Segmenting Hands of Arbitrary Color

Xiaojin Zhu Jie Yang Alex Waibel
Interactive Systems Laboratories
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA
{zhuxj, yang+, ahw}@cs.cmu.edu

Abstract

Color has been widely used for hand segmentation. However, many approaches rely on predefined skin color models. It is very difficult to predefine a color model in a mobile application where the light condition may change dramatically over time. In this paper, we propose a novel statistical approach to hand segmentation based on Bayes decision theory. The proposed method requires no predefined skin color model. Instead it generates a hand color model and a background color model for a given image, and uses these models to classify each pixel in the image as either a hand pixel or a background pixel. Models are generated using a Gaussian mixture model with the restricted EM algorithm. Our method is capable of segmenting hands of arbitrary color in a complex scene. It performs well even when there is a significant overlap between hand and background colors, or when the user wears gloves. We show that the Bayes decision method is superior to a commonly used method by comparing their upper bound performance. Experimental results demonstrate the feasibility of the proposed method.

1. Introduction

Hand segmentation is a prerequisite for many gesture recognition tasks [1]. A popular feature for hand segmentation is skin color. Many approaches rely on predefined skin color models, which work well in constrained environments. In this paper, we investigate a hand segmentation problem for a wearable computer application. We show that it is very difficult to predefine a color model in this case because the light condition may change dramatically over time. To solve the problem, we propose a novel statistical approach to hand segmentation based on Bayes decision theory. The new method still relies on color information, but requires no predefined skin color model. The key innovation is the dynamic construction of hand and background color models using Gaussian mixture models and the restricted EM algorithm. The proposed method is capable of segmenting hands of arbitrary color in a complex scene. Another contribution of this paper is the study of upper bound performance for color-based hand segmentation. The study indicates the limit for the color-based approaches. We demonstrate that the proposed method is superior to a commonly used threshold method.

The rest of the paper is organized as follows. Section 2 describes a motivating application where hand segmentation approaches based on predefined color models are challenged. Section 3 presents the new method and algorithms. Section 4 investigates the performance of the new method, and compares it to the threshold method. Section 5 addresses the limitations of the proposed method and possible improvements.

2. A Motivating Application

2.1. Finger Menu for Wearable Computers

Recent technological advances have made wearable computers available for many different applications. However, how to efficiently interact with a wearable computer is still an open question. A gesture interface is certainly a good solution to the problem. Finger can been used as pointers for menu selection [2]. In a wearable environment a head-mounted see-through display may jitter due to involuntarily head motion. In this case it may be hard to point a finger at a menu item, because the item is displayed at a fixed position on the head-mounted display and moves around with the head.

We propose a new menu selection paradigm for wearable computers, namely the "finger menu". It works on a wearable computer with a head-mounted see-through display and a head-mounted video camera. Unlike traditional

GUI's that display menu items at fixed positions on the screen, our method associates menu items onto the user's five fingers. It works as follows:

1. The user sees through the head-mounted screen. The head-mount camera also monitors the same scene. Since the user's hand is not in the scene, no menu is displayed. (Figure 1a).

2. The menu system is activated when the user moves his hand, widely opened, into the scene. The system detects the hand with the camera, and displays five menu items at appropriate positions so that they appear on the fingertips through the head-mount display. (Figure 1b).

3. The menu items 'float'. When the hand moves, they move accordingly so that they stay on the fingertips. Thus there is no need for the user to move the hand to a specific place to make a menu selection. (Figure 1c)

4. By bending a finger as if it is a virtual 'click', the user can select the menu item on that fingertip (Figure 1d).

5. The menu system is de-activated when the user moves his hand out of the view.

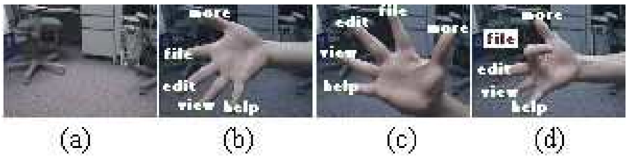


Figure 1. Finger menu demo

The advantages of this paradigm include: intuitive interaction, a user can operate the device with little or no training; efficient operation, there is no need to move a finger to a specific place to make a selection, which could be a hard head-hand coordination task; and no need for special pointing hardware.

