Segmentation of the Face and Hands in Sign Language Video Sequences Using Color and Motion Cues

Nariman Habili, *Member, IEEE*, Cheng Chew Lim, *Senior Member, IEEE*, and Alireza Moini, *Member, IEEE*

***Abstract*—We present a hand and face segmentation methodology using color and motion cues for the content-based representation of sign language video sequences. The methodology consists of three stages, namely skin-color segmentation, change detection, and face and hand segmentation mask generation. In skin-color segmentation, a universal color-model is derived and image pixels are classified as skin or nonskin based on their Mahalanobis distance. We derive a segmentation threshold for the classifier. The aim of change detection is to localize moving objects in a video sequences. The change detection technique is based on the *F* test and block-based motion estimation. Finally, the results from skin-color segmentation and change detection are analyzed to segment the face and hands. The performance of the algorithm is illustrated by simulations carried out on standard test sequences.**

***Index Terms*—Change detection, sign language, skin-color segmentation, video segmentation.**

I. Introduction

SIGN LANGUAGE is a visual language used by deaf and hearing-impaired people to communicate. A common device used by deaf people for distant communication is the text telephone. The text telephone is an assistive technology that allows direct communication over a telephone line using standard telephone equipment. This technology is also useful for individuals who have difficulty communicating using a standard telephone due to speech impairment. Unfortunately, the speed of text conversation is limited by typing ability, and is at least 10 times slower than that of sign language [1]. Moreover, sign language is the first language of many pre-lingually deaf individuals, and its speed is comparable to that of normal speech [2]. Therefore, affordable sign language video communication would greatly benefit the deaf community.

The characteristics of a sign language video sequence are different to those of a typical head-and-shoulder sequence. The additional challenges inherent in the transmission of sign language video include the presence of fast motion and the need for smooth motion perception.

The quality requirements of sign language video for distant communication have been studied by Hellström [1]. Hellström observed that for accurate comprehension, the frame rate of sign language video should be at least 20 frames per second at CIF resolution. As well as video, sign language communication systems are also required to process and transmit text and audio information [3]. People who have impaired hearing, but are not completely deaf, sometimes use voice as well as sign language to communicate. Accordingly, the transmission of sign language video over low-bit rate channels would require significant compression.

Sign language video sequences can be significantly compressed by employing content-based coding strategies [4]. Using content-based coding, video sequences are typically segmented into different objects (termed the video object plane in the MPEG-4 standard) which may be independently coded and transmitted. More resources are allocated to the perceptually important objects. As well as improved coding efficiency, content-based representation enables other functionalities, such as improved error-robustness, and scalability. In sign language video, the perceptually significant objects are the face and hands [1]. The objective of this paper is to present a methodology to segment a person's face and hands in a sign language video sequence.

Our hand and face segmentation methodology consists of three stages. In the first stage, image pixels are classified as skin or nonskin to yield a skin detection mask (SDM). The skin-color distribution is modeled as a bivariate normal distribution and the image pixels are classified based on their Mahalanobis distance. In the second stage, the F test is employed to localize moving objects in the image sequence and yield a change detection mask (CDM). The third stage involves the fusion of the SDM and the CDM to generate a face and hand segmentation mask (FHSM). The block diagram of the methodology is shown in Fig. 1.

In the section to follow, the skin-color segmentation algorithm is presented. Section III presents the change detection technique, and the FHSM generation method is discussed in Section IV. Experimental results are presented in Section V and the paper is concluded in Section VI.

II. Skin-Color Segmentation

Skin-color segmentation is feasible because the human skin has a color distribution that differs significantly, although not

Manuscript received February 7, 2002; revised October 29, 2002. This paper was recommended by Associate Editor J.-N. Hwang.

N. Habili is with iOmniscient Pty Ltd., Chatswood, 2067 Australia (e-mail: nhabili@ieee.org).

C. C. Lim is with the School of Electrical and Electronic Engineering, The University of Adelaide, Adelaide 5005, Australia (e-mail: cclim@eleceng.adelaide.edu.au).

A. Moini is with Silverbrook Research Pty Ltd., Balmain 2041, Australia (e-mail: alireza.moini@silverbrookresearch.com).

Digital Object Identifier 10.1109/TCSVT.2004.831970

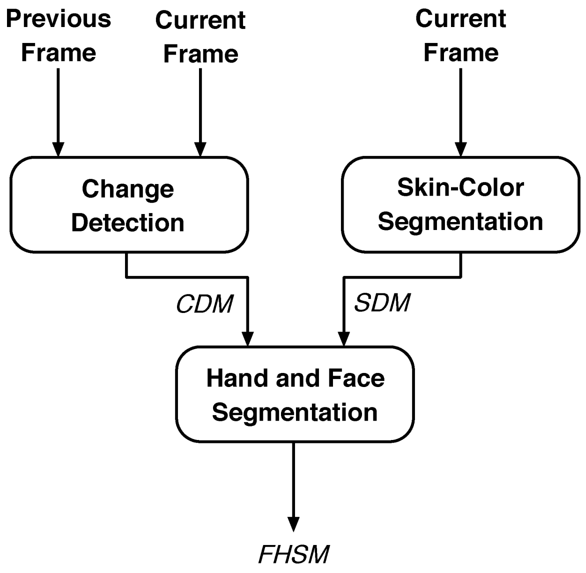


Fig. 1. Block diagram of the face and hand segmentation methodology.

entirely, from those of the background objects [5]. Skin segmentation has been mainly employed for face segmentation in digital images and video. Some major uses for face segmentation include content-based representation in MPEG-4, face recognition, and face tracking. Skin-color segmentation is usually performed using the chrominance components of image pixels and not the luminance component. The reason for this is twofold. 1) by utilizing the chrominance components only, skin-color segmentation algorithms will remain relatively invariant to changes in brightness (e.g., shadow versus no shadow), and 2) it has been widely reported that apparent differences in skin-color among different races (e.g., dark skin versus fair skin) are characterized by the difference in the brightness of the color, which is governed by the luminance component of light and not the chrominance components [5]–[8]. Another reason is that, by considering the chrominance components only, the feature space is reduced from three-dimensional (3-D) to two-dimensional (2-D), thus reducing the computational complexity of the segmentation algorithm.

There are some limitations to any skin-color segmentation algorithm that must be considered. Accurate results are obtained only if the contrast between skin and background is significant. Note that in the context of hand and face segmentation, other parts of the body, including clothing, are also considered as background. Static background regions with a similar color to that of skin do not pose a serious problem, since they can be identified by change detection. However, parts of clothing that undergo motion and have a similar color to that of skin, may pose problems. As well as poor color contrast, there are other limitations of color segmentation when an input image is taken under some particular lighting conditions.

It is important to choose an appropriate color space for skin-color segmentation. Color spaces used in the past have included the YCbCr [5], [6], HSV [7], [9], CIE L* a* b* [4], YES [10], normalized RGB [11], and the RGB [8] color spaces. Note that in the RGB color space, the luminance component and the chrominance components are not decoupled. Also, the feature space is 3-D for RGB as opposed to 2-D for chrominance skin-color segmentation.

We have considered the YCbCr color space in our research since it is effective in modeling the human skin-color. Also, since digital video is stored and coded in the YCbCr color space, our algorithm does not require color space conversion. Conversion from one color space to another is computationally expensive. In the YCbCr color space, Y reflects the luminance and is scaled to a range of 16 to 235. The chrominance components, Cb and Cr, are scaled versions of color differences B-Y and R-Y, respectively. Cb and Cr have a range of 16 to 240, inclusive.

The block diagram of our skin-color segmentation algorithm is shown in Fig. 2. The portion within the dashed line depicts the classifier training process. Unlike most other skin-color segmentation algorithms, our algorithm is intended to work on a range of skin types and lighting conditions.

