ELSEVIER

Image and Vision Computing 14 (1996) 685–702

image AND vision COMPUTING

Application note

Color texture segmentation for clothing in a computer-aided fashion design system

Chin-Chou Chang, Ling-Ling Wang*

Department of Computer Science, National Tsing Hua University, Hsinchu, 30043 Taiwan, ROC

Received 4 July 1995; revised 11 December 1995

Abstract

A traditional fashion designer has to draw a large number of drafts in order to accomplish an ideal style. Better performance can be achieved if these operations are done on computers, because the designer can easily make changes for various patterns and colors. To develop a computer-aided fashion design system, one of the most difficult tasks is to automatically separate the clothing from the background so that a new item can be 'put on'. One difficulty of the segmentation work arises from the diverse patterns on the clothing, especially with folds or shadows. In this study, circular histograms are first utilized to quantize color and to reduce shadow/highlight effects. Then a color co-occurrence matrix and a color occurrence vector are proposed to characterize the color spatial dependence and color occurrence frequency of the clothing's texture. Next, based on the two color features blocks on the clothing are found by a region growing method. Finally, post-processing is applied to obtain a smooth clothing boundary. Experimental results are presented to show the feasibility of the proposed approach.

Keywords: Image segmentation; Color quantization; Color co-occurrence matrix; Computer-aided fashion design

1. Introduction

1.1. Research motivation

A traditional fashion designer has to draw a large number of drafts in order to accomplish an ideal style. Hence, an effective tool which can provide a realistic customer model, and by which a large amount of drawing time can be saved, is very helpful for a fashion designer.

Due to the rapid development of computer software, a computer-aided fashion design system now becomes feasible. By inputting a customer's photographs to a computer, a designer can take the customer as a model. Further, through the computer program's cut & paste commands (functions), when the colors or pattern prints of the the designed clothing are to be modified, it is not necessary to redraw a new draft. Only the part to be modified is cut and pasted on the computer. Therefore, much drawing work on paper can be saved, and the best selection of colors or patterns on the clothing designed can be determined quickly according to the customer model. Better performance can thus be achieved.

For a computer-aided fashion design system, one of the most difficult tasks is to automatically cut or separate the clothing to be modified from its background so that the clothing can be 'taken off', and a new item with different colors, patterns or even new styles can be 'put on' to the model. Few commercial image processing or image editing packages perform segmentation well. For some packages the users even have to specify by hand the boundary of the clothing on the image. Some other systems might support an automatic segmentation function, but the segmentation results are not yet satisfactory. Segmentation is difficult because there are usually folds, shadows or complex patterns on the clothing, as Fig. 1 shows. In this study, designing an algorithm that can automatically separate clothing from its background is our aim.

1.2. Survey of image segmentation

Segmentation is a process of partitioning an image into several meaningful regions that are homogenous with respect to some characteristics. Various methods

* Corresponding author.

PII S0262-8856(96)01083-9

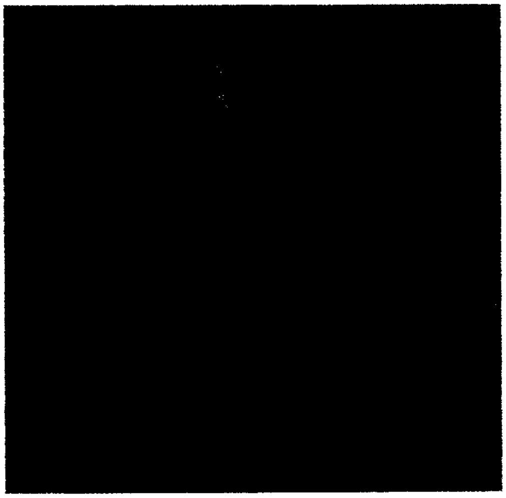


Fig. 1. An overcoat on which folds and shadows appear.

have been presented for color image segmentation or classification. The earlier ones used three-dimensional histogram clustering techniques to segment color images in a single phase. In these methods, the histograms were first computed from a set of color features, and segmentation was achieved based on the clustering result of the histograms. For example, Ohta et al. [1] proposed three color features, (R+G+B)/3, R-B and (2G-R-B)/2, and used the Karhunen–Loeve (KL) transformation to analyze histograms of the features. Tominaga [2,3] also used the KL transformation to analyze histograms, but different color spaces were utilized: the Munsell space [2] and the (L^*, a^*, b^*) space [3]. Sarabi and Aggarwal [4] proposed a cluster tree for color clustering in the (X, Y, I) space. Celenk [5] presented a clustering algorithm by dividing the (L^*, H°, C^*) color space into some circulary-cylindrical decision elements. Since a significant amount of computational effort is required in the above methods, some researchers [8–10] proposed two-phase coarse-to-fine approaches to remedy this drawback. In these methods, the scale-space filter [6,7] was used in the coarse phase to convolute histograms with the Gaussian function, and the color image was segmented coarsely using the valleys of the histograms. In the fine phase, the Markov random field [8] or fuzzy c-means algorithm [9] was used to refine the segmentation results.

A primary weakness of the above approaches is that they cannot overcome the problem of shadows and highlights. To cope with this kind of problem, Andreadis et al. [11] proposed three chromaticity features, (R/(R+G+B), G/(R+G+B) and B/(R+G+B)) for segmentation, which are invariant when the illumination changed. It was not suitable for practical situations since it was wrongly assumed that the same amount of values for R, G and B channels decreased when shadows emerged. Klinker et al. [12] used the dichromatic reflection model [13–15] to generate physical descriptions of the reflection process occurring in a scene. According to the reflection process, highlights and shadows could be located and removed from an image. However, a priori information about the camera should be known. It prohibited flexible image segmentation. All the above methods processed only color information, but no texture pattern was considered in the segmentation.

So far, only a few papers have discussed color texture segmentation and recognition. Scharcanski et al. [16,17] proposed characteristic colors as features to represent the color aspect of textures. Panjwani and Healey [18] applied the Markov random fields to model color textures. Caelli and Reye [19] proposed a unified scheme to extract features in a single spatial-chromatic space without separating color, texture and shape into different channels. These methods performed texture segmentation well when no rotation, shadow, highlight and fold were on the textures.