In order to implement the finger menu system, we need to recognize hands from images taken by the head-mounted camera. Hand segmentation is an essential preprocessing step. However it is a hard problem for the following reasons:

- There are no restrictions on the background.
- The camera moves with the user's head.
- The light conditions may change dramatically. This includes changing shadow and varying light colors, e.g. under a sodium-vapor lamp.

There are numerous publications on hand segmentation. Two common methods are background subtraction and skin color segmentation. Obviously background subtraction is infeasible since there is no constant background. Color segmentation [3] [4] [5] [6] [7] [8] [9] [10] is more suitable in our case. Nevertheless, previous methods often use one static skin color model, which is inadequate for us. In the rest of this paper, we present a new way of segmenting hands with color information.

2.2. Problem Formulation

We formulate the hand segmentation problem as follows: The hand is known to be in an image. The hand color is unknown in advance (different environments may result in different hand colors), but is assumed to be largely consistent within the image. In addition, we are concerned with initial hand segmentation, not subsequent hand tracking. Thus we limit ourselves to a single image. Under these conditions, we want to segment the hand from the background, i.e. for each pixel in the image, we want to classify it as either a hand pixel or a background pixel.

2.3. The Data Set

To facilitate the discussion, we introduce our hand image data set. A user recorded image sequences of his hand with a head-mounted camera while performing the various finger menu actions. The actions were performed at typical places where a wearable computer is used including: office, home, vehicles, parks, and streets etc., with various backgrounds and light conditions. Some sequences were taken while the user was wearing gloves, which is a reasonable situation for working in the field. From the sequences, 326 images were selected randomly and the hands in these images were manually segmented. Each image is $80 * 60$, 24 bit color. The 326 images were randomly divided into two halves, with 163 images as training data and the other 163 images as test data. Figure 2 shows random samples of the training data set.

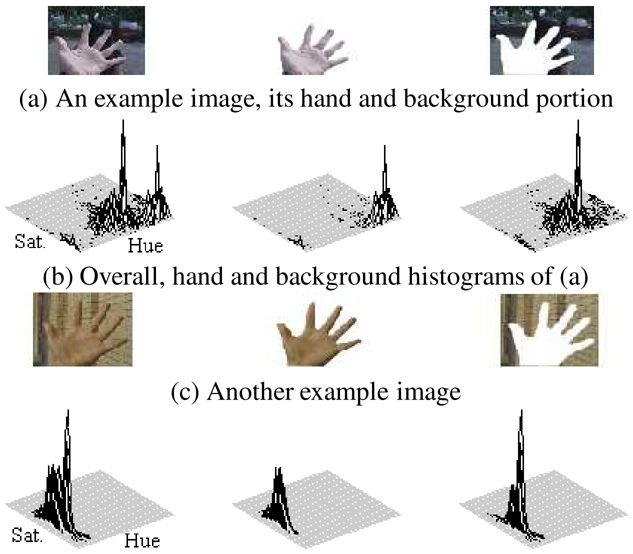
Figure 2. Some images in the training data set

In this work we use the HSV color space [11] instead of the RGB color space. Moreover, we use only Hue and

Saturation and ignore V (brightness) in order to minimize the influence of shadow and uneven lighting. We plot Hue-Saturation histograms for analysis. Since the images are manually segmented, we are able to plot the color histograms for the overall image, the hand portion, and the background portion respectively, as shown in Figure 3. Two observations arise after investigating some images:

- The peak of hand color is not fixed at a certain position on the H-S plane (compare Figure 3b with 3d). This means we cannot build a static hand color model for all images. Instead, we will build a different hand color model for different image.

- The hand color may partially overlap with the background color, as shown in Figure 3d. This means some hand pixels and background pixels have the same color. Thus misclassification is inevitable in color segmentation. However we want to minimize the error.



(a) An example image, its hand and background portion

(b) Overall, hand and background histograms of (a)

(c) Another example image

(d) Histograms of (c)

Figure 3. Example color histograms

3. A Novel Color Segmentation Method

In this section, we first introduce a Bayes decision theory framework for segmentation. The framework needs a hand color model and a background color model to work. We then present an approach to build the models dynamically for any given image using Gaussian mixture models and the restricted EM algorithm, which is the key innovation of this paper.