A. Generation of the Skin-Color Model

To generate the skin-color model, we manually segmented training images into skin and nonskin classes. The training images were obtained from the world wide web, and were of different subjects (with different ethnicities, e.g., European, Asian, and African), body poses, background complexities, and lighting conditions (indoor, outdoor, and studio). Fig. 3 shows the distribution of the CbCr components of the skin training pixels in the CbCr plane.

1) The Skin-Color Model: Let $\mathbf{c}$ denote the feature vector by the Cb and Cr components of a pixel (i.e., $\mathbf{c} = [\text{Cb Cr}]^T$), and $\mathbf{c}$ is in a 2-D Euclidean space $\mathbf{R}^2$, called the feature space. The skin and nonskin classes are denoted by ω_S and $\omega_{\bar{S}}$, respectively. The skin-color distribution in the feature space is modeled as a bivariate normal distribution, with $\boldsymbol{\mu}_S$ as the mean vector and $\boldsymbol{\Sigma}_S$ the covariance matrix. The components of $\boldsymbol{\mu}_S$ and $\boldsymbol{\Sigma}_S$ are estimated from the skin training pixels.

The quantity d in

$$d^2 = (\mathbf{c} - \boldsymbol{\mu}_S)^T \boldsymbol{\Sigma}_S^{-1} (\mathbf{c} - \boldsymbol{\mu}_S) \qquad (1)$$

is the *Mahalanobis distance* from $\mathbf{c}$ to $\boldsymbol{\mu}_S$. It follows from (1) that the contours of constant density are ellipses for which d is constant. The principal axes of these ellipses are given by the eigenvectors of $\boldsymbol{\Sigma}_S$, and the eigenvalues of $\boldsymbol{\Sigma}_S$ determine the length of these axes. Equation (1) provides a mapping from the 2-D feature space to a one-dimensional (1-D) distance space. The value of d is related to the probability that a given pixel belongs to class ω_S. A small value of d indicates a high skin pixel probability and vice-versa.

B. Generation of the SDM

In this section, we describe the classification method employed to classify pixels as skin or nonskin. Prior to pixel classification, a median filter [12] is applied to the Cb and Cr components of each image or frame to be segmented. In median filtering, the gray-level of each pixel is replaced by the median of the gray-levels in the neighborhood of that pixel. The filter removes outliers in skin regions, while preserving edges. The size of the kernel was chosen based on an empirical study of sign

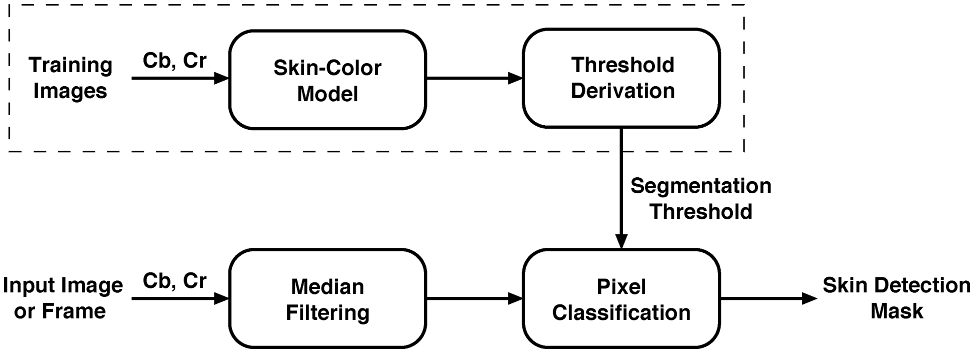


Fig. 2. Block diagram of the skin-color segmentation algorithm.

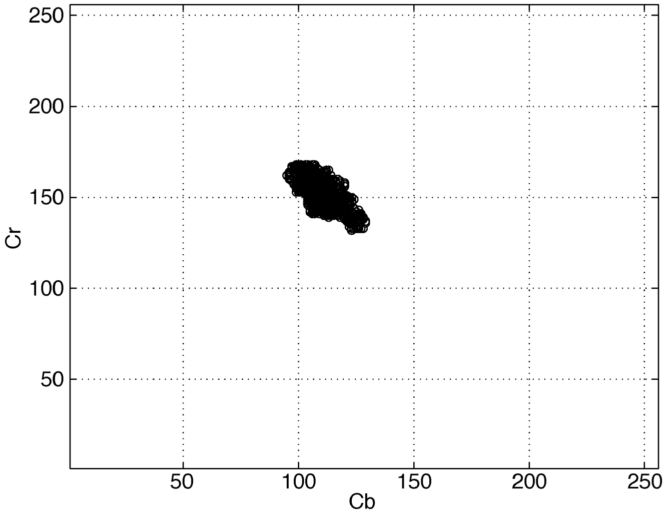


Fig. 3. Skin training pixels in the CbCr plane.

language video frames. Our experimental data suggests that if the size of the kernel is large (i.e., 7 × 7 or larger for a frame of size QCIF), the face and hand objects would merge if they are close to each other. This is not desirable for face detection. Alternatively, a small kernel size (i.e., 3 × 3) would be ineffective in eliminating outliers. We have found a kernel size of 5 × 5 pixels to be effective in eliminating outliers without merging many nearby skin-color regions.

1) Pixel Classification: With the skin-color distribution in the CbCr plane modeled as a bivariate normal distribution, the contours of constant density are ellipses of constant Mahalanobis distance to $\boldsymbol{\mu}_S$. To classify a pixel as skin or nonskin, we measure the Mahalanobis distance between the feature vector of the pixel and $\boldsymbol{\mu}_S$. By comparing the Mahalanobis distance against a predetermined threshold, the pixel is classified as skin if the distance is not greater than the threshold, otherwise it is classified as nonskin.

More specifically, the SDM (SDM) is defined as

$$SDM(x,y,k) = \begin{cases} 1, & \text{if } d_{x,y,k} \leq \tau \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

where τ is the segmentation threshold. With $\hat{\boldsymbol{\mu}}_S$ and $\hat{\boldsymbol{\Sigma}}_S$ as the sample mean vector and covariance matrix, respectively, the Mahalanobis distance $d_{x,y,k}$ is defined as

$$d^2_{x,y,k} = (\mathbf{c}_{x,y,k} - \hat{\boldsymbol{\mu}}_S)^T \hat{\boldsymbol{\Sigma}}_S^{-1} (\mathbf{c}_{x,y,k} - \hat{\boldsymbol{\mu}}_S). \tag{3}$$

The parameter $\boldsymbol{c}_{x,y,k}$ denotes the feature vector of a pixel in frame k, at spatial location (x,y).

2) Derivation of the Segmentation Threshold: We define the probability of error as

$$P_{\text{error}} = P_M(\theta)P(\omega_S) + P_F(\theta)P(\omega_{\bar{S}}), \tag{4}$$

where θ is a threshold, $P_F(\theta)$ is the probability of false alarm, $P_M(\theta)$ is the probability of miss, $P(\omega_S)$ and $P(\omega_{\bar{S}})$ denote the a priori probabilities of the skin and nonskin classes, respectively, and $P(\omega_S) + P(\omega_{\bar{S}}) = 1$.