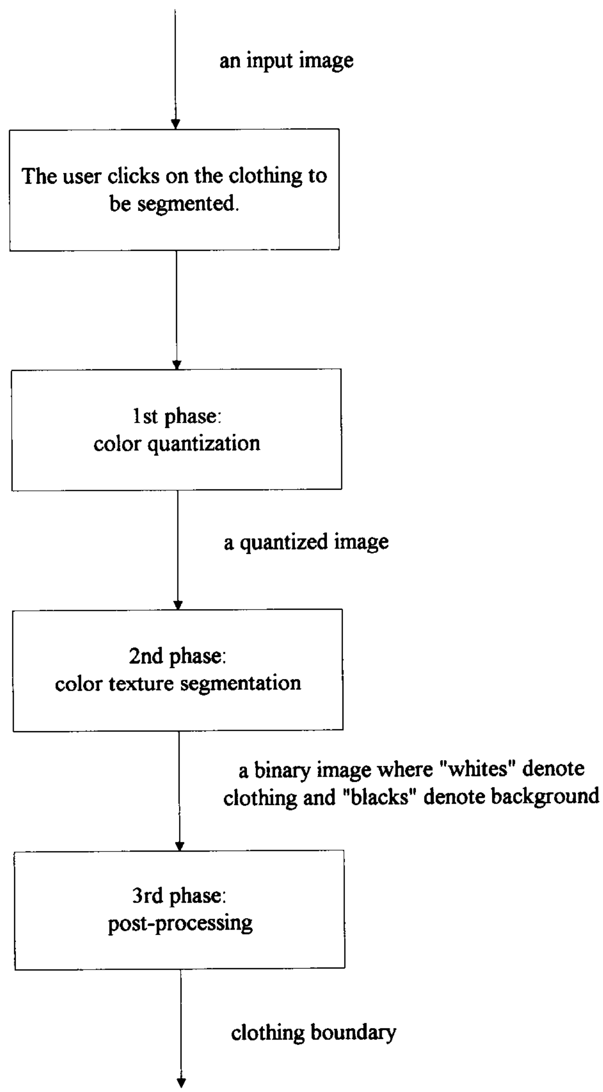


Fig. 2. Block diagram of the proposed approach.

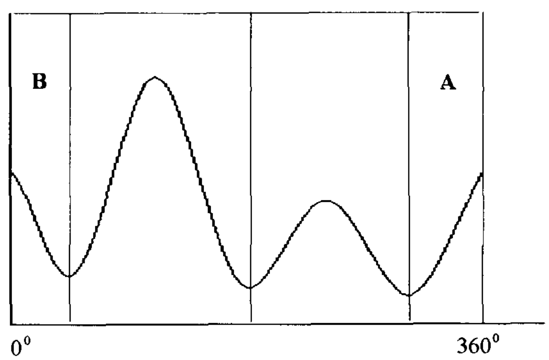


Fig. 3. Example of wrong classification caused by thresholding without considering the periodicity of the hue histogram.

1.3. Overview of proposed approaches

In this study, we aim to develop a color texture segmentation method which is somewhat tolerant to shadows, highlights and folds on the textures for a computer-aided fashion design system. Using the automatic segmentation method, a designer does not have to manually specify the boundary of the clothing to be modified. The designer only needs to give a 'seed' point on the clothing, and then it can be automatically separated from the background. We divide the segmentation process into three phases, as shown in Fig. 2. A color image with 24 bits per pixel might contain 2^{24} different colors in the image. This causes a problem of exhaustive computational burden, hence in the first phase of the proposed method, an input color image is quantized to fewer colors. As a result, the computational requirement in the measurement of blocks' similarity (in the second phase) can be saved. Furthermore, if an appropriate color model is selected, the effects of highlights and shadows on the image can be reduced after color quantization. In the second phase, a seed block which embodies enough information about the color and texture of the clothing is first located from the seed point. By a region growing method, blocks having the similar color co-occurrence matrices and color occurrence vectors (which will be discussed later) with the seed block can then be found. These blocks form the clothing part. Finally, post-processing is applied in the third phase to extract and smooth the clothing boundary. The proposed two color texture features, color co-occurrence matrix and color occurrence vector, are insensitive to fold and orientation variations on the textures. Experimental results indicate that the proposed method is promising for color texture segmentation.

Detailed descriptions of the proposed method are given in Section 2. Experimental results are presented in Section 3. Conclusions appear in the last section.

2. Proposed approach

In the proposed approach there are three major tasks: color quantization, color texture segmentation and post-processing. Described in Section 2.1 is the color quantization method. The features used for color texture segmentation are introduced in Section 2.2, and the detailed description of the color texture segmentation approach appears in Section 2.3. Finally,

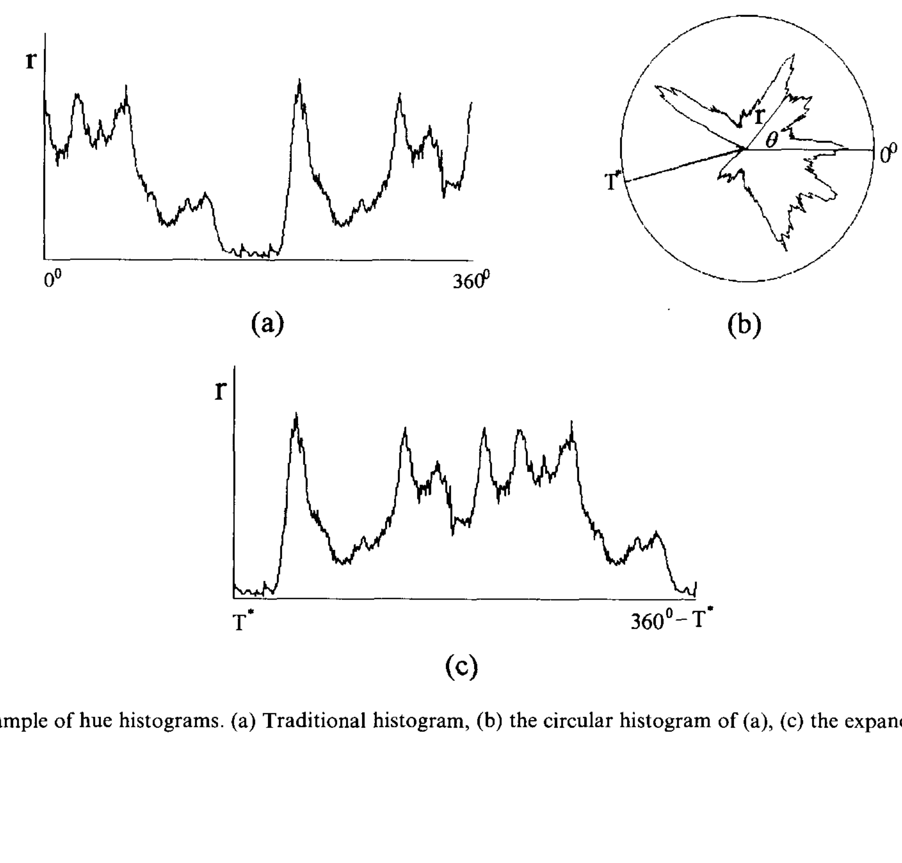


Fig. 4. Example of hue histograms. (a) Traditional histogram, (b) the circular histogram of (a), (c) the expanded result of (b).