3.1. Hand Segmentation with Bayes Decision Theory

Given the color c and coordinates x, y of a pixel, we want to classify it as a hand pixel if

$$P(hand|c,x,y) > P(background|c,x,y) \qquad (1)$$

Applying the conditional version of Bayes rule, we get

$$P(hand|c,x,y) = \frac{P(c|hand,x,y) * P(hand|x,y)}{P(c|x,y)}$$

We assume c is conditionally independent of x, y given hand, i.e. $P(c|hand,x,y) = P(c|hand)$. Thus

$$P(hand|c,x,y) = \frac{P(c|hand) * P(hand|x,y)}{P(c|x,y)}$$

And similarly for $P(background|c,x,y)$. Note $P(hand|x,y) + P(background|x,y) = 1$, therefore (1) becomes

$$\begin{aligned} P(c|hand) * P(hand|x,y) &> \\ P(c|background) * (1 - P(hand|x,y)) & \qquad (2) \end{aligned}$$

(2) is our Bayes decision criterion, which will be used to classify the pixels in an image.

We need three models to compute (2). The first one, $P(c|hand)$, is the hand color model of an image. The second one, $P(c|background)$, is the background color model of the image. These two models need to be built for each image dynamically, as discussed in Section 3.2. The third one, $P(hand|x,y)$, describes the spatial distribution of hand pixels, i.e. how likely the pixel (x,y) is a hand pixel. We can estimate it from the training data set as follows:

$$P(hand|x,y) = \frac{\sum_{i \in training\ images} \delta_i(x,y)}{\sum_{i \in training\ images} 1} \qquad (3)$$

where $\delta_i(x,y) = 1$ if (x,y) is a hand pixel in image i, 0 otherwise. Figure 4 is the $P(hand|x,y)$ distribution of our training data set. The highest region corresponds to the palm. Since the user tends to place the hand at the center of the view, this distribution is reasonable.

3.2. Obtaining Color Models from Gaussian Mixture Models

We need to estimate the hand color model $P(c|hand)$ and the background color model $P(c|background)$ for any given image. Since hand color may change from image to image, and the hand color may partly overlap with the background color (as in Figure 3), this is not a trivial task.

One observation is that hand color is largely consistent within an image. This means hand pixels tend to concentrate

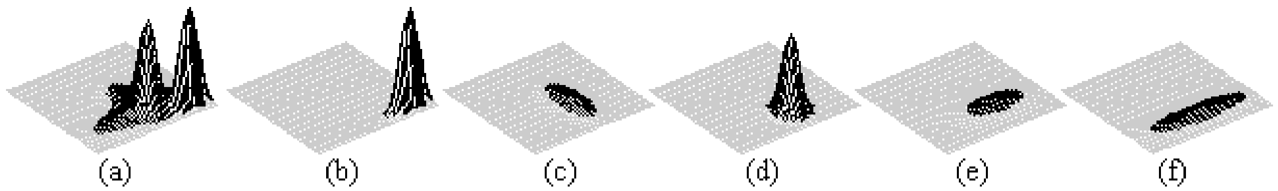


Figure 5. (a) GMM of Figure 3a by standard EM. (b-f) The weighted component Gaussians $w_i * N_i$

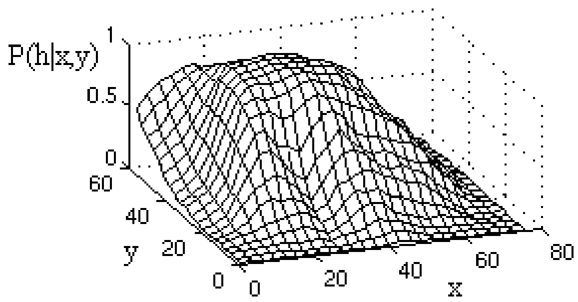


Figure 4. $P(hand|x, y)$ **of the training data**

and form a single peak in a hand color histogram. Intuitively it suggests to model the hand color $P(c|hand)$ with a Gaussian distribution. This leads to the following method.