The probabilities P_M and P_F are evaluated using the skin and nonskin class training data. For the set of training images I_j, $j = 1, \ldots, J$

$$P_M(\theta) = \frac{1}{N_S} \sum_{j=1}^{J} \sum_{(x,y)\in I_j} \alpha(\mathbf{c}_{x,y,j}, \theta) \tag{5}$$

$$P_F(\theta) = \frac{1}{N_{\bar{S}}} \sum_{j=1}^{J} \sum_{(x,y)\in I_j} \beta(\mathbf{c}_{x,y,j}, \theta) \tag{6}$$

where $\alpha(\mathbf{c}_{x,y,j}, \theta)$ and $\beta(\mathbf{c}_{x,y,j}, \theta)$ are defined as

$$\alpha(\mathbf{c}_{x,y,j}, \theta) = \begin{cases} 1, & \text{if } \mathbf{c}_{x,y,j} \in \omega_S \text{ and } d_{x,y,j} > \theta \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

$$\beta(\mathbf{c}_{x,y,j}, \theta) = \begin{cases} 1, & \text{if } \mathbf{c}_{x,y,j} \in \omega_{\bar{S}} \text{ and } d_{x,y,j} \leq \theta \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

The parameter $\mathbf{c}_{x,y,j}$ denotes the feature vector of a pixel in training image I_j at spatial location (x,y). The Mahalanobis distance $d_{x,y,j}$ is defined as

$$d^2_{x,y,j} = (\mathbf{c}_{x,y,j} - \hat{\boldsymbol{\mu}}_S)^T \hat{\boldsymbol{\Sigma}}_S^{-1} (\mathbf{c}_{x,y,j} - \hat{\boldsymbol{\mu}}_S). \tag{9}$$

The problem is how to derive a suitable segmentation threshold. If $P(\omega_S)$ is known, the segmentation threshold can be derived by minimizing (4), i.e.

$$\tau = \arg \min_{\theta} \left(P_M(\theta)P(\omega_S) + P_F(\theta)P(\omega_{\bar{S}}) \right). \tag{10}$$

To solve (10), we plot θ against P_{error} and find the minimum P_{error}. The skin class prior $P(\omega_S)$ is simply the number of skin pixels in an image divided by the number of pixels in an image. Using the dimensions of a QCIF frame, the number of image pixels is 176 × 144 = 25 344. This leaves only the number of skin pixels to be estimated. For sign language video, let us assume a face size of 50 × 50 pixels and a hand size of 25 × 25 pixels [4]. This results in $P(\omega_S) = 0.15$ and

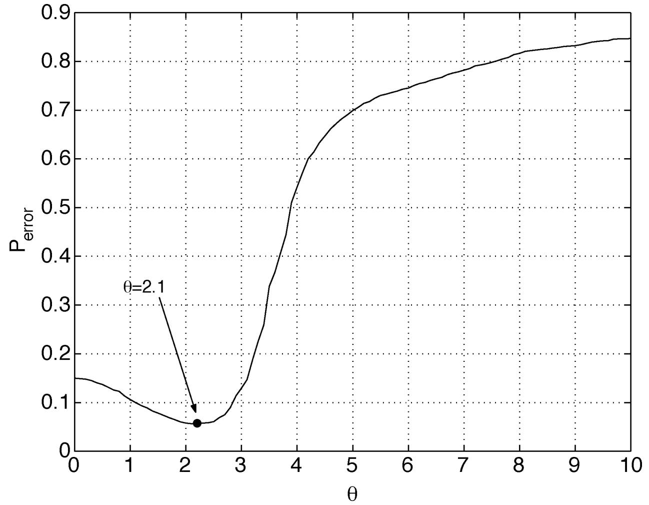


Fig. 4. Probability of classification error versus θ for $P(\omega_S) = 0.15$ and $P(\omega_{\bar{S}}) = 0.85$.

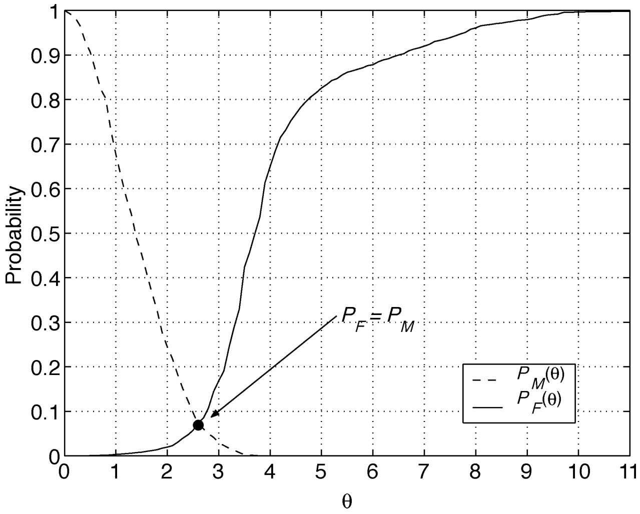


Fig. 5. The probability of miss and false alarm as a function of θ.

$P(\omega_{\bar{S}}) = 0.85$. Fig. 4 shows the probability of error versus θ for $P(\omega_S) = 0.15$. The minimum probability of error is indicated in the graph and corresponds to $\theta = 2.1$ (i.e., $\tau = 2.1$).

If $P(\omega_S)$ is unknown, the minimax test [13] can be employed to derive the segmentation threshold. Since $P(\omega_S) + P(\omega_{\bar{S}}) = 1$, (4) reduces to

$$P_{\text{error}} = (P_M(\theta) - P_F(\theta))\, P(\omega_S) + P_F(\theta). \quad (11)$$

To minimize the maximum possible P_{error}, θ should be set to make the coefficient of $P(\omega_S)$ in (11) zero, regardless of $P(\omega_S)$. That is, we need to solve

$$P_M(\theta) = P_F(\theta) \quad (12)$$

for θ (similar to [10]). This choice of θ would render P_{error} independent of $P(\omega_S)$, and hence guarantee that the maximum error probability is minimized regardless of any changes in $P(\omega_S)$. In order to find the segmentation threshold based on the minimax test, we need to show the miss and false alarm probabilities as a function of θ (Fig. 5). The point where $P_M(\theta) = P_F(\theta)$ is indicated in the graph and corresponds to $\theta = 2.6$.

III. Statistical Change Detection

We derive temporal information by segmenting a video sequence into moving (i.e., foreground) and stationary (i.e., background) regions. We are only interested in determining which regions in a frame have changed due to motion, and this information is provided by a change detector. Change detection methods are generally less computationally expensive than motion estimation and optical-flow methods, and would therefore promote real-time segmentation.

If the background is stationary (i.e., no camera panning or zooming) and there are no changes in the image acquisition parameters (i.e., camera focus etc.), taking the luminance-level (i.e., gray-level) difference between two frames is an effective way to detect changed regions with respect to the previous frame. The gray-level difference frame (DF) between frames $F(x,y,k)$ and $F(x,y,k-1)$ is defined as

$$\text{DF}_{k,k-1}(x,y) = F(x,y,k) - F(x,y,k-1). \quad (13)$$

Assuming that the illumination remains constant from one frame to the next, pixel locations where $\text{DF}_{k,k-1}(x,y)$ differ from zero indicate objects that are moving or changing their shape. Moreover, the intensity at each pixel in the current frame is either a displaced value from the previous frame (i.e., a moving pixel), the same value as in the previous frame (i.e., a stationary pixel), or an uncovered-background value (i.e., an uncovered-background pixel). Furthermore, unless the objects are textured, only the boundaries of moving objects can be observed, and not the objects themselves. In sign language video, the moving eyes, nose, mouth and fingers add texture to the face and hands.

Non-zero differences can also occur due to camera or quantization noise. Figs. 6(a) and (b) show frames 13 and 14 of the *Silent* sequence, respectively. The binary difference frame (BDF) is shown in Fig. 6(c), and is obtained by allocating a binary "1" to nonzero differences, i.e., $|\text{DF}_{k,k-1}(x,y)| > 0$, and a binary "0" to zero differences. Since the background is stationary, one would expect the background to consist entirely of binary "0" pixels, however due to noise, the background appears very noisy. The image noise is usually modeled as additive white Gaussian noise [14]. The objective of change detection is to distingubetween temporal variations caused by noise from those caused by object motion.