post-processing for extracting and smoothing the clothing boundary is given in Section 2.4.

2.1. Color quantization

To save the computational requirement in segmentation, an input color image is quantized so that the number of colors contained in the image is reduced while the primary chromatic information about the image still remains. In the quantization method, we first determine the number of quantized colors, say k, by a histogram thresholding technique. Then, the k-means classification algorithm [9,29] is performed to classify each pixel in the image to one of the k colors. The k-means algorithm is effective in classification. However, if the number of clusters is unknown in advance, we may have to perform the algorithm for each k to select the optimal one. Hence, determining the number of clusters based on the histogram technique makes the classification efficient. A detailed description is given below.

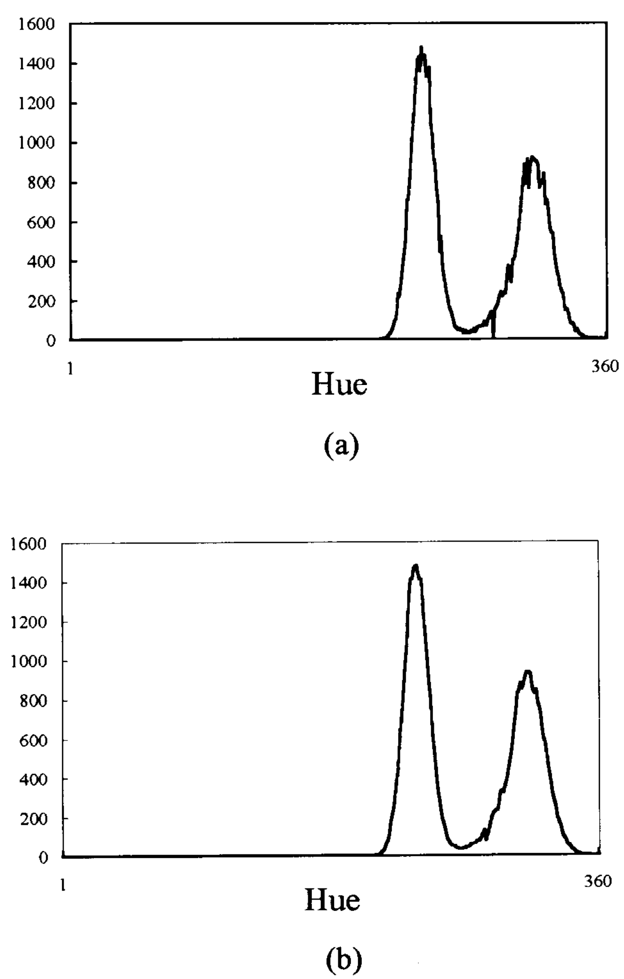


Fig. 5. Scale-space filtering. (a) Original histogram, (b) smoothed result.

2.1.1. Color model

For highlights and shadows on the image not to have a great influence over the quantized result, an appropriate color model should be selected. So far, various color models have been proposed for color specification [20,21], such as CIE (R, G, B), (X, Y, Z), $(L^*, u^* v^*)$, and so on. The spectral primary sources of the CIE (R, G, B) system do not yield a full gamut of reproducible colors. That is, the (R, G, B) color space has negative tristimulus values. This has led to the development of the CIE (X, Y, Z) system. The three tristimulus values in the (X, Y, Z) color space are positive everywhere. But it is not a uniform color space. In a uniform color space, the Euclidean distance between two colors is proportional to the color difference perceived by humans. In 1976, the CIE standardized the (L^*, u^*, v^*) uniform color space. Nevertheless, these color models are not intimately related to the way in which humans perceive color. Accordingly, the perceptual (L^*, C^*, H^*) color space was proposed, where L^* represents the lightness, C^* the chroma and H^* the hue of the color. The color transformation steps used in the study are summarized as follows:

$$\text{NTSC (R, G, B)} \rightarrow \text{CIE (X, Y, Z)} \rightarrow \text{CIE } (L^*, u^*, v^*) \rightarrow (L^*, C^*, H^*)$$

where NTSC(R, G, B) is the National Television System Committee (NTSC) receiver primary system (R, G, B), which is a standard for television receivers. The detailed transformation formulas are given below:

NTSC(R, G, B) → CIE(X, Y, Z)

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.813 & 0.011 \\ 0.000 & 0.010 & 0.990 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

CIE(X, Y, Z) → CIE(L^*, u^*, v^*):

$$u' = \frac{4X}{X + 15Y + 3Z}, \quad v' = \frac{9Y}{X + 15Y + 3Z}$$

$$u_w = 0.1978, \quad v_w = 0.4683, \quad y_w = 1$$

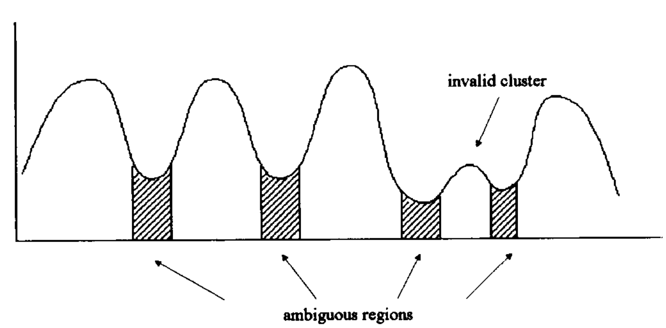


Fig. 6. Example of four ambiguous regions and an invalid cluster in a histogram.

