Saliency Model based Face Segmentation and Tracking in Head-and-Shoulder Video Sequences

Hongliang Li*, King N. Ngan*

*Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

In this paper, a novel face segmentation algorithm is proposed based on facial saliency map (FSM) for head-and-shoulder type video application. This method consists of three stages. The first stage is to generate the saliency map of input video image by our proposed facial attention model. In the second stage, a geometric model and an eye-map built from chrominance components are employed to localize the face region according to the saliency map. The third stage involves the adaptive boundary correction and the final face contour extraction. Based on the segmented result, an effective boundary saliency map (BSM) is then constructed, and applied for the tracking based segmentation of the successive frames. Experimental evaluation on test sequences shows that the proposed method is capable of segmenting the face area quite effectively.

1. Introduction

Object segmentation plays an important role in content-based multimedia applications. From the content-related image and video segmentation, a higher semantic object can be detected and exploited to provide the user with flexibility in content-based access and manipulation[1][2][4]. As an important key to the future advances in human-to-machine communication, the segmentation of a facial region also can be applied in many fields, such as encoding, indexing, and pattern-recognition purposes [3]. In the literature, large number of face segmentation algorithms based on different assumptions and applications have been reported. According to the primary criterion for segmentation, two categories can be classified: color-based methods [3]-[22] and facial features-based methods [23]-[29].

The color-based segmentation methods aim to exploit skin color information to locate and extract the face region. A universal skin-color map is introduced and used on the chrominance component to detect pixels with skin-color appearance in [3]. In order to overcome the limitations of color segmentation, five operating stages are employed to refine the output result, such as density and luminance regularization, geometric correction. Based on color clustering and filtering using approximations of the YCbCr and HSV skin color subspaces, a scheme for human faces detection by performing a wavelet packet decomposition was proposed in [4]. The wavelet coefficients of the band filtered images are used to characterize the face texture and form compact and meaningful feature vectors.

*The authors are with the Department of Electronic Engineering, The Chinese University of Hong Kong (e-mail: {hlli, kmngan}@ee.cuhk.edu.hk).
It is known that both of the previous two methods are based on the linear classifier for skin color pixels. Using a lighting compensation technique and a nonlinear color transformation, a face detection algorithm [5] for color images was presented to detect skin regions over the entire image. This algorithm extracts facial features by constructing feature maps for the eyes, mouth, and face boundary. Based on the joint processing of color and motion, a nonlinear color transform relevant for hue segmentation is derived from a logarithmic model [6]. Markov random field (MRF) model that combines hue and motion detection within a spatiotemporal neighborhood is used to realize the hierarchical segmentation. In addition, a hand and face segmentation method using color and motion cues for the content-based representation of sign language videos was also proposed in [7]. This method consists of three stages: skin-color segmentation, change detection and face and hand segmentation mask generation.

The facial features-based method, on the other hand, utilizes the facial statistical or other structural features rather than skin-color information to achieve face detection/segmentation [23]-[30]. In [23], a probabilistic method for detecting and tracking multiple faces in a video sequence was presented. The proposed method integrates the information of face probabilities provided by the detector and the temporal information provided by the tracker to produce the available detection and tracking methods. In [30], a statistical model-based video segmentation for head-and-shoulder type video was addressed. The head is modeled with a "blob", which is segmented based on the assumption that a background scene contains no foreground in order to satisfy the creation of a background model. Recently, many segmentation works have been developed to extract the object in the image/video [31]-[34]. A bilayer video segmentation method was proposed based on the tree classifiers [31], [35]. In that work, visual cues such as motion, motion context, colour, contrast and spatial priors are fused together by means of a Conditional Random Field model, and then segmented by binary min-cut [36]. In addition, In [34], a random walker segmentation method is proposed for performing multilabel, interactive image segmentation.

In this paper, we will concentrate on the specific application domain, namely head-and-shoulder type videos existing in many multimedia service such as videophone, videoconferencing, and web chatting, etc. An effective face segmentation algorithm is presented based on skin color, face position, and eye-map information. Our segmentation method consists of three stages. The first stage is to generate the saliency map of input video image by our new facial attention model. In the second stage, a geometric model and an eye-map built from chrominance components are employed to localize face regions. The third stage involves the boundary correction and the final face contour extraction. Finally, we employ the proposed BSM to segment the facial region in the successive frames.

The main contributions and advantages of this work are the novel face segmentation algorithm developed for head-and-shoulder type video application based on facial saliency map. In this work, the facial and boundary saliency maps are constructed successfully based on the attention model to achieve the face segmentation and tracking, which have been evaluated by a large number of images/videos with good performance. In addition, this work provides a general method to build the saliency model by combining different cues, such as edge, color, and shape. This work can be easily extended to other objects by appropriate design of the object saliency model.
This paper is organized as follows. The face segmentation algorithm will be presented in Section 2. Section 3 presents the tracking based segmentation. Experimental results are provided in Section 4 to support the efficiency of our face segmentation algorithm. Finally, in Section 5, conclusions are drawn and further research proposed.

2. Face Segmentation Algorithm

Generally, video segmentation can be decomposed into two sub-problems, namely video object segmentation and video object tracking [1][2]. The first stage is used to detect the object and extract it from the input frame. Then the object in the successive frames will be segmented in the following tracking procedure. In this section, we will present our method to locate candidate face areas in the color image. It is known that there are many color space models relevant to different applications, such as RGB for display purpose, HSV for computer graphics and image analysis, and YCbCr. Since the YCbCr color space is usually employed for the video storage and coding, and can provide effective analysis for human skin color [3][7], we will use this color space for our input video sources. Namely, the format the video image is the YCbCr color space with the spatial sampling ratio of 4:2:0 in our work. In addition, we assume that the person in a head-and-shoulder pattern appears in the typical video sequences with front or near front views.