Aach *et al.* [15] proposed several statistical change detection methods. In one proposal, the camera noise is modeled as a zero-mean normal distribution, and the chi-square test is used to detect changed regions in the DF. The chi-square test requires the variance of the background population σ_0^2 in DF.

A recursive method that automatically estimates the background population variance was proposed by Ziliani [16]. First, σ_0^2 is estimated for the used camera, and then change detection is performed on the DF. The areas in the DF that are declared as background are used to estimate a new σ_0^2. This allows σ_0^2 to be automatically adapted to the Gaussian noise. Unfortunately, it is difficult to estimate the variance of the Gaussian camera noise for the used camera system.

Noting the difficulty of estimating the variance of the background population, Kim *et al.* [17] proposed a change detection

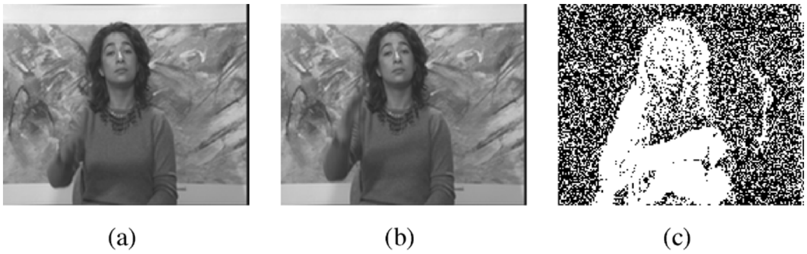


Fig. 6. Example of frame differencing. Frames 13 (a) and 14 (b) of the *Silent* sequence, and (c) BDF.

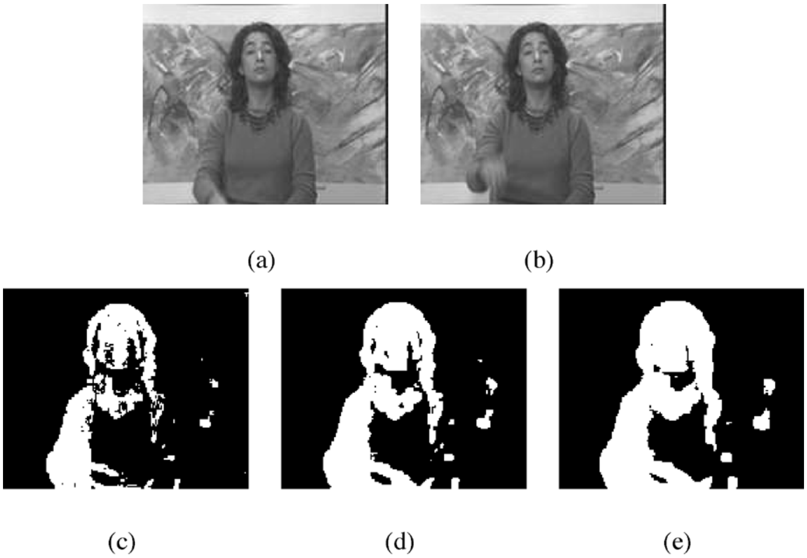


Fig. 7. The effect of increasing the size of W. (a)–(b) Frames 14 and 15 of the *Silent* sequence. (c) $W = 3 \times 3$ pixels. (d) $W = 5 \times 5$ pixels. (e) $W = 7 \times 7$ pixels.

method based on the F test. Instead of the background population variance, the F test requires a sample variance of the difference pixels in a background region. The problem is how to find difference pixels in a background region of DF. To make the change detection technique automatic in the sense that no manual manipulation is required, we employ block-based motion estimation to find difference pixels in a background region of DF.

A. Change Detection Based on the F Test

In order to make change detection less sensitive to noise, thresholds are usually calculated based on the statistics of a small region in DF, rather than the difference level of a single pixel. Therefore, the hypothesis test is based on the statistical properties of the samples in a square observation window, W. To form a CDM, a binary “1” is allocated to the center pixel in W if the null hypothesis (i.e., the hypothesis that the difference pixels in W are drawn from the background population) is rejected, otherwise a binary “0” is allocated. The use of a window for thresholding corresponds to applying a low-pass filter to the DF. This will cause a blurring effect in the CDM because changes in the observation window are attributed to the center pixel in W, regardless of precisely where the changes occur. Fig. 7shows the effect of increasing the size of W. As the size of W is increased, the blurring effect becomes more profound. We have found that the blur does not adversely affect the outcome of our segmentation results if a window size of 3 × 3 pixels is used (for frames of size QCIF).

Another parameter that must be considered is the significance level, α. The significance level is the probability of detecting background pixels as foreground. The value of α is critical, since too high a value will swamp the CDM with spurious changes, while too low a value will suppress significant changes. A significance level of 0.01 was found to be appropriate.

1) The F Test: The F test is used to test if the standard deviations of two populations are equal. Suppose that the difference pixels in W are drawn from a normal population with variance σ_1^2. The F hypothesis test is defined as

$$\begin{aligned} H_0 &: \sigma_0^2 = \sigma_1^2, \\ H_1 &: \sigma_0^2 < \sigma_1^2. \end{aligned} \tag{14}$$

The null hypothesis, H_0, implies that σ_1^2 and σ_0^2 are equal, while the alternative hypothesis, H_1, implies that σ_0^2 is less than σ_1^2. The hypothesis test is based on the notion that the intensity variation induced by a moving object is greater than that of the background due to the higher intensity gradient at the edge and within a moving object.

Let S_0^2 (respectively, S_1^2) be the estimator of σ_0^2 (respectively, σ_1^2), and n_0 (respectively, n_1) be the sample size. If the null hypothesis is true, then the ratio

$$F = \frac{S_1^2}{S_0^2} \tag{15}$$

has an F-distribution with $n_0 - 1$ and $n_1 - 1$ degrees of freedom. Since hypothesis test (14) is an upper one-tailed test, the null hypothesis is rejected if $F > F_{(\alpha, n_1-1, n_0-1)}$, where $F_{(\alpha, n_1-1, n_0-1)}$ is the critical value of the F-distribution with $n_0 - 1$ and $n_1 - 1$ degrees of freedom, and a significance level of α. Note that the F test does not require the background population variance.

The sample variance of the background population must be derived from an area in DF that does not contain any moving regions. To this end, we advocate a method based on block-based motion estimation. The procedure is described in the next section.

B. Estimation of the Background Sample Variance

Block-based motion estimation is a core component of the H.261, H.263, MPEG-1, MPEG-2, and MPEG-4 video coding standards. Given a reference frame (frame $k-1$) and an $N \times N$ block in the current frame (frame k), the objective of block-based motion estimation is to seek the $N \times N$ block in the reference frame that best matches (according to a given cost function) the characteristics of the block in the current frame. The relative displacement between a block in the current frame and a block in the reference frame is described by a motion vector (v_x, v_y). To reduce the computational complexity, the search is usually restricted to a search region around the original location of the block in the current frame. The full search algorithm exhaustively searches the entire search window in the reference frame to find the optimal match. However, this is at the expense of higher computational cost. Various other sub-optimal block-based motion estimation techniques have been proposed [18], [19] that aim to reduce the computational cost.