$$L^* = \begin{cases} 116\left(\frac{Y}{y_w}\right)^{\frac{1}{3}} - 16 & \text{if } \frac{Y}{y_w} > 0.008856 \\ 903.3\frac{Y}{y_w} & \text{if } \frac{Y}{y_w} \leqslant 0.008856 \end{cases}$$

$$u^* = 13L^*(u' - u'_w), \quad v^* = 13L^*(v' - v'_w)$$

$\mathrm{CIE}(L^*, u^*, v^*) \rightarrow (L^*, C^*, H^*)$:

$$H^* = \tan^{-1}\left(\frac{v^*}{u^*}\right)$$

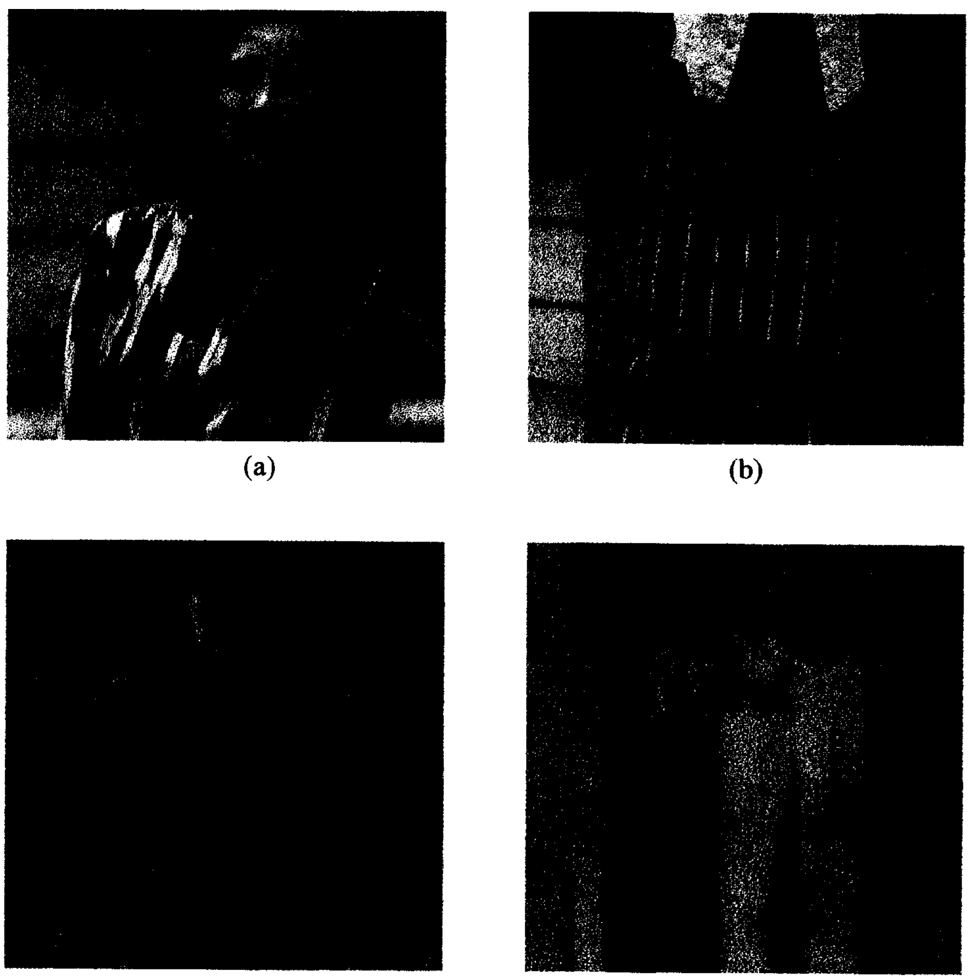
$$C^* = [(u^*)^2 + (v^*)^2]^{\frac{1}{2}}$$

$$L^* = L^*.$$

Color classification using the perceptual color space has two advantages [25]. First, specifying or analyzing colors using the perceptual color space has more visual intuition for humans than using the (R, G, B) color space. Second, the intensity component can be decoupled from the color information. This advantage is useful for quantizing a color image when shadows and highlights are present, since the influence of shadows and highlights on the intensity component is greater than that on the chromaticity components. Therefore, we analyze color images using only the chromaticity hue information to reduce the effects of shadows and highlights on the image.

2.1.2. Color quantization method

Hue is an attribute associated with the dominant wavelength in a mixture of light waves. It is a promising feature for color image segmentation [26]. Thus, only the hue component is used here to construct a histogram for quantization. Note that the hue value is meaningless for a pixel when its lightness value is very high or very low. If the lightness value of a pixel is larger than a bright threshold, it is quantized to white. On the other hand, if the lightness is smaller than a dark threshold, it is quantized to black. The hue values range from 0° to 360°. The difference between 0° and 359° is as small as 0° and 1°, therefore when there is an object whose hue



(a) (b) (c) (d)

Fig. 7. Four model images.