2.1. Facial Saliency Map

Motivated by the visual attention model [37], which has been successfully extended to some applications [38]-[40], we will employ the similar idea to construct a facial saliency map to indicate the face location in the typical head-and-shoulder video sequences. Assume $(x, y)$ represent the spatial position of a pixel in the current image. The corresponding luminance and chrominance components of the pixel are denoted by $Y(x, y)$, $Cb(x, y)$, and $Cr(x, y)$, respectively. The FSM in our work can be defined as:

$$ S(x, y) = P_1(x, y) \cdot P_2(x, y) \cdot P_3(x, y), $$

where $P_1$, $P_2$, and $P_3$ denote the "conspicuity maps" corresponding to the chrominance, position, and luminance components, respectively. The detailed discussion of the maps can be found in the following.

2.1.1. Chrominance Conspicuity Map (CCM) $P_1$

Skin-color can be detected by the presence of a certain range of chrominance values with narrow and consistent distribution in the YCbCr color space. The empirical ranges for the chrominance values employed are typically $Cr_{skin} = [133, 173]$ and $Cb_{skin} = [77, 127]$ [3]. It is known that face region usually exhibits the similar skin-color feature regardless of different skin types. Therefore, using the skin-color information, we can easily construct the facial saliency map to locate the potential face areas.

To investigate the skin-color distribution, we manually segmented the training images into face patches. The data were taken from the California Institute of Technology face databases and CVL Face Database that are provided by the Computer Vision Laboratory [41], which contains 1248 color human faces. Each person has different poses and emotions. Different lighting conditions and face types can be found for these image sources. It should be noticed that there are no test images in the experiments included in the training data.
The histogram results for Cb and Cr components are presented in Fig. 1(a) and (b). We can see that the values of chrominance for different facial skin colors are indeed narrowly distributed, which is identical with that statistical results in [3][4][7]. The corresponding histograms exhibit distinctly the Gaussian-similar distribution rather than the uniform distribution. The larger the offset of a pixel away from the mean value, the smaller the probability of the pixel belongs to the face area. In addition, from the distribution of facial pixels in the CbCr plane shown in Fig. 1(c), we find that an obviously declining angle can be observed between two chrominance components. Let $\mu$ and $\Delta$ denote the mean and variance, and $\theta$ be the angle. Based on the above analysis, we can define the CCM $P_1$ as

$$P_1(x, y) = \exp\{-\omega_{cr}(x, y) \cdot \frac{Cr(x, y)^2}{2\Delta_{cr}^2} + \omega_{cb}(x, y) \cdot \frac{Cb(x, y)^2}{2\Delta_{cb}^2}\},$$

(2)

where $\mu_{cr} = 153$, $\mu_{cb} = 102$, $\Delta_{cr} = 20$, $\Delta_{cb} = 25$, $\theta = \frac{\pi}{4}$, which are determined from the training data and experimental test. We believe that more accurate parameters estimation and update method can be employed to improve the parameters estimation. $Cr'$ and $Cb'$ are computed from the rotation of coordinate, i.e., $Cr'(x, y) = (Cr(x, y) - \mu_{cr}) \cdot \cos(\theta) + (Cb(x, y) - \mu_{cb}) \cdot \sin(\theta)$, and $Cb'(x, y) = -(Cr(x, y) - \mu_{cr}) \cdot \sin(\theta) + (Cb(x, y) - \mu_{cb}) \cdot \cos(\theta)$. Variable $\omega_v(x, y)$ ($v = cb$ or $cr$) is a weight coefficient, which is employed to adjust the decreasing level of the chrominance distribution curve. Namely, if the chrominance values of a pixel exceed the facial distribution region, its corresponding CCM obtained by the weight computation will tend to have smaller probability to be classified as facial pixel.

It is given by

$$\omega_v(x, y) = a^{\lambda_v(x, y)},$$

(3)

$$\lambda_v(x, y) = \begin{cases} \text{sym}\{\mu_{cr} + \Delta_{cr} - Cr(x, y)\}(Cr(x, y) - \mu_{cr} + \Delta_{cr})\}, & \text{if} \ v = cr \\ \text{sym}\{\mu_{cb} + \Delta_{cb} - Cb(x, y)\}(Cb(x, y) - \mu_{cb} + \Delta_{cb})\}, & \text{if} \ v = cb \end{cases}$$

(4)

Here, $a$ is a constant with the value of 2 is employed in our work, and $\text{sym}\{k\}$ denotes a
sign function:

\[
\text{sym}(k) = \begin{cases} 
1, & \text{if } k \geq 0 \\
0, & \text{otherwise}
\end{cases}
\] (5)

2.1.2. Position Conspicuity Map (PCM) \( P_2 \)

We have found that in a typical head-and-shoulder video sequences, most of the face locations appear at or near the center of the image in order to attract user attention distinctly. Few human faces are captured and shown at the boundary of the image, especially the bottom of the image. Fig. 2 illustrates a statistical result of the face pixel positions (denoted by the white area) from standard video sequences, including *Akiyo*, *Carphone*, *Claire*, and *Salesman* with total 1200 frames. From the obtained result, we can see that the larger the distance between the current and the center positions, the smaller the possibility of face appearance. Hence, it is reasonable to assume that the probability of the facial pixels appearing at the center of the image will be larger than other locations. Let \( H \) and \( W \) denote the height and width of the image, respectively. Based on this characteristic, we define the Position Conspicuity Map \( P_2 \) as

\[
P_2(x, y) = \exp\left\{ -\frac{(x - H/2)^2}{0.8 \cdot (H/2)^2} - \frac{(y - W/2)^2}{2 \cdot (W/3)^2} \right\}.
\] (6)

From (6), we found that the probability of vertical orientation decays faster than that of the horizontal. The rectangle area of \( \frac{H}{2} \times \frac{W}{3} \) at the center holds the larger conspicuity values.