Let the search range be $\pm R$ pixels in both horizontal and vertical directions. With a stepsize of one pixel, the total number of candidates matching blocks in the search window is $(2R+1)^2$. Our block-based motion-detection strategy is conceptually simple. We first choose an $N \times N$ block in frame k, and define a search window in frame $k-1$. An $N \times N$ block located at the upper border of a frame is usually chosen, since we do not expect motion at the upper border. Note that if the block is at the border of the frame, the number of candidate matching blocks is reduced. Since the probability of motion at the upper border is low, the value of R is set to 7. Usually a value of $R = 15$ is chosen for head and shoulder type scenes [20], however a large R value would substantially increase the computational cost of motion estimation. The full search algorithm is then employed to find the best matching block in frame k. The matching of the blocks can be quantified according to various error performance criteria including the minimum mean square error and the minimum mean absolute error. The mean absolute error (MAE) was chosen as the matching cost function because it is a popular choice for hardware implementation [14]. The MAE is defined as:

$$\mathrm{MAE}(v_x, v_y) = \frac{1}{N^2} \times \sum_{(v_x, v_y) \in \mathcal{B}} |F(x, y, k) - F(x + v_x, y + v_y, k-1)| \quad (16)$$

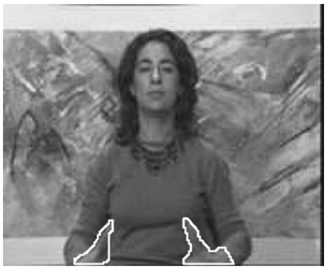


Fig. 8. Frame 218 of the *Silent* sequence, showing the hand objects.

where $\mathcal{B}$ denotes an $N \times N$ block, for a set of candidate motion vectors (v_x, v_y). The estimate of the motion vector is taken to be the value of (v_x, v_y) that minimizes the MAE, i.e.,

$$[\hat{v}_x \; \hat{v}_y]^T = \arg \min_{(v_x, v_y)} \mathrm{MAE}(v_x, v_y). \quad (17)$$

If the motion vector is zero, i.e., $[\hat{v}_x \; \hat{v}_y]^T = [0 \; 0]^T$, we assert that the corresponding $N \times N$ block in DF does not contain any foreground pixels. However, if the motion vector is nonzero, we assume that the corresponding block in DF contains foreground pixels. If the motion vector is nonzero, another block in frame k is chosen, and the above procedure repeated. Since a block was chosen at the upper border of a frame where the probability of motion is low, we found that motion estimation was often performed only once. The procedure rarely had to be performed more than twice.

The assumption of a common displacement (v_x, v_y) for all pixels in the block implies that a local smoothness constraint is imposed on the motion vector field. The local smoothness constraint is only satisfied for small block sizes. The choice of the dimensions of the block is the result of tradeoffs among the three conflicting requirements [20]:

1) a small value for N is preferable, since the smoothness constraint would be easily met at this resolution;
2) a small value for N would reduce the reliability of the motion vector (v_x, v_y), since few pixels would participate in the matching process;
3) fast algorithms for finding motion vectors are more efficient for larger values of N.

Based on experimental data, a block size of $N \times N = 8 \times 8$ pixels (i.e., 64 samples) was found to be appropriate. For a significance level of 0.01, with $n_0 - 1 = 8$ and $n_1 - 1 = 63$ degrees of freedom, the critical value of F is $F_{(0.01, 63, 8)} = 5.02$. The following is a summary of the change detection procedure based on the F test.

1) Compute the background sample variance S_0^2 using block-based motion estimation.
2) Compute the sample variance S_1^2 of the difference pixels in observation window W.
3) Compute the test statistic (15), where the degrees of freedom are 8 and 63.
4) If $F > 5.02$, the center pixel in W is declared a foreground, otherwise it is declared a background.
5) Repeat steps 2–4 for all pixel locations in DF.

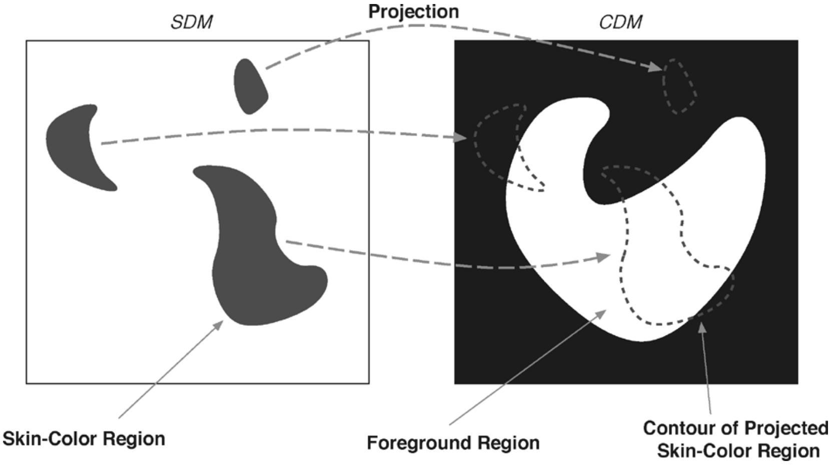


Fig. 9. SDM projected onto the CDM.

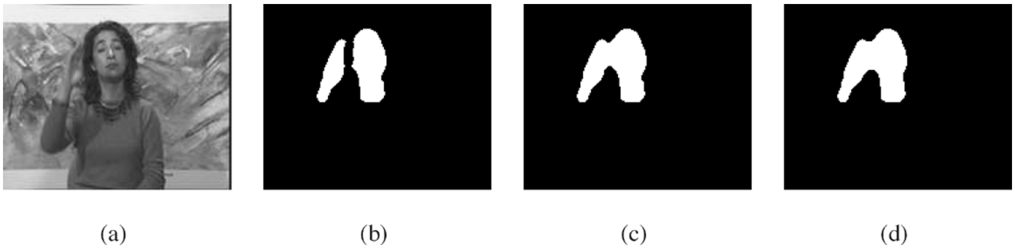


Fig. 10. Effect of varying the size of the structuring element. (a) Frame 16 of the *Silent* sequence. (b) Structuring element with a diameter of 7 pixels. (c) Structuring element with a diameter of 9 pixels. (d) Structuring element with a diameter of 11 pixels.

IV. Generation of the Face and Hand Segmentation Mask

To generate the FHSM, color and motion information from the SDM and the CDM are utilized. We first note that the skin-color regions in a frame are localized in the SDM. Also, as noted previously, sign language is characterized by the motion of the arms, the hands, and the face (including the eyes and the mouth).

Moving objects entail intensity changes between consecutive frames, which are marked as foreground in the CDM. Thus, the CDM can be used to separate the moving skin-color regions from the stationary skin-color regions in the SDM. The FHSM is a binary map where a binary "1" indicates a moving skin-color region, and a binary "0" indicates a background pixel. The FHSM is analogous to the alpha map in the MPEG-4 standard. The postprocessing stages are described below.

