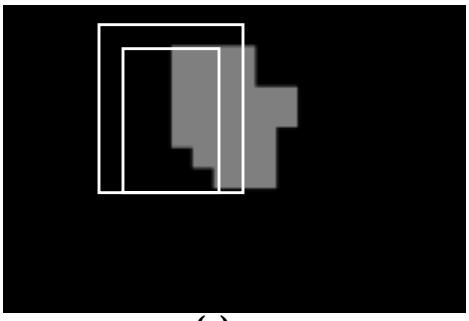
FIGURE 3. Macroblock classification by average chrominance: (a) the original video frame; (b) the initial classification result; (c) the result after median filtering.



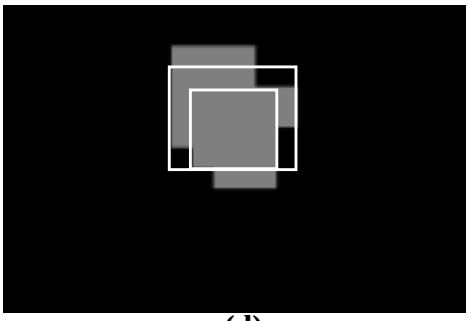
(a)



(b)



(c)



(d)

FIGURE 4. Face region template matching: (a) a rectangular face template with a background region; (b) a match; (c) and (d) non-matches.

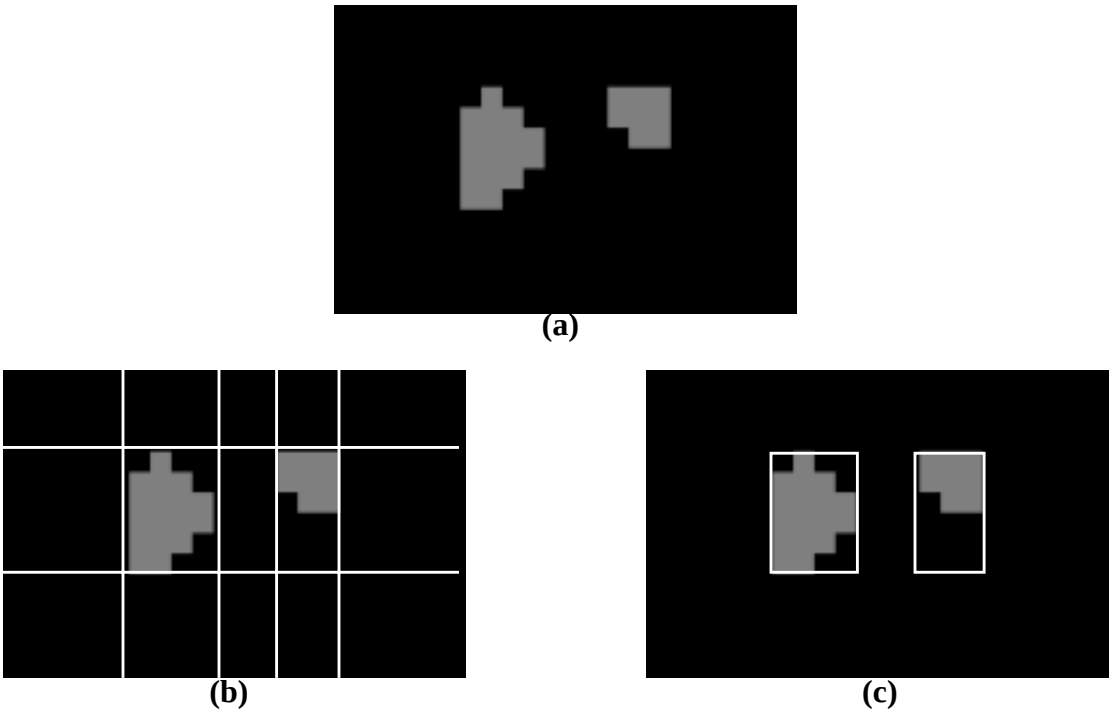


FIGURE 5. Reduction of search area: (a) the binary mask image corresponding to a video frame with two faces; (b) segmentation based on the projections to x and y axes; (c) the final search area for matching.

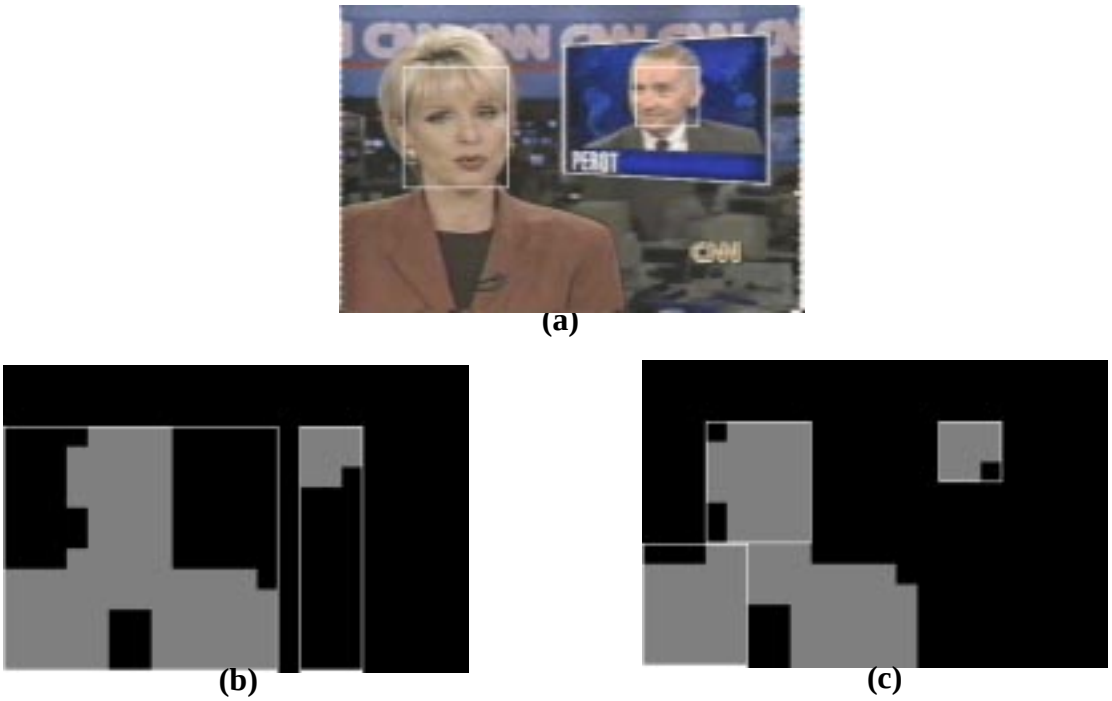


FIGURE 6. Face region detection: (a) the original video frame with final detection result overlaid; (b) the binary mask image, with reduced search region; (c) detected face regions (with a false alarm).