2.1.3. Luminance and Structure Conspicuity Map (LSCM) \( P_3 \)

Although there is no narrow distribution in the face area for the luminance component, different density values can be found distinctly on the interval \((0, 255)\). As shown in Fig. 3, the region of \([128 - 50, 128 + 50]\) tends to contain most of conspicuity values for the facial skin area. The darker the intensity value of a pixel, the less possible it will be a skin-tone color. Similar result can also be found for the very bright pixels. The reason is that a "head-and-shoulder" region, as the foreground, usually exhibits more uneven distribution of brightness, and provides a clearer visual result for user rather than the
Figure 3. The histogram of luminance component Y for the test face data.

background. In addition, it is known that the brightness in the facial area is usually nonuniform, and exhibits larger deviation than chrominance components. Based on the analysis, we next define the LSCM $P_3$ as

$$P_3(x, y) = \left( 1 - \frac{1}{(1 + \sigma(N_{w1}(x, y)))^{1/n_e}} \right) \cdot \exp\left\{ -\frac{(\gamma(x, y) \cdot Y'(x, y) - \mu_L)^2}{2 \cdot \Delta_L^2} \right\}, \tag{7}$$

$$n_e = \max\{\log_2\left( \frac{\max(N_{w1}(x, y))}{\text{mean}(N_{w1}(x, y))} \right), 1\}, \tag{8}$$

where $\mu_L = 128$, $\Delta_L = 50$, $\gamma(x, y)$ denotes the luminance compensation coefficient, which is defined as $\gamma(x, y) = \frac{\mu_L}{\sum_{k=1}^{\infty} \gamma'(x_k, y_k)}$ for $(x_k, y_k) \in N_{w2}(x, y)$ and $0.3|Cb(x_k, y_k) - Cb(x, y)| + 0.7|Cr(x,k, y_k) - Cr(x, y)| < 2$. $N_{w}(x, y)$ represents the $w \times w$ neighborhood of pixel $(x, y)$. $\sigma(N_{w1}(x, y))$, $\max(N_{w1}(x, y))$ and $\text{mean}(N_{w1}(x, y))$ denote the standard deviation, maximal and mean values of the $w1 \times w1$ neighborhood, respectively. Due to the sample format of (4:2:0), we use $Y'(x, y)$ to denote the mean value of four luminance values corresponding to the chrominance location $(x, y)$. Here, the first term in (7) represents the structural coefficient, which is employed to characterize the luminance variation in face region. However, it is observed that a region with larger local smooth areas and many sharp edges is likely to exhibit distinct deviation in brightness. In order to avoid large prediction errors in the presence of edges, we employ a coefficient $n_e$ to approximately estimate the edge character. We can see that only those larger derivation with small edge strength will be taken into account, which means that the influence yielded by the high-frequency contents such as the sharp edges will be reduced significantly by (7).

Additionally, it is widely reported that the appearance of the skin-tone color is characterized by the brightness of color, which is governed by the luminance component of light \[3\][5][7][9][14]. We now introduce a local lighting balancing technique, which has a similar objective as the method in [42], to normalize the color appearance. $\gamma(x, y)$ given in (7) is proposed to reduce the influence of the light variation on the face detection. For the current pixel $(x, y)$, we first consider all the pixels in the window of $w2 \times w2,$
which are centered at \((x, y)\). The corresponding distances of chrominance components between the pixel \((x, y)\) and others are then computed, respectively. Only those pixels with smaller distance values (i.e., \(< 2\)) are selected as the light balancing pixels. From (7), we can see that if the current pixel has the lower density value (i.e., in the darker area), its luminance level would be increased since the balancing coefficient \(\gamma(x, y)\) corresponds to a larger value. It should be noticed that this balancing process is only employed to the pixel with its chrominance values satisfying \(Cb(x, y) \in [\mu_{cb} - \Delta_{cb}, \mu_{cb} + \Delta_{cb}]\) and \(Cr(x, y) \in [\mu_{cr} - \Delta_{cr}, \mu_{cr} + \Delta_{cr}]\) in order to reduce the computation complexity.

According to (1), the final FSM can be easily obtained by employing the three conspicuity maps (2), (6), and (7). For example, the FSMs of four video frames, namely Claire, Carphone, Foreman, and News, are calculated, and presented in Fig. 4, respectively. It is observed that most face regions in the test frames achieve the large saliency values with respect to other objects. Using the facial saliency map, the first face candidate map \(f(x, y)\) can be obtained by setting

\[
f(x, y) = \begin{cases} 
1, & \text{if } S(x, y) \geq \tau_1 \\
0, & \text{otherwise}
\end{cases}
\] (9)

where \(\tau_1\) is a threshold.

Once the candidate face areas are obtained, we can then begin the binary map regularization. The morphological operators [43], i.e., dilation and erosion, are employed. In our work, we use the square structuring elements which have the width of two and three pixels for erosion and dilation.

2.2. Geometric Verification Model

After the previous stage, one or more potential candidate face areas may be obtained. As illustrated in the News sequence in Fig. 4, some background regions (i.e., two dancers) are probably detected as candidate regions due to the similar saliency features. In this section, we will construct a simple geometric model relative to shape, size, and variance to validate these candidates, and then remove most of false regions.