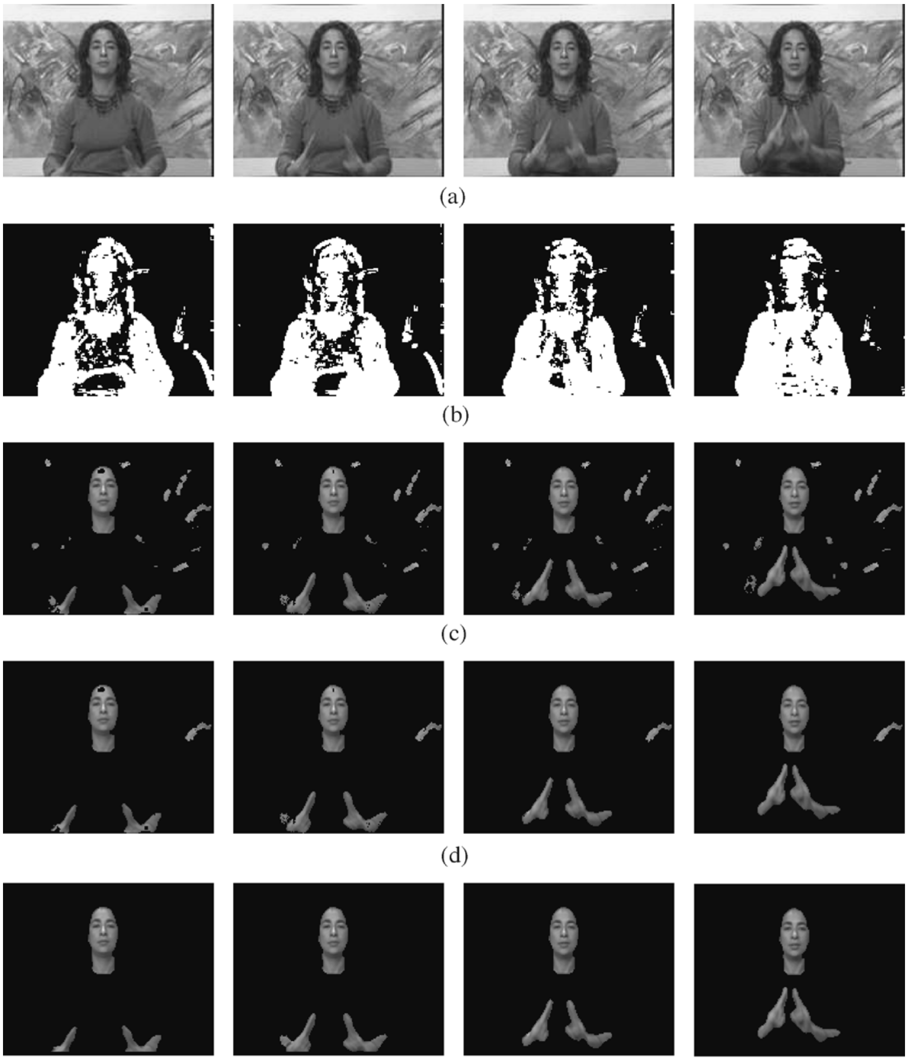
To generate the FHSM, connected components labeling [21] is first performed on the SDM to find the connected components (with 8-neighborhood connectivity). If the size of a connected component is less than a certain threshold, we assume that it is a false alarm and eliminate it from the SDM. To determine a suitable threshold, we must examine the size of the face and hand objects in the sequence. We note that the size of the face object remains fairly constant throughout a sign language sequence, however the size of the hand objects vary depending on their position. Fig. 8 shows frame 218 of the *Silent* sequence. The size of the right hand is 243 pixels, and the size of the left hand is 117. After an extensive analysis of different hand positions and their corresponding sizes in sign language video sequences, we found that a suitable threshold is 100 pixels. This threshold value was derived empirically. Thus, if the size of a connected component in the SDM is below 100 pixels, it is assumed to be a false alarm and discarded.

To identify the moving skin-color regions and hence eliminate skin-color regions in the stationary background, the skin-color regions in the SDM are projected onto the CDM, as shown in Fig. 9. When the majority of a connected component in the SDM is covered by a foreground region in the CDM, the connected component is declared as a moving skin-color region. We expect the moving skin-color regions to represent either the face or hand objects, however the FHSM may also contain false alarms due to the following reasons:

1) moving skin-color regions due to clothing or hair with a size greater than 100 pixels.
2) skin-color regions in the uncovered background. The uncovered background areas are marked as changed in the CDM. To overcome this, the uncovered background areas must be identified, e.g., [22].
3) shadows produced by moving objects will entail intensity variations that are marked as changed. This may result in false alarms if a skin-color region coincides with a foreground region associated with shadows.

To overcome this, shadow cancellation strategies can be employed, e.g., [23].

The face object may contain holes due to the presence of the eyes, mouth, and eyebrows. In addition, "bright spots" and shadows may also produce holes in the face and hand objects. To fill these holes, we employ the *morphological closing operator* [21]. Morphological closing has the effect of filling small and thin holes, connecting nearby regions, and generally smoothing the boundaries of regions without significantly changing their areas.



(a)

(b)

(c)

(d)

(e)

Fig. 11. Segmentation results for the *Silent* video sequence. (a) Original frames, (b) CDMs. (c) SDMs. (d) SDMs after connected components analysis and elimination. (e) FHSMs.

We found that a large structuring element for the closing operator would merge nearby regions. Fig. 10(a) shows frame 16 of the *Silent* sequence, and Figs. 10(b), (c) and (d) show the result of the morphological closing operator applied to the corresponding FHSM with circular structuring elements of varying diameters. Nearby (although distinctly separate) hand and face objects merge if a structuring element with a diameter of nine pixels or higher is applied. We also found that a small structuring element would not effectively fill holes in some cases. A circular structuring element with a diameter of 7 pixels was found to be most effective. Circular structuring elements tend to promote the formation of smooth and curved object boundaries, which closely resemble those of real objects.

V. Experimental Results

This section presents the simulation results to evaluate the performance of the face and hand segmentation methodology. For simulation we used standard MPEG test sequences in the QCIF format.

The performance of the hand and face segmentation methodology is evaluated both qualitatively and quantitatively. To quantitatively measure the accuracy of the proposed methodology, each frame was manually segmented into skin and nonskin classes. The manually segmented images serve as a reference to which the automatically segmented images are compared. The false alarm rate (R_F) and miss rate (R_M) are evaluated for each image by

$$R_F = \frac{\text{Number of false alarm pixels}}{\text{Number of nonskin pixels}} \times 100 \qquad (18)$$

$$R_M = \frac{\text{Number of miss pixels}}{\text{Number of skin pixels}} \times 100 \qquad (19)$$

R_F and R_M are expressed as a percentage.

A. Skin-Color Segmentation

The skin-color segmentation results for six consecutive frames of the *Silent* (frames 218 to 223) and *Irene* (frames 210 to 215) are shown in Figs. 11(c) and 12(c). For both sequences, the face and hands of the subjects have been segmented reasonably well, however some background regions have also been