Given an image, we can easily compute its overall color distribution $P(c)$ by normalizing the color histogram of the whole image:

$$P(c) = \frac{Count(pixels\ with\ color\ c)}{Count(all\ pixels)}$$

It has the following relationship with the (yet unknown) hand color model and background color model:

$$\begin{aligned} P(c) = \quad & P(hand) * P(c|hand) + \\ & (1 - P(hand)) * P(c|background) \end{aligned} \tag{4}$$

where $P(hand)$ is the percentage of hand pixels in the image, or the relative hand size (not to be confused with $P(hand|x, y)$ in the previous section, which is a pixel level value).

We can approximate $P(c)$ with a Gaussian Mixture Model (GMM) $P'(c)$ [12]. The GMM is a weighted sum of K Gaussian distributions $N_1, \ldots, N_K$:

$$P'(c) = \sum_{i=1}^{K} w_i * N_i(c),\ where\ \sum_{i=1}^{K} w_i = 1$$

K is determined empirically. We found $K = 5$ is sufficient. Let μ_i, Σ_i denote N_i's mean and covariance matrix respectively. With the Expectation Maximization (EM) algorithm [13] [14], we can train the GMM, that is to find a set of parameters w_i's, μ_i's and Σ_i's, such that $P(c) \approx P'(c)$. Figure 5 is the GMM trained with random starting parameters for the image in Figure 3a. Note how well the GMM approximates the actual overall color distribution in Figure 3b. Also note how well $w_1 * N_1$ (Figure 5b) resembles the actual hand color distribution in Figure 3b. Since we assume the hand color can be modeled with a Gaussian distribution, it is natural to think N_1 can be used as the hand color model. An immediate question follows: Can we guarantee that N_1 approximates the actual hand color distribution well enough for any image? If the answer is yes, then by the definition of GMM:

$$P'(c) = w_1 * N_1(c) + \sum_{i=2}^{K} w_i * N_i(c) \tag{5}$$

And comparing (4) with (5), we would have the following parametric forms to solve the problem:

$$\begin{aligned} P(hand) &= w_1 \\ P(c|hand) &= N_1(c) \qquad (6) \\ P(c|background) &= \frac{1}{1 - w_1} \sum_{i=2}^{K} w_i * N_i(c) \qquad (7) \end{aligned}$$

Unfortunately, in most cases the answer is no if we use the standard EM algorithm. In fact, the standard EM algorithm only guarantees the overall fitting of the whole distribution, but has virtually no control over the individual component Gaussians. There is no guarantee that any of the K Gaussian components will be a reasonable approximation to the actual hand color distribution. Better starting parameter heuristics will not help either. Figures 6(a-d) show an example image and its histograms. Figures 6(f-j) are the Gaussian components obtained using the standard EM algorithm. Obviously none of the Gaussian components resembles the actual hand color distribution (Figure 6c).

However, we will show that with certain modifications to the standard EM algorithm, we can enforce N_1, the first Gaussian component of a GMM, to be a good approximation of the hand color distribution such that we can use (6)(7). This is the key innovation of our method.

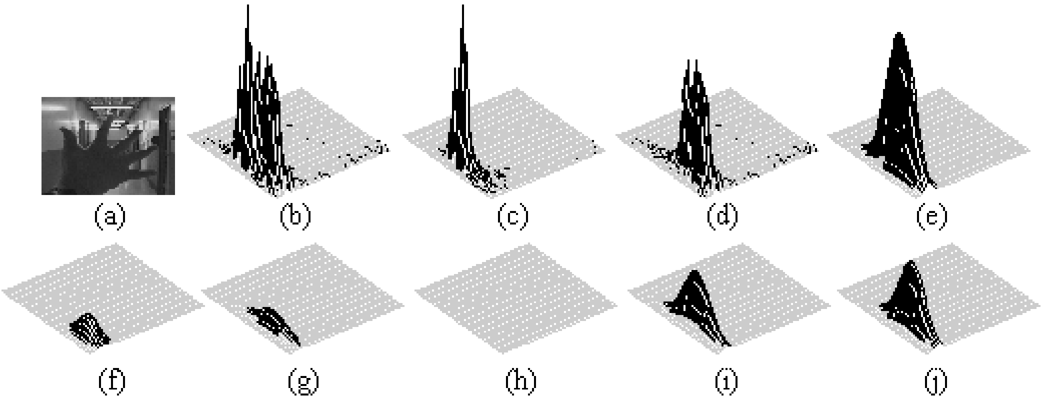


Figure 6. (a) An image showing the deficiency of standard EM. (b,c,d) The overall, hand and background color histogram. (e) The GMM by standard EM. (f-j) The weighted component Gaussians.