Let \(NoP_k\) denote the number of pixels of the current candidate face area \(k\). Assume that the ideal face height for the given pixels is denoted by \(a\), we then set the ideal face width to \(\frac{2a}{3}\). Thus, we have \(a = \sqrt{\frac{3NoP_k}{2}}\). Let \(h\) and \(w\) represent the real height and width
of this area, respectively. $V_L$ and $V_R$ are used to denote the standard deviations of the left and right parts. The geometric model is defined as

$$G(k) = \text{sym}(NoP_k - F\text{min}) \cdot \left\lfloor \left\lceil \frac{(h - a)(w - \frac{2a}{3})}{NoP_k} \right\rfloor \cdot \left\lfloor 1 - \frac{V_L}{V_L + V_R} \right\rfloor, \right.$$  

(10)

where $F\text{min}$ denotes the minimal size of displaying a face structure. The allowable sizes of 80 $\times$ 48 and 50 $\times$ 50 for $F\text{min}$ had been employed in the literature [4] and [7]. We found that the size of 12 $\times$ 10 is capable of describing the outline of a face structure. From (10), we can see that three features are taken into account to perform the geometric verification. The first is the minimal pixels, which can be employed to remove most of small candidate areas. The second is relative to the shape, which is used to eliminate the regions without the face contour characteristic. The third corresponds to the variance, which is based on the assumption that the human face in the head-and-shoulder type video appears in or near front view rather than the side-view of face initially. If the constraint of (10) is satisfied, we will label the current region as the candidate face area, namely

$$G_{ID}(k) = \begin{cases} 
1, & \text{if } G(k) \leq \tau_2 \\
0, & \text{otherwise} 
\end{cases}$$  

(11)

where $\tau_2$ is a threshold. According to the performance of a lot of experiments, the value in (0.003~0.009) is recommended, which can provide better constraint result for the candidate faces selection.

### 2.3. Eye-map Verification

In this stage, we will verify the face detection result generated by the previous stage, i.e., for the case of $G_{ID}(k) = 1$, and to remove those false alarms caused by objects with similar skin color and facial geometric structure. Among various facial features, eye and mouth are the most prominent features for recognition and estimation of 3D head pose[5]. Next, we will concentrate on the detection of eye location from the marked face areas. In [5], an eye map based on the observation of chrominance values is constructed, and is used to locate the eye area in color images. Unlike the high resolution color image, the video conferencing sequences usually do not have enough chrominance information to generate good eye-map by the method in [5]. For example, Fig. 5(a) shows the constructed EyeMap (from (1) and (2) in [5]) for the first Carphone video frame. It is observed that the eye locations cannot be easily derived from the obtained eye map. In order to construct an effective eye map, we first manually segmented the previous face data into eye patches. Different lighting conditions are considered in 314 experimental samples. Since it is difficult to get the accurate region around eyes, the rectangular window is used to extract the eyes. The color distributions are presented in Fig. 6. From the statistical results, it can be observed that values of $Cb > 100$ and $Cr < 155$ with respect to the skin-tone color distribution are found around the eyes. Based on the above analysis, we then propose a simple method to construct the corresponding eye map.

Let $m_{Y'}(k)$, $m_{Cb}(k)$, and $m_{Cr}(k)$ denote the mean values of the color components $Y'$, $Cb$, and $Cr$ for the $k$th candidate face area, respectively. The eye map can be written as

$$\text{Eyemap} = \text{sym}(Y' - m_{Y'}(k)) \AND \text{sym}(Cb - \max\{100, m_{Cb}(k)\}) \AND \text{sym}(Cr - \min\{155, m_{Cr}(k)\}),$$  

(12)
Figure 5. The construction result of EyeMap of Carphone video: (a) by [5], (b) by our method.

Figure 6. The histograms of eye skin colors. (a) Luminance Y, (b) Cb, and (c) Cr.

Figure 7. The neighborhood structure of the candidate eye area.
where, the symbols ‘\( \text{max} \)’ and ‘\( \text{min} \)’ denote the maximum and minimum operation. \( \text{sym}() \) represents the inverse function of \( \text{sym}() \). In addition, it should be noted that the luminance \( Y' \) is the normalized value that has the same dimension as the chrominance components. Fig. 5(b) shows the eye map obtained by our proposed method. As compared with the method [5], we can see that the locations of eyes can be clearly observed. For each candidate eye area, we first select the corresponding rectangle region with the size of \( h_e \times w_e \) in the luminance space. Then the neighborhood pixels with a constant offset \( e \) are taken into account. As shown in Fig. 7, we compare the mean value of the eye area with those of its neighborhood, i.e., \( N_1, N_2, N_3, \) and \( N_4 \). If a low value and the corresponding appropriate location (i.e., the upper part of the current candidate area but not the boundary) are detected, we will declare that there exists an eye structure in the candidate face region.

2.4. Adaptive Boundary Correction

After the analysis of the previous stages, we obtain the final candidate face areas. In this section, we will segment the face region using an adaptive boundary correction method. The flow chart of this stage is shown in Fig. 8.

For each face area, we first calculate the mean value \( m_i \) and standard derivation \( \sigma_i \) of the chrominance component \( i \) (i.e., \( i = Cb \) or \( Cr \)). We then label all the boundary points. For each point \((j, k)\), we compute the global distance \( D(j, k) \) and the local distance \( d_w^R(j, k) \), which are defined as

\[
D(j, k) = \sqrt{(Cb(j, k) - m_{Cb})^2 + (Cr(j, k) - m_{Cr})^2}
\]

\[
d_w^R(j, k) = \sqrt{(Cb(j, k) - m_{Cb'})^2 + (Cr(j, k) - m_{Cr'})^2 + (Y'(j, k) - m_{Y'w'})^2}
\]

where \( w \) denotes a window of \( w \times w \) that is centered at the current point \((j, k)\). \( R \) is used to indicate the area properties in the window, i.e., \( R = F \) (Candidate face region) or \( R = B \) (Background region).