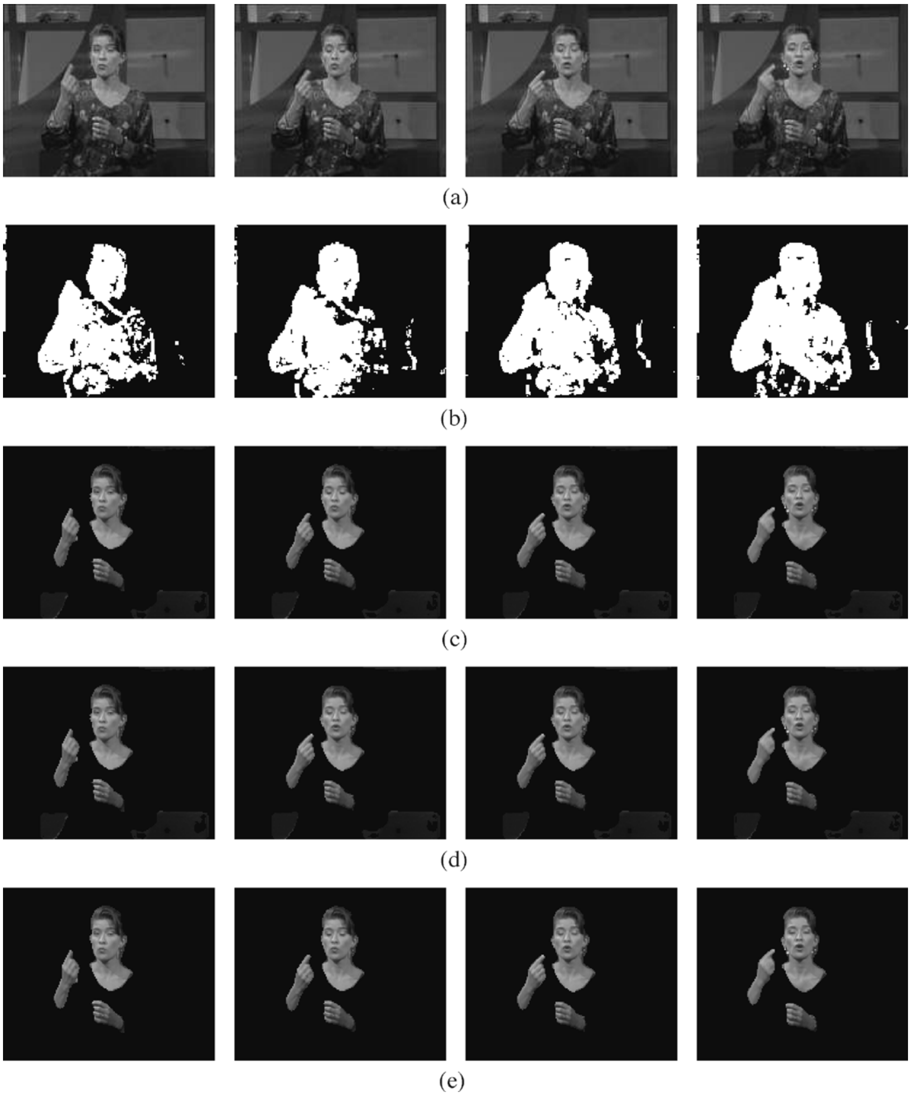


Fig. 12. Segmentation results for the *Irene* video sequence. (a) Original frames. (b) CDMs. (c) SDMs. (d) SDMs after connected components analysis and elimination. (e) FHSMs.

detected as skin. The false alarms present in the SDMs are due to similar skin and background color characteristics.

The skin-color segmentation algorithm was also tested on other standard video sequences that contain skin regions. Fig. 13 shows the results for six consecutive frames of the *Carphone* sequence. The face regions have been segmented well, with only a minimal amount of false alarms.

B. Statistical Change Detection

The CDMs for frames 218–223 of the *Silent* sequence and frames 210–215 of the *Irene* sequence are shown in Figs. 11(b) and 12(b). The bright areas indicate the foreground regions. Since the test statistic is the ratio of the variance estimate of the difference pixel in the observation window to the variance estimate of the background, the value of the test statistic would be large when the window passes over moving objects. Sign language is characterized by the motion of the mouth, eyes, face, and hands. Since these regions are textured, we would expect the hand and face objects to be marked as foreground in the CDM. The foreground regions cover the face and hand objects reasonably well, with little residual noise (i.e., false alarms) present in the CDMs.

Note that only some parts of the chest area of the subject in the *Silent* sequence are marked as changed. This is due to insufficient texture in the moving regions. It is difficult to detect intensity changes in moving objects if there is insufficient texture. On the other hand, the clothing of the subject in the *Irene* sequence is textured, and consequently the chest area of the subject is marked as changed. The foreground regions on the right hand side of the CDMs (i.e., right side of the moving person) are due to shadow.

Change detection results for six consecutive frames of the "Hall Objects" sequence are shown in Fig. 14. The occlusion regions of the moving person have been marked as changed, however the person's chest and briefcase are marked as unchanged due to insufficient texture.

C. Generation of the FHSM

Figs. 11(d) and 12(d) show the SDMs (after connected components labeling and elimination). The FHSMs are shown in

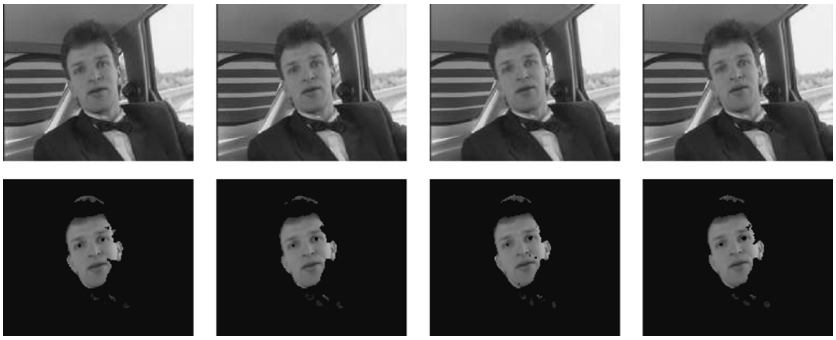


Fig. 13. Skin-color segmentation results for the *Carphone* sequence.

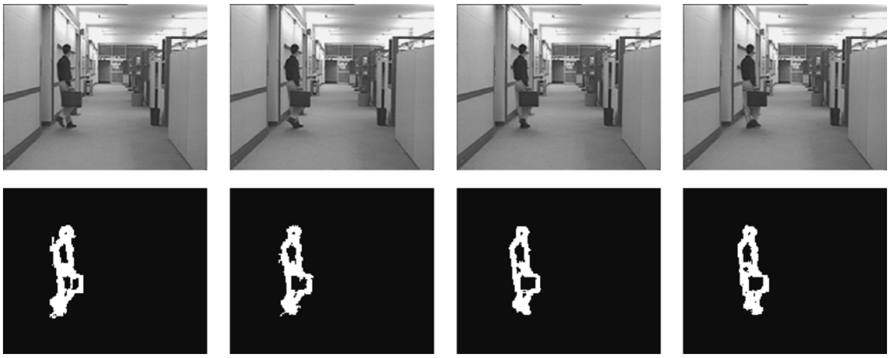


Fig. 14. Change detection results for the *Hall Objects* sequence.

TABLE I
AVERAGE MISS AND FALSE ALARM RATES FOR THE *SILENT* AND *IRENE* SEQUENCES

	Average R_F		**Average R_M**	
Sequence	*SDM*	*FHSM*	*SDM*	*FHSM*
Silent	7.2	5.9	4.4	1.6
Irene	3.5	2.9	6.6	3.2

Figs. 11(e) and 12(e). After connected components analysis and elimination, most of the false alarms in the SDMs of the silent sequence have been successfully discarded since their size is less than 100 pixels. For the *Irene* sequence, there are two false alarm regions that remain after connected components analysis and elimination. This is because their size is larger than 100 pixels. However, these false alarm regions reside in the stationary background, and are therefore eliminated by the use of motion information. We observe that some of the hair of the subject in the *Irene* sequence has been detected as skin. We do not expect this to pose a significant problem in any content-based coding strategies.

We now quantitatively evaluate the effect of SDM/CDM fusion, connected components analysis, and the morphological closing operator. The average R_F and R_M values for the *Silent* and *Irene* sequences are given in Table I. Note that for both sequences, the average R_F and R_M values for the FHSMs are lower than those for the SDMs. This is understandable since the use of motion information and connected components labeling has effectively eliminated most of the false alarms (i.e., decrease in R_F), and the morphological closing operator has filled the holes in the face and hand objects (i.e., decrease in R_M).

Figs. 15 and 16 show the segmentation results obtained by using the method described in [24]. The average R_F and R_M values for the *Silent* sequence are 8.6% and 12.1%, respectively, and 5.7% and 13.2%, respectively, for the *Irene* sequence. Note the high miss and false alarm rates for both sequences as compared to those stated in Table I.

VI. CONCLUSION

The face and hand segmentation methodology presented consists of three steps, namely skin-color segmentation, change detection, and face and hand extraction. The skin-color distribution in the CbCr plane is modeled as a bivariate normal distribution. Image pixels are classified as skin if the Mahalanobis distance between their feature vectors and the mean vector of the skin class is less than a predetermined segmentation threshold. The segmentation threshold was derived by minimizing the probability of error. The skin color regions in a frame are indicated in a SDM.

In the second stage, frames were segmented into foreground and background regions to yield a CDM. The change detection method is based on the F test and block based motion estimation. The F test compares the sample variance of the difference pixels in an observation window with the sample variance of background pixels. To evaluate the background sample variance, we advocated a method based on block-based motion estimation.