3.3. The Restricted EM Algorithm

During the Maximization step of the standard EM algorithm, the mean, covariance and weight of each Gaussian component can be adjusted freely. However we can fix some parameters to certain values, or limit their ranges during EM training. We call it the restricted EM algorithm. It will still converge to a local maximum in terms of likelihood [15]. More specifically, in this paper we will fix $\mu_1 = \tilde{\mu}_1$ and limit w_1 to be within range $[w_{low}, w_{high}]$. The meaning and value of $\tilde{\mu}_1$, w_{low}, and w_{high} will be discussed in next section. By restricting these two parameters, we can enforce N_1 to approximate the hand color distribution. The restricted EM algorithm is:

Initialization:
  Let $\mu_1 = \tilde{\mu}_1$, $w_1 = (w_{low} + w_{high})/2$.
  Set other parameters randomly.
During the E-step:
  Collect counts as in standard EM algorithm.
During the M-step:
1. Adjust only $\mu_2, \ldots, \mu_K$ as in standard EM, leave μ_1 unchanged.
2. Adjust $\Sigma_1, \ldots, \Sigma_K$ as in standard EM
3. Adjust $w_1, \ldots, w_K$ as in standard EM
4. If $(w_1 > w_{high})$ Then
   $w_i = w_i * (1 + \frac{w_1 - w_{high}}{1 - w_1}), i = 2, \ldots, K$
   $w_1 = w_{high}$
5. Similarly for the $w_1 < w_{low}$ case.

Iterating the E-step and M-step until converge.

Figure 7 shows the effect of the restricted EM algorithm. Figure 7a is the GMM obtained with the restricted EM algorithm on the same image of Figure 6a. Note the first Gaussian component (Figure 7b) now approximates the actual hand color histogram (Figure 6c).

3.4. Estimating the Restrictions

$\tilde{\mu}_1$ is the estimated mean of the hand color distribution. It needs to be estimated for each image as follows. Consider three random variables C_{xy}, C_{hand} and $C_{background}$ which take value of possible colors. C_{hand} has distribution $P(c|hand)$, and $C_{background}$ has distribution $P(c|background)$. C_{xy} is the color of pixel (x, y). We assume the following generative random process:

$$\begin{aligned} C_{xy} &= P(hand|x,y) * C_{hand} + \\ &\quad (1 - P(hand|x,y)) * C_{background} \end{aligned}$$

That is, pixel (x, y)'s color is generated in such a way: Firstly the 'identity' of this pixel is chosen to be 'hand' with probability $P(hand|x, y)$ and 'background' with probability $1 - P(hand|x, y)$ (see Figure 4 for the $P(hand|x, y)$ distribution). Secondly, if it is 'hand' a color is randomly picked for the pixel according to $P(c|hand)$, otherwise the color is picked according to $P(c|background)$.

Now consider a set of pixels in an image with hand probability $P(hand|x, y) = p$, where p is a fixed value between 0 and 1. Denote this set as $S_p = \{(x, y)|P(hand|x, y) = p\}$. The expectation of the previous equation over this set is

$$\begin{aligned} (x,y) \in S_p : \; E[C_{xy}] &= p * E[C_{hand}] + \\ &\quad (1-p) * E[C_{background}] \end{aligned}$$

By definition $E[C_{hand}] = \tilde{\mu}_1$. And for $(x, y) \in S_p$:

$$\begin{aligned} E[C_{xy}] &= \frac{\sum_{(x,y)\in S_p} color\ of\ (x,y)}{||S_p||} \\ &= Average\ color\ of\ pixels \in S_p \end{aligned}$$