In addition, in order to improve the accuracy of the boundary correction and avoid the occurrence of the false detection especially in the case of blurry region, we next introduce the constraint of boundary curvature. It is known that the contour of human face usually exhibits elliptical shape and smaller curving level. To reduce computation complexity, we employ a simple scheme to approximately depict the curvature feature at the boundary point \((j, k)\) in our work. Assume that \((j_n^+, k_n^+)\) and \((j_n^-, k_n^-)\) denote the boundary points which have the offset \( n \) relative to the initial point \((j, k)\) in counterclockwise and clockwise directions, respectively. The corresponding curvature value \( cv_n(j, k) \) and direction \( cd_n(j, k) \) are defined as

\[
cv_n(j, k) = \frac{\sqrt{(j_n^+ - j_n^-)^2 + (k_n^+ - k_n^-)^2}}{\sqrt{(j_n^+ - j_n^-)^2 + (k_n^+ - k_n^-)^2} + \sqrt{(j_n^+ - j_n^-)^2 + (k_n^+ - k_n^-)^2}}
\]

\[
cd_n(j, k) = \begin{cases} 1, & \text{if } f([\frac{j_n^+ + j_n^-}{2}], [\frac{k_n^+ + k_n^-}{2}]) = 1 \\ -1, & \text{otherwise} \end{cases}
\]

where the symbol \([\cdot]\) denotes the downward truncation operation. If the condition (17) is satisfied (i.e., \( T_1(j, k) = 1 \)), we will remove this point, and then select its neighborhood.
Input candidate face area

Calculate the mean and variance of Cb and Cr

Label the boundary point

For each boundary point

T+1 = 1?

Y

Delete this point and generate new boundary point

N

Add this neighbors as new boundary point

Y

T2 of its neighbor pixels = 1?

N

Done for all boundary pixels?

N

Contour Extraction

Y

Figure 8. Flow chart of the presented contour correction method.
pixels in the candidate region as new boundary points. On the other hand, if its neighborhood pixel \((j_s, k_s)\) that does not belong to the marked area also tends to exhibit similar feature as the face area, i.e., the condition \((18) \left( T_2(j_s, k_s) = 1 \right)\) is satisfied, we then extend this pixel as a new boundary point. Finally, we employ dilation and erosion with a square structuring element to perform the binary image regularization.

\[
T_1(j, k) = \begin{cases} 
1, & \text{if } \{ D(k, j) > 2\sqrt{\sigma^2_{Cb} + \sigma^2_{Cr}} \text{ or } \frac{d^b_{n}(j, k)}{d^e_{n}(j, k)} < 1 \} \\
0, & \text{otherwise} 
\end{cases} 
\quad \text{(17)}
\]

\[
T_2(j_s, k_s) = \begin{cases} 
1, & \text{if } D(j_s, k_s) < 3\sqrt{\sigma^2_{Cb} + \sigma^2_{Cr}} \text{ and } \frac{d^b_{n}(j_s, k_s)}{d^e_{n}(j_s, k_s)} \geq 1 \\
0, & \text{otherwise} 
\end{cases} 
\quad \text{(18)}
\]

3. Tracking Based Face segmentation

The face segmentation in the successive frames is achieved by the tracking based method. Many algorithms have been proposed to track human faces in a video [46]-[49]. Generally, much computational complexity will be involved in order to achieve good tracking performance. In our work, the face tracking is used as the initial step for the faces segmentation in the subsequent frames. A simple and efficient tracking method is developed. The flowchart of our algorithm is shown in Fig. 9. The major steps in this stage are the boundary matching and connection, which is enclosed in the rectangular block in Fig. 9. In this section, a novel boundary saliency map is first constructed to determine the face boundary. Then a connective technique between two key points is employed to extract this region.

3.1. Boundary Saliency Map

The first step for tracking the face region is the projection of the information at the previous frame \((i-1)\) into the current frame \(i\) [2]. By applying the motion estimation technology, we can easily obtain the current position for each candidate face area. Here, a simple coarse-to-fine motion estimation technique is employed to obtain the projected position. We first perform the full-search motion estimation in the sampling space of half the spatial resolution within a given window. Then, the search in finer scale (i.e., eight neighboring points) will be performed for the best matched position. The sum of absolute difference (SAD) is used as the similarity measure between two regions. Generally, we can use large search window to deal with different degrees of face moving which can track the face over successive frames. In our work the 16 x 16 search window is considered in the motion estimation stage, which can cover those range of face moving for the head-and-shoulder videos.

Without loss of generality, we only take one face region into account in the following analysis. Assume \(f_{i-1}\) denotes the candidate face region of the \((i-1)\)th frame in the video sequence. \(E_l, l = 1, 2, ..., L\) corresponds to its boundary points. We then use \(f_i'\) with the boundary points \(E_i'\) to denote the projected face area by motion compensation in the \(i\)th frame. Let \((x, y)\) be the position of a pixel in the current frame \(i\). We define the
Figure 9. Flowchart of the proposed tracking based segmentation algorithm.