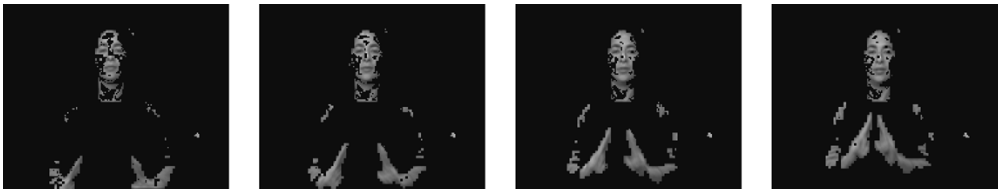


Fig. 15. Results for the *Silent* sequence obtained using the method described in [24].

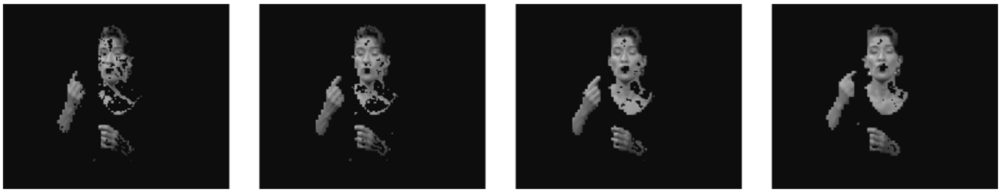


Fig. 16. Results for the *Irene* sequence obtained using the method described in [24].

In the preprocessing stage, connected components labeling was employed to label all connected components in the SDM. Connected components of 100 or less pixels were eliminated, since our experiments suggested that these regions are false alarms. The moving skin-color regions were then identified in the SDM by using motion information. Finally, the morphological closing operator was applied to the remaining regions to fill any holes.

Experimental results indicate that the technique is capable of segmenting the hands and face quite effectively. The algorithm allows the flexibility of incorporating additional techniques to enhance the results. For example, work is currently under way to generate the FHSMs by combining the SDMs and CDMs using thresholds based on statistics.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their useful comments.

REFERENCES

[1] G. Hellström, "Quality measurement on video communication for sign language," in *Proc. 16th Int. Symp. Human Factors in Telecommunications*, Oslo, Norway, May 1997, pp. 217–224.

[2] *Supplement 1 to Series H: Application Profile—Sign language and lip-reading real-time conversation using low bit-rate video communication*, ITU-T, May 1999.

[3] G. Hellström, "Total conversation, the key to equal status in telecommunication," in *Proc. Int. Conf. Computers Helping People With Special Needs*, Karlsruhe, Germany, July 2000, pp. 303–312.

[4] R. P. Schumeyer, "A video coder based on scene content and visual perception," Ph.D. thesis, University of Delaware, Newark, 1998.

[5] D. Chai and K. N. Ngan, "Face segmentation using skin-color map in videophone applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 551–564, June 1999.

[6] H. Wang and S.-F. Chang, "A highly efficient system for automatic face region detection in MPEG video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 615–628, Aug. 1997.

[7] K. Sobottka and I. Pitas, "A novel method for automatic face segmentation, facial feature extraction and tracking," *Signal Process. Image Commun.*, vol. 12, pp. 263–281, June 1998.

[8] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," in *Proc. Computer Vision and Pattern Recognition*, Ft. Collins, Colorado, June 1999, pp. 274–280.

[9] X. Zhu, J. Yang, and A. Waibel, "Segmenting hands of arbitrary color," in *Proc. IEEE 4th Int. Conf. Automatic Face and Gesture Recognition*, Grenoble, France, Mar. 2000, pp. 446–453.

[10] E. Saber, A. M. Tekalp, R. Eschbach, and K. Knox, "Automatic image annotation using adaptive color classification," *Graph. Models Image Process.*, vol. 58, pp. 115–126, Mar. 1996.

[11] L. M. Bergasa, M. Mazo, A. Gardel, M. A. Sotelo, and L. Boquete, "Unsupervised and adaptive Gaussian skin color model," *Image Vision CompuT.*, vol. 18, pp. 987–1003, 2000.

[12] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1992.

[13] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic, 1990.

[14] A. M. Tekalp, *Digital Video Processing*. Upper Saddle River, NJ: Prentice–Hall, 1995.

[15] T. Aach, A. Kaup, and R. Mester, "Statistical model-based change detection in moving video," *Signal Process.*, vol. 31, pp. 165–180, 1993.

[16] F. Ziliani, "Spatio-temporal image segmentation: A new rule-based approach," Ph.D. thesis, Swiss Federal Institute of Technology, Lausanne, Switzerland, 2000.

[17] M. Kim, J. G. Choi, D. Kim, H. Lee, M. H. Lee, C. Ahn, and Y.-S. Ho, "A VOP generation tool: automatic segmentation of moving objects in image sequences based on spatio-temporal information," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1216–1226, Dec. 1999.

[18] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 313–317, June 1996.

[19] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 369–377, Aug. 1998.

[20] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architectures*. Norwell, MA: Kluwer, 1995.

[21] R. Haralick and L. Shapiro, *Computer and Robot Vision*. Reading, MA: Addison-Wesley, 1992, vol. 1.

[22] K. Matthews and N. M. Namazi, "A Bayes decision test for detecting uncovered-background and moving pixels in image sequences," *IEEE Trans. Image Process.*, vol. 7, pp. 720–728, May 1998.

[23] P. L. Rosin and T. Ellis, "Image difference threshold strategies and shadow detection," in *Proc. 6th British Machine Vision Conf.*, Birmingham, U.K., 1995, pp. 347–356.

[24] S. Akyol and P. Alvardo, "Finding relevant image content for mobile sign language recognition," in *Proc. Int. Conf. Signal Processing, Pattern Recognition, and Applications*, Rhodes, Greece, July 2001, pp. 48–52.

Nariman Habili (S'97–M'02) received the B.Sc. and B.Eng. (with honors) degrees in biomedical engineering from the Flinders University of South Australia, in 1997 and 1998, respectively, and the Ph.D. degree in electrical engineering from the University of Adelaide, Adelaide, Australia, in 2002.

From February 2002 to February 2003, he was a Research Associate at the University of Western Australia, and a Visiting Researcher at the Nanyang Technological University, Singapore. In March 2003, he joined iOmniscient Pty Ltd., Sydney, Australia, as a Research and Development Engineer. His research interests include computer vision, pattern recognition, image/video processing, and video surveillance.

Cheng Chew Lim (M'81-SM'02) received the B.Sc. (with honors) degree in electronic and electrical engineering in 1977, and the Ph.D. degree in 1981, both from Loughborough University of Technology, U.K.

He is currently an Associate Professor and Reader in the School of Electrical and Electronic Engineering at the University of Adelaide, Adelaide, Australia. His current research interests include image classification and target detection, array processor architectures, guidance control and multichannel digital receiver systems.

Alireza Moini (S'93–M'98) received the B.Sc. degree from the Department of Electrical and Electronics Engineering, Sharif University of Technology, Iran, in 1989, and the M.Eng.Sc. and Ph.D. degrees from the University of Adelaide, Adelaide, Australia, in 1994 and 1997, respectively.

From 1998 to 2001 he was a Lecturer in the Department of Electrical and Electronic Engineering, University of Adelaide. In 2001, he joined Intelligent Pixels Inc. and since 2002, he has been with Silverbrook Research Pty. Ltd. He is the author of *Vision Chips* (Norwell, MA: Kluwer, 1999). His research interests include analog VLSI for computer vision and neural network applications, CMOS imager design, mixed analog/digital signal processing, and high-speed GaAs VLSI digital circuits.