Therefore we get

$$\tilde{\mu}_1 = \frac{1}{p}(Average\ color\ of\ pixels \in S_p) -$$

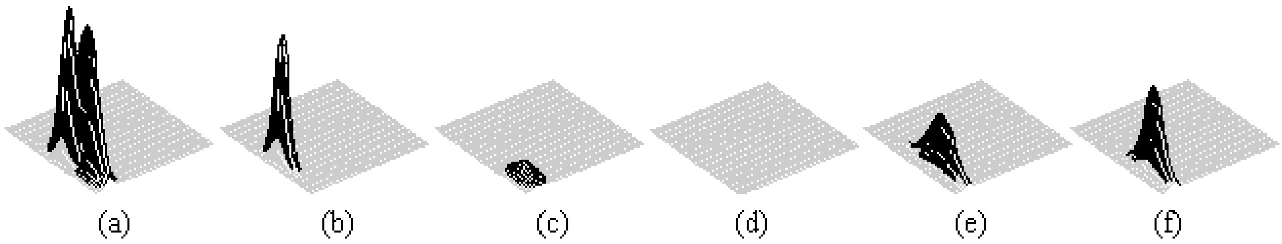


Figure 7. (a) GMM of Figure 6a by restricted EM. (b-f) The weighted component Gaussians $w_i * N_i$

$$\frac{1-p}{p} E[C_{background}] \tag{8}$$

where $E[C_{background}]$ can be estimated from the training data. In particular, if we can find some p close to 1 such that $||S_p||$ is large enough, we can use the approximation

$$\tilde{\mu}_1 \approx (Average\ color\ of\ pixels \in S_p)$$

The restrictions w_{low}, w_{high} control w_1, which can be interpreted as $P(hand)$, the relative hand size. They are estimated from the training data. The distribution of $P(hand)$ in the training data is plotted in Figure 8. We compute the

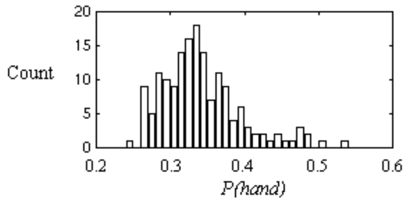


Figure 8. $P(hand)$ **in training data**

mean μ and standard deviation σ of this distribution. Since we expect the hand size in a new image to be comparable to those in the training data, we let

$$w_{low} = \mu - 2\sigma,\ w_{high} = \mu + 2\sigma \tag{9}$$

3.5. The Complete Algorithm

The complete algorithm is as follows.

During training:
  Build $P(hand|x, y)$ with (3)
  Estimate the weight restrictions w_{low}, w_{high} with (9)
During decoding:
  Estimate the mean restriction $\tilde{\mu}_1$ with (8)
  Run the restricted EM algorithm in Section 3.3
  Generate the hand color model with (6)
  Generate the background color model with (7)
  Classify each pixel with (2)

4. Performance Analysis

4.1. The Performance on Test Data

We tested the proposed method on the 163 test images. We achieved an average false positive (background misclassified as hand) rate of 4.0%, false negative rate of 7.5%, and total error rate of 11.5%. The decoding takes about 0.1 second for each image on a PC.

Figure 9 shows some segmentation results. Since our method is a pure pixel-wise statistical classifier, no filtering is applied. Therefore some segmented hands have holes or background dots. Figure 9g, 9h show the user wearing gloves. Since the gloves have consistent color, they are easily recognized. Figure 9k is an example where the hand Gaussian mistakenly 'grabs' a nearby background color peak during restricted EM training. Figure 9l shows another case where our method fails. This is because the hand color in this image is not consistent: the image was taken inside a car, the thumb and part of the palm was rendered bluish under the windshield. Therefore a single Gaussian can no longer model the hand color, which leads to the failure. Nonetheless, given the difficulty of the test set, we consider our method to be promising.

4.2. The Upper Bound Performance of Our Method

We are interested in the upper bound performance of the proposed method. That is the performance when we have 'perfect' hand and background color models. Since the test data are also manually segmented, we are able to build a 'perfect' hand color model for each test image by normalizing its hand color histogram:

$$P^*(c|hand) = \frac{Count(hand\ pixels\ with\ color\ c)}{Count(hand\ pixels)}$$

and similarly a 'perfect' background color model. Then we use these 'perfect' models to classify pixels in the same image. The performance is considered the upper bound of our method, because the models obtained by (6) and (7) are approximations to these 'perfect' models.