boundary saliency map (BSM) as

\[ Bsm(x, y) = Q_1(x, y) \cdot Q_2(x, y) \cdot Q_3(x, y), \]  \hspace{1cm} (19)  

\[ Q_1(x, y) = \exp\left\{ -\frac{(Cr(x, y) - m_{Cr_{i-1}})^2}{8\sigma^2(Cr_{i-1})} + \frac{(Cb(x, y) - m_{Cb_{i-1}})^2}{8\sigma^2(Cb_{i-1})} \right\}, \]  \hspace{1cm} (20)  

\[ Q_2(x, y) = N\{Es(Y'(x, y))\} \cdot N\{Es(Cr(x, y))\}, \]  \hspace{1cm} (21)  

\[ Q_3(x, y) = \frac{c}{c + \min\{\sqrt{(x - x_{E'_l})^2 + (y - y_{E'_l})^2}, for \ l = 1, 2, ..., L\}}, \]  \hspace{1cm} (22)  

where \( Q_1, Q_2, \) and \( Q_3 \) denote the "conspicuity maps" corresponding to the color, edge, and position components, respectively. Here, \( m_{Cr_{i-1}} \) and \( m_{Cb_{i-1}} \) correspond to the mean values of color components of the segmented face area in frame \( i - 1 \), while \( \sigma(\cdot) \) denotes the standard derivation. The symbol \( N \) is a normalized operator. \( Es(\cdot) \) at position \( (x, y) \) denotes the edge strength obtained by Canny algorithm [44]. In (22), \( (x_{E'_l}, y_{E'_l}) \) represents the coordinates value of boundary point \( E'_l \). Variable \( c \) is used to adjust the influence level caused by the distance of the current detected pixel. Generally, we can set a small value to \( c \) if the current frame is in a static or small motion level, while the larger value can be employed for the fast motion face region. From (19), we can see that if the edge points that are close to the projected boundaries have the same color feature as the segmented face region in the previous frame, the large BSM values will be achieved.
for these points. On the contrary, those background points tend to have smaller BSM values due to the unconspicuous color or position features. For example, Fig. 10(b) and (d) show the obtained BSMs of the 2th and 65th frames in the Claire and Carphone video sequences (namely Fig. 10(a) and (c)), respectively. It is observed that most of the facial boundary points exhibit the larger values distinctly than other background points. In addition, we find that if we take all the pixels of the current frame into account, the computation complexity will increase significantly. In fact, most background pixels that are away from the face contour will tend to hold very small BSM values. Therefore, in order to save computational load, we only consider the neighboring pixels of the facial boundary points in the following matching procedure, i.e., within a window of $w \times w$. Assume that the pixels in the given window have the same values of position conspicuity. The corresponding BSM can be rewritten as

$$Bsm(x, y) = Q_1^w(x, y) \cdot Q_2^w(x, y)$$  \hspace{1cm} (23)

where $Q_1^w$ and $Q_2^w$ denote the the "conspicuity maps" calculated from the given window of $w \times w$.

### 3.2. Boundary Extraction

After the calculation of BSM, the next step is to find the boundary points in the current frame $i$ and segment the corresponding face region. Here, we use two stages to realize the extraction process.

The first is the boundary matching, which aims to find the best points with maximal BSM values for the projected facial boundary. Assume $\hat{E}_l^i, l = 1, 2, \ldots, L$ denote the estimation of the positions for the previous boundary points $E_l$ in the current frame. We have

$$\hat{E}_l^i = \arg \max_{(x, y) \in w(E_l^i)} \left\{ \sum_{i=-M}^{+M} (Bsm(x_i, y_i) \cdot g(x_i, y_i)) \right\}^{p(x, y)},$$  \hspace{1cm} (24)

$$g(x_i, y_i) = \frac{1}{\sqrt{\pi(2M + 1)}} \exp\left(-\frac{i^2}{2M + 1}\right),$$  \hspace{1cm} (25)
\[ p(x, y) = 1 - \frac{1}{2M + 1 + J} \sum_{i=-M}^{+M} \sum_{j \in N_{x_i,y_i}^E} Q_i^1(x_i, y_i)(1 - \frac{1}{2M + 1 + J} \sum_{i=-M}^{+M} \sum_{j \in N_{x_i,y_i}^B} Q_i^1(x_i, y_i)) \]

where the \((x_{+i}, y_{+i})\) and \((x_{-i}, y_{-i})\) have the same description as \((x_{+i}^+, y_{+i}^+)\) and \((x_{-i}^-, y_{-i}^-)\), respectively, which have been defined in Section 2.4. \(w(E_i^j)\) denotes the window that is centered at \(E_i\). Equation (25) is used to eliminate the impact of the singular point by means of computing the weighting mean value of the BSM at \((x, y)\). The exponential variable \(p\) depicts the boundary feature in the given window. In (26), \(N_{x_i,y_i}^F\) and \(N_{x_i,y_i}^B\) denote the neighborhood of the point \((x_i, y_i)\), which belong to the projected facial and background areas, respectively. Similarly, \(J\) and \(\bar{J}\) are used to indicate the number of pixels in the corresponding areas. From (19), it is observed that large BSM value usually corresponds to the facial boundary. But in some cases of the lighting variation, the similar BSM values may be observed for the actual and other edges. We can see that \(p\) can provide an approximate evaluation of boundary position at the pixel \((x, y)\) within the search region. If \(p(x, y)\) holds a small value, it means that the current detected points tend to be the actual facial boundary. Otherwise, it should be classified as the point in the face region or in the background. In addition, it should be noted that variable sizes are employed for the parameters \(M\) and \(N_{x_i,y_i}\) in our work, i.e., 10 and 3 for the static level while 5 and 5 for motion case, respectively. The reason is that the values of small \(M\) and large \(N_{x_i,y_i}\) are more suitable to satisfy the requirement of the facial boundary deformation during the motion instance. On the contrary, in order to avoid false detection in the case of the small boundary movement, more boundary points and the small window size are employed to impose the strictly curvature and moving step constraints during the search procedure.