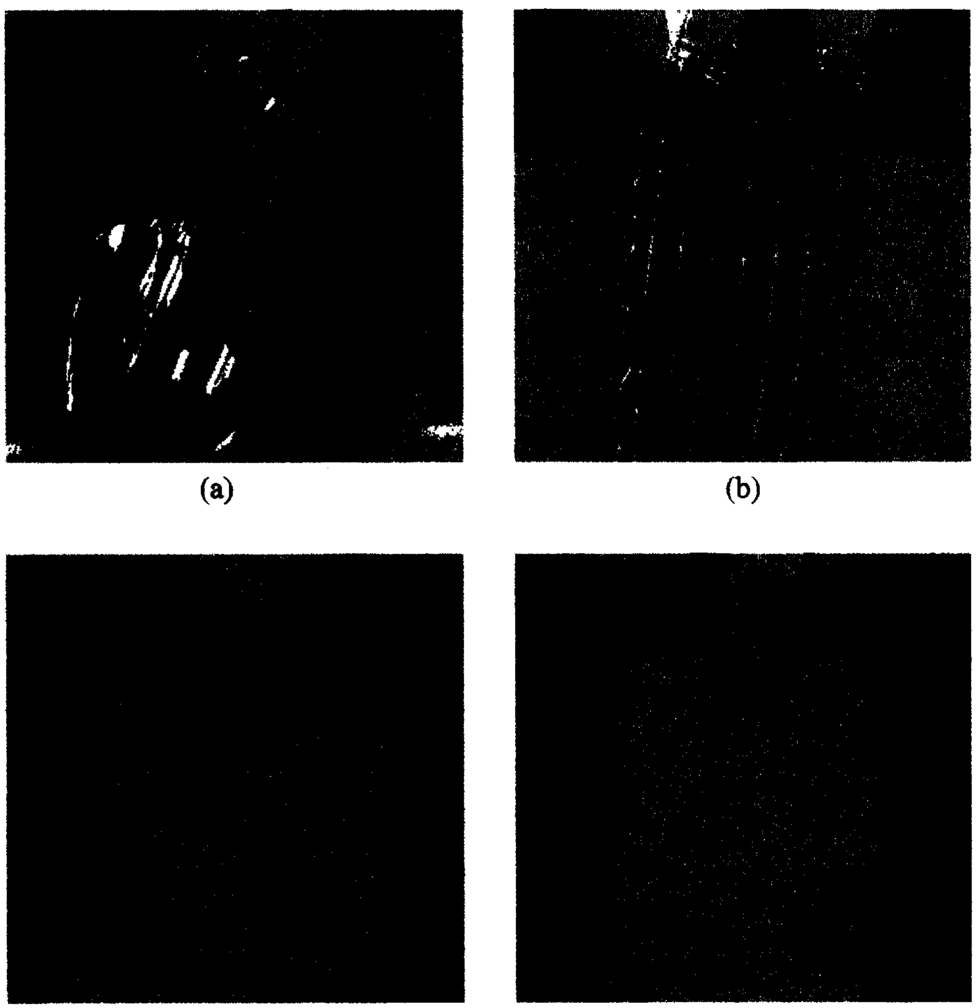
values concentrate near 0° or 360°, a wrong classification will occur if we use the traditional thresholding technique on the hue histogram without considering the periodicity of the hue values. See Fig. 3 for an illustration, where peaks A and B belong to the same cluster but are separated to two distinct color clusters after thresholding. Joining together 0° and 360° of the hue histogram to a circular one is a feasible solution to this problem [26]. In a circular histogram, angle θ indicates the hue value and radius r the number of pixels. To use the traditional thresholding technique on the circular hue histogram, it is converted to a traditional one. A method is to first find the angle T^* at which the following quantity is minimum:

$$\min_T \sum_{j=-5^\circ}^{5^\circ} f(T+j), \quad 0^\circ \leq T < 360^\circ$$

where f denotes the function of the circular histogram. Then, the circular histogram is cut at T^* and is expanded to a traditional histogram. See Fig. 4 for an example. Shown in Fig. 4(a) is a traditional histogram; its corresponding circular histogram is given in Fig. 4(b), and Fig. 4(c) shows the resulting expansion of the circular histogram. A histogram often contains many rugged peaks, and it results in a difficulty with thresholding. A scale-space filter [6,7] can be used to smooth the histogram:

$$F(x,\tau) = f(x) * g(x,\tau) = \int_{-\infty}^{\infty} f(u)\frac{1}{(2\pi)^{1/2}\tau} \times \exp\left[\frac{(x-u)^2}{2\tau^2}\right] du$$

where g is the Gaussian function and '$*$' denotes a 1D convolution. τ is the Gaussian deviation. The larger the value of τ, the smoother the histogram. Its value is fixed to be 2 (which is an empirical value) in the experiments.



(a) (b) (c) (d)

Fig. 8. Color quantization of images in Fig. 7. (a) Quantized to seven colors, (b) quantized to seven colors, (c) quantized to eight colors, (d) quantized to six colors.

An example is shown in Fig. 5, where Fig. 5(a) is the hue histogram of Fig. 1. After scale-space filtering, we can easily find valleys in a histogram by computing the first and second derivatives of the histogram and finding locations which satisfy:

$$\frac{\partial F}{\partial x} = 0, \quad \frac{\partial^2 F}{\partial x^2} > 0$$

Thresholding or classification can be achieved using these valleys. The number of clusters corresponds to the number of significant peaks in the histogram. However, adjacent clusters frequently overlap, so pixels in ambiguous regions, as shown in Fig. 6, are difficult to classify. In addition, a cluster whose number of pixels is significantly small is frequently considered invalid. Hence, classifying the pixels belonging to ambiguous regions or invalid clusters is limited if only the valleys of the histogram are used. The k-means algorithm is applied here to accomplish the classification task [9,29]. In the algorithm, the desired number of clusters, k, is the difference between the number of significant peaks in the hue histogram and the number of invalid clusters. After the classification process finishes, all pixels in a cluster are assigned the same mean color of the cluster. Therefore, the image is quantized to k hue values. The quantized colors will be denoted as $\{1, 2, \ldots, k\}$ hereafter. That is, each quantized color is labeled as a number. Shown in Fig. 7 are four model images; their quantized results are shown in Fig. 8.

2.2. Color texture features

The following factors constitute the problem of clothing segmentation. First, textures or patterns on the clothing to be segmented are very diverse, and texture orientations on different parts of the clothing are distinct. Besides, there are folds on the clothing, as shown in Fig. 1. Hence, flexible features which are adaptive to these variations are desired in clothing segmentation.

The co-occurrence of gray levels characterizes the gray level spatial dependence of a gray-scale image block. The gray level spatial dependence carries much of the texture information. Therefore, the gray level co-occurrence matrix is often used to define features for texture analysis and recognition [27,28]. The concept of a gray level co-occurrence matrix can be easily adapted for characterizing color images. For a color image block, the (i, j)-th entry of the color co-occurrence matrix with distance d records the occurrence frequency that a pixel with the ith color has a d-distance neighbor with the jth color. Assume that the image block is of size $L_x \times L_y$. Let G be the set of color indexes of the image block. Formally, for angles quantized to 45° intervals, four color co-occurrence matrices of the image block are defined as follows:

$$\begin{aligned}
P(i,j,d,0^\circ) = \#\{((k,l),(m,n)) \in (L_x \times L_y) \\
&\times (L_x \times L_y) | \\
&k - m = 0, |l - n| = d, \\
&I(k,l) = i, I(m,n) = j, \text{ where } i,j \in G\} \\
P(i,j,d,45^\circ) = \#\{((k,l),(m,n)) \in (L_x \times L_y) \\
&\times (L_x \times L_y) | \\
&(k - m = d, l - n = -d) \\
&\quad \text{or } (k - m = -d, l - n = d), \\
&I(k,l) = i, I(m,n) = j, \text{ where } i,j \in G\} \\
P(i,j,d,90^\circ) = \#\{((k,l),(m,n)) \in (L_x \times L_y) \\
&\times (L_x \times L_y) | \qquad (1) \\
&|k - m| = d, l - n = 0, \\
&I(k,l) = i, I(m,n) = j, \text{ where } i,j \in G\} \\
P(i,j,d,135^\circ) = \#\{((k,l),(m,n)) \in (L_x \times L_y) \\
&\times (L_x \times L_y) | \\
&(k - m = d, l - n = d) \\
&\quad \text{or } (k - m = -d, l - n = -d), \\
&g(k,l) = i, g(m,n) = j, \text{ where } i,j \in G\}
\end{aligned}$$

where # denotes the number of elements in a set, and g represents the color index function of the image block. An orientation-independent color co-occurrence matrix, denoted by $\mathbf{O}$, can be obtained using the summation of the four matrices as follows:

$$\mathbf{O}(i,j,d) = \frac{1}{L_x \times L_y} \times [P(i,j,d,0^\circ) + P(i,j,d,45^\circ) + P(i,j,d,90^\circ) + P(i,j,d,135^\circ)]$$

where each entry of $\mathbf{O}$ is normalized to be between 0 and 1. In the following discussion, when the color co-occurrence matrix is referred, it denotes matrix $\mathbf{O}$. For a 24-bit color image, the number of colors exceeds 16 million. Hence, the storage required for a color co-occurrence matrix is significantly large, and the calculation is computationally infeasible. A solution of this problem is to quantize the color such that the number of colors descends to a feasible value, for example, less than 10. This justifies the reason why color quantization is performed in the first phase in the approach proposed. Note that in the study, G is $\{1, 2, \ldots, k\}$, where k is the total number of quantized colors in an image.

The color co-occurrence matrix characterizes the spatial relationship of colors on textures, but the color occurrence frequency information is not explicit in the matrix. The information is important especially when we classify blocks near the clothing contour. We define it as color occurrence vector. Each entry of the vector, $\mathbf{V}(i)$, records the frequency that the i-th color occurs in an image. Its definition is given by

$$\mathbf{V}(i) = \frac{\text{occurrence times of the } i\text{th color}}{L_x \times L_y}, \quad i \in G.$$

See Fig. 9 for an example. Fig. 9(a) shows a 4×4, 4-color image block, where $G = \{1, 2, 3, 4\}$. Its color occurrence vector and color co-occurrence matrix with distance $d = 1$ are listed in Figs. 9(b) and (c), respectively.

In clothing segmentation, the color occurrence vector and color co-occurrence matrix are used as features for each image block. The distance measures between two color occurrence vectors, $\mathbf{V}_1$ and $\mathbf{V}_2$, and two color co-occurrence matrices, $\mathbf{O}_1$ and $\mathbf{O}_2$, are defined, respectively, as

$$D_{cv} = \sum_{i=1}^{k} |\mathbf{V}_1(i) - \mathbf{V}_2(i)| \tag{2}$$

$$D_{ccm} = \sum_{i=1}^{L_x} \sum_{j=1}^{L_y} |\mathbf{O}_1(i,j,d) - \mathbf{O}_2(i,j,d)| \tag{3}$$

where k denotes the number of quantized colors in the image, and the value of d is set to 1 in the following experiments. From the experiments, it is found that setting k to be 1, 2 or 3 does not influence clothing segmentation significantly.

2.3. Color texture segmentation

Given an input image, we don't have to segment the overall image; only the clothing where the user marks a seed point is to be separated from the background. Therefore, a region growing procedure starting from the seed point seems effective for the clothing segmentation. In the first step of the growing method, a seed block which embodies enough information (which will be described later) about the color and texture of the clothing is determined from the given seed point. Next, each of the eight nearest neighboring blocks of the seed block (as Fig. 10 shows) is checked to see whether it has a similar color occurrence vector and color co-occurrence matrix to the seed block. If it does, this block is considered to be on the clothing and is accepted. Otherwise, this block may be completely or partially on the background. If the whole block is on the background, it is rejected. But if it is partially on the background, this block is split into four subblocks, each of which is checked again. The growing process is repeated for each accepted block until no more blocks are accepted. Finally, all the accepted blocks are considered to be the constituents of the clothing to be segmented. In the growing process, whether a block is accepted, rejected or split depends on a dissimilarity measure (which will be defined later). If this measure is small enough, then this

1	1	2	2
1	2	2	2
4	4	4	4
3	3	4	4

(a)

	Color label 1	2	3	4
Occurrence frequency	0.1875	0.3125	0.125	0.375

(b)

Color label	1	2	3	4
1	0.053	0.039	0	0.026
2	0.039	0.211	0	0.092
3	0	0	0.026	0.079
4	0.026	0.092	0.079	0.237

(c)

Fig. 9. Color vector and color co-occurrence matrix of an image block. (a) 4×4, 4-color image block, (b) color occurrence vector, (c) the color co-occurrence matrix with distance $d = 1$.

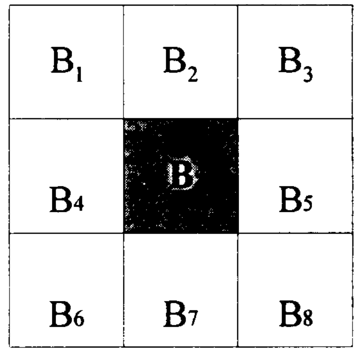


Fig. 10. Eight nearest neighboring blocks $B_1, B_2, \ldots, B_8$ of B.

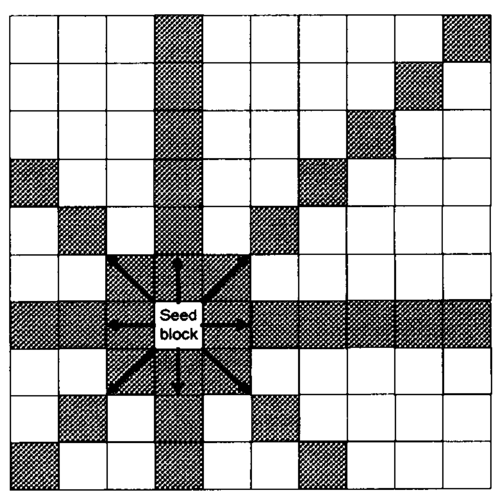


Fig. 11. The gray blocks radiating from the seed block are used in constructing the D_{cv} and D_{ccm} histograms.

block is accepted. If it is large enough, this block is rejected. Otherwise, this block is split. All these decisions are based on several threshold values (which will be discussed later). Hence, determination of these thresholds is crucial.