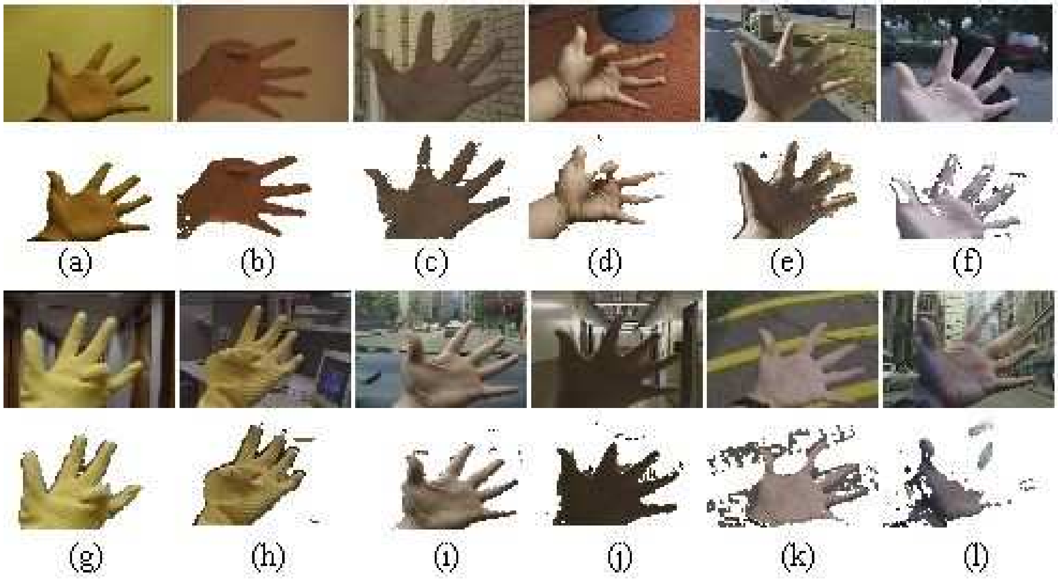


Figure 9. Segmentation results

4.3. The Upper Bound Performance of A Simple Threshold Method

We compare the Bayesian method with a simple skin color segmentation method that is frequently (but often implicitly) used. The simple method builds a hand color model $P(c|hand)$ and classifies a pixel as a hand pixel if

$$P(color\ of\ the\ pixel|hand) > T$$

for some threshold T, otherwise it classifies it as a background pixel.

We are interested in the upper bound performance of this method. As in Section 4.2 we build a 'perfect' hand color model for each image, and classify the pixels in the same image with the simple method. We repeat the experiment with different thresholds T. Figure 10a shows the performance curve with different T. The lowest error rate is 18.3% with T=0.015. Figures 10b and 10c show how false negative and false positive rates change with respect to T in the method. (the color distributions are mapped to one dimension). Since the two distributions are intrinsically overlapping, a small T will generate less false negative but more false positive, and vise versa.

Table 1 summarizes the performances of different methods. Obviously the Bayes decision method is better than the simple threshold method.

	False Neg.	False Pos.	Error
Bayes Method (Actual)	7.5%	4.0%	11.6%
Bayes Method (Upper Bound)	3.1%	3.6%	6.7%
Simple Method (Upper Bound)	6.4%	11.9%	18.3%

Table 1. Performance comparison

5. Conclusions

We proposed a new way of color segmentation for hand recognition in a wearable environment. The method builds statistical hand and background color models for each image using GMM and the restricted EM algorithm, and classifies pixels with Bayes decision criterion. The performance of the proposed method is promising.

The success of this method relies on the assumption that hand color in a given image is consistent, and hence can be modeled by a Gaussian distribution. Another important prerequisite is that there need to be a few positions where hand tends to occur with high probability, so that the average hand color in a given image can be estimated reliably. The wearable computer application mentioned in Section 2 satisfies these requirements.