The second is the postprocessing stage, which is used to connect two break boundary points. This procedure is only applied to the nonclosed contour after the boundary matching. It is known that if the deformation of the face contour appears during the face movement, such as the face approaching to the camera, some new contour points will then be generated, which results in the incontinuity of the matching boundary. Here, a simple linear connection method between two break boundary points is employed to form a closed loop for the face contour. Finally, for the closed contour, we use the filling-in technique to extract the corresponding area.

4. Segmentation Experiments

In this section, we evaluate the effectiveness of the proposed face segmentation algorithm. For the evaluation, we use some standard MPEG test sequences in QCIF (176 × 144) format (namely Akyio, Claire, Carphone, and Foreman) as the input videos.

4.1. FSM-based Segmentation

The segmentation results on the first frames for four video sequences (i.e., Akyio, Claire, Carphone, and Foreman) are shown in Fig. 11, respectively. The first row is the original input frames, while the second corresponds to the segmented results. We can see that the face areas in the images are segmented successfully for all test sequences. Most of the background regions with similar skin-tone color are removed from the detected images, while reserving the prominent face saliency area. Only a minimal amount of false alarms,
such as the inner boundary of the safety helmet for the Foreman frame image, are detected as face region due to the similar skin-color feature. Additionally, it is observed that some of the hair on the head has been detected as skin, which should not be considered as a significant problem in object-based coding scheme [7].

The comparison results between the existing skin-color segmentation methods are illustrated in Fig. 12. The result depicted in Fig. 12(a) is obtained by using the method [3]. It should be noted that the horizontal and vertical scanning processes in the geometric correction stage are skipped because of the small sample size for QCIF image. Namely, the operation that any group of less than four connected pixels with the value of one will be eliminated and assigned to zero for CIF image [3] is not employed in our experiment. Fig. 12(b) shows the skin color segmentation result by using the method in [5]. This method employs color correction in RGB color space before skin detection, which is based

Figure 11. First row: Four test MPEG video sequences (Akyio, Claire, Carphone, Foreman). Second row: The corresponding segmentation results based on our proposed method.

Figure 12. Segmentation results for the first frame in Carphone sequence: (a) Chai [3], (b) Hsu [5], (c) Jones [9], and (d) Habili[7].
on the assumption that the dominant bias color always appears as "real white". If the number of reference-white pixels [5] exceeds 100 and the average color is not similar to skin tone, the color adjustment will be performed. From the result, we find that a lot of background pixels are also detected as skin region. Similar evaluation result of this method can be found in [45]. Fig. 12(c) shows the detection result by using the mixture of Gaussian model of skin color described in [9]. The skin color segmentation result for the Carphone sequence by the method [7] can be found in Fig. 12(d). It is observed that high miss detection are generated for the last two algorithms. In addition, we applied the postprocessing method in [3] to the output skin color map obtained from the linear classifier [4], the similar segmentation result is observed as compared to Fig. 12(a). From a lot of experiments, we have found that the skin color detection results will change significantly with various image sources or different lighting conditions for the existing skin color detection schemes.

To further validate our proposed algorithm, other three human objects with different lighting and noisy conditions are fist tested. One image with indoor lighting, which is selected from MPEG-7 News1 video sequence, is shown in Fig. 13(a), while the face area appears near the top left of the image instead of the center position. The image with outdoor lighting and large illumination variation, which is selected from [52], is depicted in Fig. 13(b). A noisy image (i.e., movie poster from WWW) is shown in Fig. 13(c). The corresponding segmentation results are illustrated in Fig. 13(d), (e) and (f),
Figure 14. Face segmentation results of our method for Berkeley database.

Figure 15. Face segmentation results for interactive graph cut.
respectively. It is found that the face region has been segmented very well, and most of the facial pixels are correctly identified with only a minimal amount of false alarms for the boundary points. More examples are shown in Fig. 14, where the first row of images are selected from the widely used The Berkeley Segmentation Dataset and Benchmark [50]. The segmentation results of our method are given in the second row. From the experimental results, we can see that better performance can be achieved by using our proposed face segmentation method.

We also perform the comparisons of our method with the well known graph cut method [36], which is considered as a successful bilayer segmentation method. The results are shown in Fig. 15, where the first row denotes the user’s marks of the face (red lines) and the background (blue lines). The second row is the corresponding results. We can see our attention based automatic face segmentation performs equally well with the distinct advantage of being unsupervised.

4.2. BSM-based Segmentation for Tracking Procedure

After the segmentation of the first frame, the face regions in successive frames are obtained by the tracking based segmentation procedure. We apply our proposed algorithm to the sequence of Carphone, which is characterized by the mild-fast motion of the face and the background in the video scene. Some original frames are shown in the first and the third rows of Fig. 16. The final segmentation results are shown in the second and final rows, respectively. As opposed to the static video sequence, such as Akyio or Claire, Carphone sequence represents a scene with the deformable face contour due to the fast head movement. It is observed that the face contour appears significant difference with the varying postures of the human object. The segmentation result indicates that our proposed method is capable of capturing the deformable face boundary effectively in the moving condition. Moreover, as shown in Fig. 16, the appearance of the hand in the 178th frame, which also has the skin color feature, does not alter the segmentation of the tracking face region.

Another experiment of our segmentation algorithm is performed on two handset video sequences in QCIF format. Some representative frames are shown in the first and second rows of Fig. 17. Unlike the previous sequences, such as Carphone, we use the low quality compressed handset videos as the input sequences, which have very low signal-to-noise ratio. The first sequence given in the first row of Fig. 17 includes the fast movement of the head and shoulder with the static background. In the second sequence, the moving background is considered as well as varying light conditions. Their corresponding segmentation results are given in the second and the fourth row of Fig. 17. We can see that good performance can be achieved using our proposed segmentation algorithm for those videos with noise and varying illumination conditions.