In the following, a detailed description of the clothing segmentation process will be given, including determination of the seed block, selection of the threshold values and the region growing algorithm.

2.3.1. Seed block determination

Because the input seed point does not have enough information about the clothing, a seed block which contains sufficient information about the color and texture of the clothing should be located for comparison during the growing process. Determination of the seed block is accomplished through an iteration process. We use the seed point as the center of the seed block and increase the block size gradually until the color and texture information contained in the seed block is enough. The seed block is assumed to be of a size

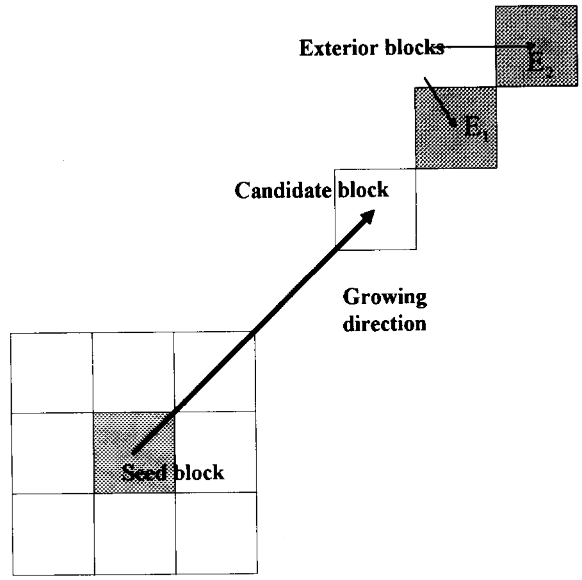
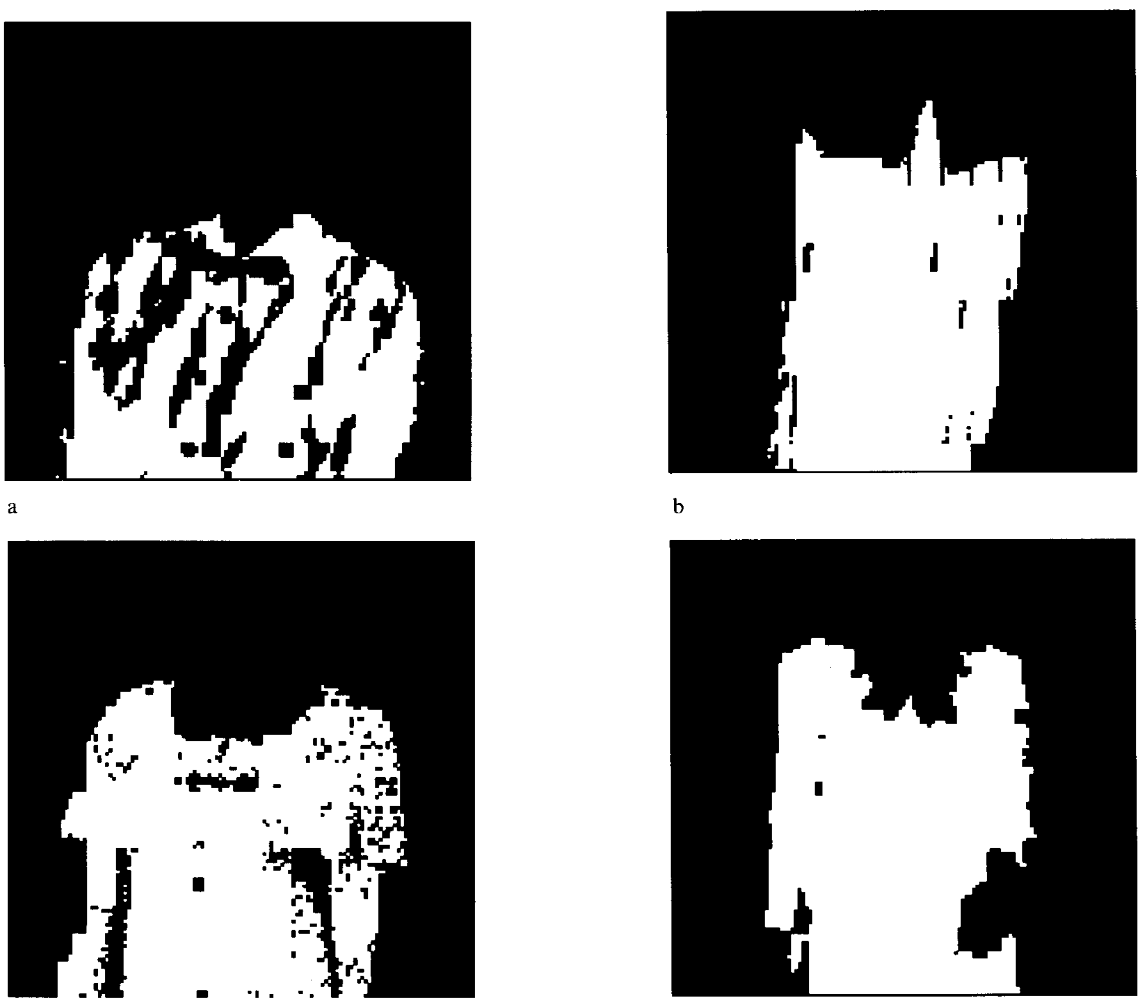


Fig. 12. Example of the exterior blocks for a candidate block.



a b c d

Fig. 13. Region-growing results of images in Fig. 8.