Many things can be done for further improvement. For example, some conventional image processing methods such as filtering and region growing will definitely help. Moreover, currently each pixel is processed individually, whereas it might be beneficial to consider interactions between pixels. In addition, we only considered color information. As the upper bound performance reveals, there is a limit on how

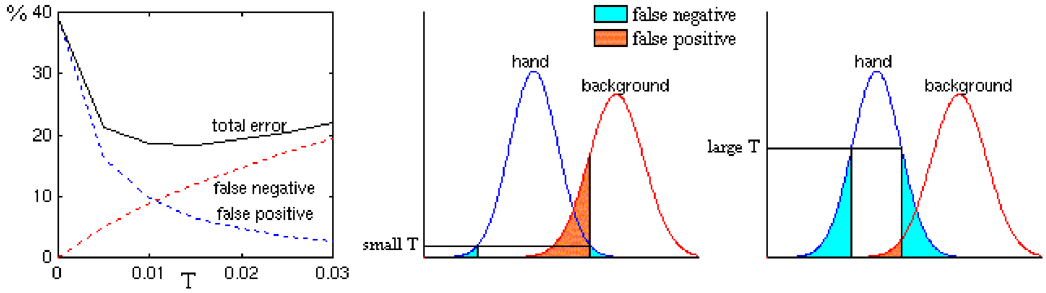


Figure 10. The average upper bound performance of a simple method and its analysis

well we can do with color. Adding different information, such as shape, would be helpful. We are investigating some of them, and are applying the proposed method to create a gesture based wearable computer interface.

6. Acknowledgement

The authors would like to thank Larry Wasserman, Roni Rosenfeld, Hua Yu, Pan Yue, Ke Yang, Iain Matthews, William Kunz and all members in the Interactive Systems Labs for their inspiring suggestions and help. This research is partially supported by the Defense Advanced Research Projects Agency under contract number DAAD17-99-C-0061. Xiaojin Zhu's research is supported in part by the National Science Fundation under grant SBR-9720374.

References

[1] V. Pavlovic, R. Sharma, T. S. Huang. *Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review.* IEEE PAMI, Vol 19, No.7, pp. 677-695, 1997

[2] T. Starner, S. Mann, B. Rhodes, J. Levine, J. Healey, D. Kirsch, R. Picard, A. Pentland. *Augmented Reality Through Wearable Computing.* Presence Vol. 6, No. 4, 1997.

[3] S. Ahmad. *A Usable Real-Time 3D Hand Tracker.* Proceedings of the Twenty-Eighth Asilomar Conference on Signals, vol.2, pp. 1257-1261, 1995.

[4] K. Imagawa, S. Lu, S. Igi. *Color-Based Hands Tracking System for Sign Language Recognition.* Proc. 3rd Int'l Conf. On Automatic Face and Gesture Recognition, pp 462-467, 1998

[5] R. Kjeldsen, J. Kender. *Finding skin in color images.* Proc. 2nd Int'l Conf. On Automatic Face and Gesture Recognition, pp. 312-317, 1996.

[6] Y. Raja, S. J. McKenna, S. G. Gong. *Tracking and Segmenting People in Varying Lighting Conditions using Colour.* Proc. 3rd Int'l Conf. On Automatic Face and Gesture Recognition, pp 228-233, 1998

[7] D. Saxe, R. Foulds. *Towards Robust Skin Identification in Video Images.* Proc. 2nd Int'l Conf. On Automatic Face and Gesture Recogntion, pp. 379-384, 1996.

[8] T. Starner, J. Weaver, A. Pentland. *Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video.* IEEE PAMI, vol.20, no.12, pp. 1371-1375, 1998.

[9] J. C. Terrillon, M. David, S. Akamatsu. *Automatic Detection of Human Faces in Natural Scene Images by Use of a Skin Color Model and of Invariant Moments.* Proc. 3rd Int'l Conf. On Automatic Face and Gesture Recognition, pp.112-117, 1998

[10] M. H. Yang, N. Ahuja. *Extraction and Classification of Visual Motion Patterns for Hand Gesture Recognition.* Proc. IEEE CVPR, pp. 892-897, 1998

[11] J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes. *Computer Graphics: Principles and Practice.* 2nd ed. p. 590. Addison-Wesley, Mass., 1993

[12] R. O. Duda, P. E. Hart. *Pattern classification and scene analysis.* Wiley, NY, 1973

[13] A. P. Dempster, N. M. Laird, D. B. Rubin. *Maximum likelihood from incomplete data via the EM algorithm.* Journal of the Royal Statistical Society, 39 No.B:1-38, 1977

[14] T. Yamazaki. *Introduction of EM algorithm into color image segmentation.* Proc. ICIPS'98, pp. 368-371, Aug. 1998

[15] L. Wasserman. *personal communications.* 1999