Next, we would like to analyze the behavior of the proposed tracking based segmentation algorithm in case of errors in the object segmentation result. Firstly, the appearance of the false alarms in the previous segmentation is taken into account. As shown on the top left of Fig. 18, we manually add some of background areas to the original segmentation result. The following figures on the top row show the segmentation results obtained by using our proposed BSM based segmentation algorithm in the tracking procedure. It is observed that the false parts disappear gradually after several frames segmentation. The
Figure 16. Segmentation results for *Carphone* video sequence after face tracking. The first and the third rows: original input frames. The second and the final rows: segmentation results.
Figure 17. Face segmentation results for two handset videos.

Figure 18. Example of robustness of the proposed BSM algorithm in case of errors in the segmentation. Top: the appearance of false alarms. Bottom: the appearance of miss alarms.
boundary points in the person’s clothing are rapidly converged to the correct boundary. From the result shown in the 9th frame, we can see that most of the false alarms are corrected by our proposed method. Secondly, the case of miss alarms is also tested in our experiments. In the same way, we manually take off some facial pixels from the original segmentation, which is shown in the left figure at the bottom of Fig. 18. We can see that a large part of face region is removed as the miss alarms. However, the face contour is correctly identified after the following frames segmentation. The corresponding face boundary is again resumed to the correct status, which can be found in the result of Frame 14. From the evaluation of this sequence, it is shown that even if the segmentation does not present a satisfied result for the following tracking stage, the proposed BSM based segmentation algorithm is capable of correcting these errors effectively, and keeps the better segmentation results.

Now, we will concentrate on the computational complexity of the proposed FSM-based face segmentation algorithm. In our method, the facial saliency map, geometric model verification, and eye-map detection should be taken into account in order to obtain the correct facial region. However, it can be seen that the computational load of the last two stages is mainly imposed on the output candidate face areas with only smaller size with respect to the original image. Generally, in order to decrease the computational operation, we can directly skip the last two stages if only one candidate face area is detected. In addition, for the tracking based segmentation algorithm, i.e., BSM based segmentation, most of the computation will be consumed in the stage of motion estimation process. Only a small number of operations are required for the following BSM computation. In addition, it should be noted that the proposed method employs the adjustable threshold to obtain the candidate face areas from the FSM. Generally, the smaller \( \tau_1 \) is, the more false alarms are allowed, and vice versa. By varying \( \tau_1 \), we can control the amount of false alarms and false dismissals allowed in the stage of the FSM. Based on a large amount of experiments, a dynamic threshold \( \tau_1 \) (namely \( (0.3 \sim 0.5)S_{\text{max}} \)), which can provide better candidate outputs, is reasonable for pixels classification. Some examples are illustrated in Fig. 19. Here, \( S_{\text{max}} \) denotes the maximal value of the facial saliency map. Of course, it is difficult to obtain the accurate face boundary only by the determination of the threshold. Nevertheless, it can be improved by using the subsequent stages of the geometric verification model and adaptive boundary correction. That is, the false face candidates can be successfully removed by employing the following verification models. Similarly, those false and miss alarms in the selected face regions can be eliminated by using the final adaptive boundary correction.

Moreover, we also test the selection of the local window size on the segmentation performance. Fig. 20 shows the segmentation results of a frame for *carphone* video using different local windows, i.e. 4-neighborhood (diamond window), 3×3, 5×5 and 7×7. Similar results can be obtained for those small windows, such as 4-neighborhood and 3×3.

In addition, compared with Viola and Jones method, which is used to detect the human face based on the AdaBoost learning method, it is observed that there are distinct differences between our segmentation method and Viola and Jones method. Firstly, Our method concentrates on the face segmentation based on the attention model rather than the face location in the input image. The obtained facial saliency map not only provides...
Figure 19. Experimental results of akyio, carphone, and foreman videos for different $\tau_1$. The values of $\tau_1$ from left to right: 0.3, 0.45, 0.5, 0.6.

Figure 20. Experimental results of carphone video for different local windows. Left to right: 4-neighborhood, $3\times3$, $5\times5$ and $7\times7$. 
the candidate face position, but also achieves the coarse segmentation. However, the Viola and Jones method focuses on finding the face location using the detection window at many scales. In order to extract the human face, another segmentation method has to be employed. Secondly, Viola and Jones method is designed for the grey-level image without the consideration of the skin-color information. In our work, the chrominance components are the important cues in our facial saliency model. Finally, the skin color and the gradient values are usually employed in pre-filtering the input image to speed up the face detection for the Viola and Jones method.

5. Conclusion

In this paper, an effective face segmentation algorithm is proposed from the facial saliency point of view. The basic idea is motivated by the observation that face area usually exhibits its special color distribution and structural feature, especially for typical ”head-and-shoulder” video. Therefore, a novel facial saliency map is first constructed based on the color, structure, and position information. Then a geometric model and facial features are used to verify the candidate face areas. Based on the segmented result, an effective boundary saliency map is then constructed, and applied to the subsequent tracking process. Unlike other approaches, the proposed method puts heavy emphasis on the facial saliency features for the head-and-shoulder videos. Experimental results were obtained by applying the proposed method to a large number of video sequences. It is shown that our method is capable of segmenting the face quite effectively. In our the future work, we hope to design more robust and efficient face segmentation algorithm in the case of lighting variations and complex background, and make it suitable for videophone and video-conferencing applications.

Acknowledgment

This work was partially supported by the Shun Hing Institute of Advanced Engineering and the Research Grants Council of the Hong Kong SAR, China (Project CUHK415505).

REFERENCES


41. CVL FACE DATABASE: http://www.lrv.fri.uni-lj.si/facedb.html