$(2k+1) \times (2k+1), k \geq 1$. Initially, the size of the seed block is set to 3×3. Next, we take a 5×5 reference block whose center is also the seed point, and calculate the values of D_{cv} and D_{ccm} between the seed block and the reference block according to Eqs. (2) and (3). When the values of D_{cv} and D_{ccm} are smaller than the thresholds T_{bcv} and T_{bccm}, respectively, the information contained in the larger reference block is not considered to be more than that in the smaller seed block. That is, the information contained in the seed block is enough to specify the color and texture of the clothing. Otherwise, the size of the seed block is increased and the above checking procedure is done again. The process is repeated until the difference measures D_{cv} and D_{ccm} between the seed block and its corresponding reference block are smaller than T_{bcv} and T_{bccm}, respectively. In this study, both the values of T_{bcv} and T_{bccm} are 0.2, which are obtained from experiments. The major steps of the seed block determination process is given as follows.

Algorithm. Determination of the seed block

Input. (1) A quantized image; (2) a seed point; (3) thresholds T_{bcv} and T_{bccm}.

Output. A seed block.

Step 1. Set $k = 3$. Initialize a $k \times k$ seed block centered at the position of the seed point.

Step 2. Locate a $(k+2) \times (k+2)$ reference block with the same center as the seed block.

Step 3. Compute the values of D_{cv} and D_{ccm} betweeen the seed block and the reference block.

Step 4. If $D_{cv} < T_{bcv}$ and $D_{ccm} < T_{bccm}$, go to Step 6.

Step 5. Increase the size of the seed block to $(k+2) \times (k+2)$. Set $k = k + 2$. Go to Step 2.

Step 6. Output the seed block. **Stop.**

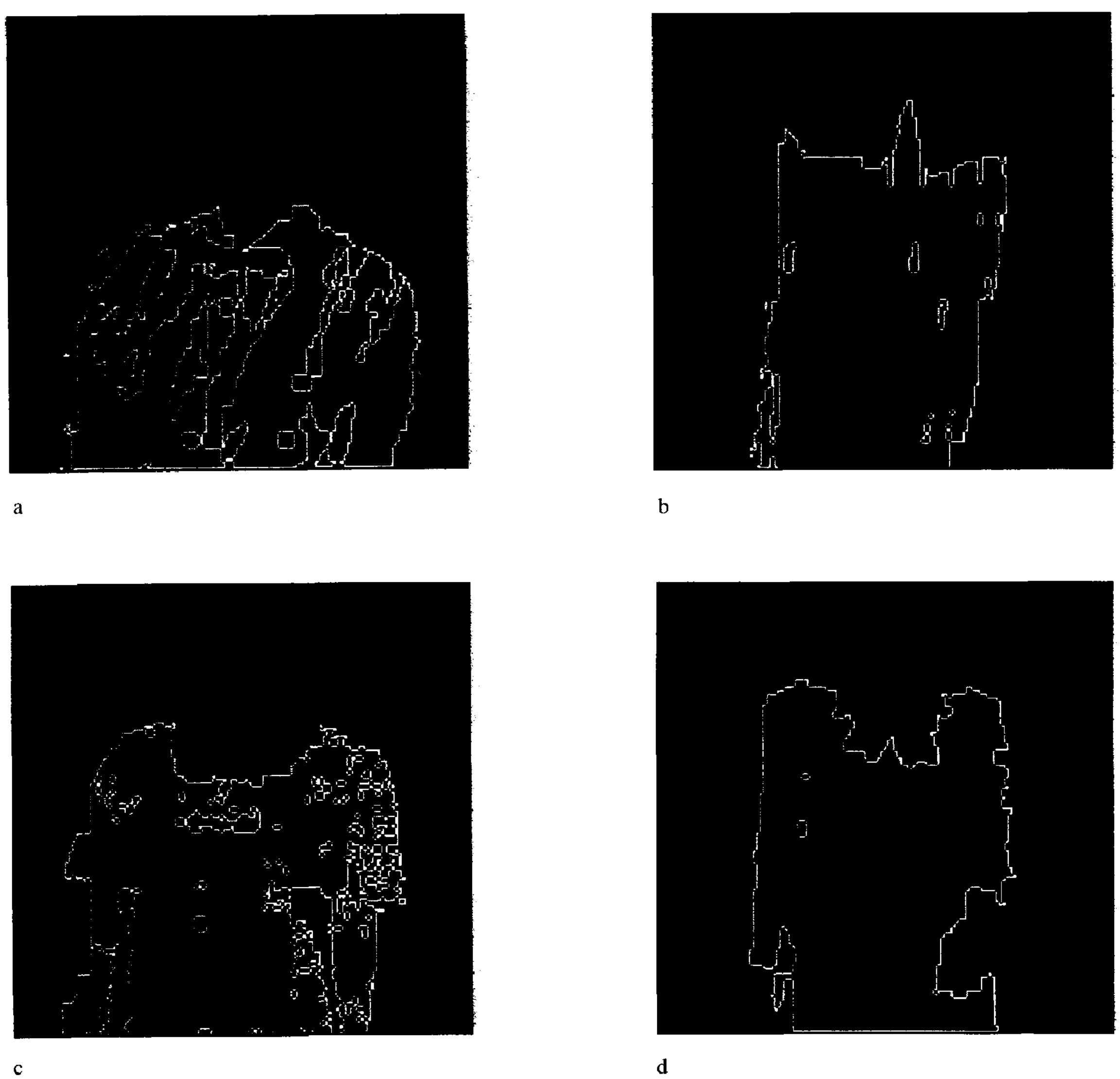


Fig. 14. Edge-detecting results of binary images in Fig. 13.

2.3.2. Threshold selection

In the region growing process, whether a block is accepted, rejected or split depends on the distance measures of D_{cv} and D_{ccm} between this block and the seed block. If their values are less than thresholds T_{cv1} and T_{ccm1}, respectively, the block is accepted. But if they are respectively larger than thresholds T_{cv2} and T_{ccm2}, the block is rejected. Otherwise, the block is split into four smaller ones to be checked further. Determination of these four threshold values T_{cv1}, T_{cv2}, T_{ccm1} and T_{ccm2} are crucial. Fixing these threshold values will prohibit flexible segmentation. Adapting these thresholds to changing input images will provide more flexibility. In the study, histograms constructed from the values of D_{cv} and D_{ccm} are used to determine the four thresholds.

First, the color occurrence vectors and color co-occurrence matrices of the blocks radiating from the seed block in eight different directions, as shown in Fig. 11, are computed. The distance measures D_{cv} and D_{ccm} of these blocks with respect to the seed block are then used to construct the D_{cv} and D_{ccm} histograms, respectively. Since the blocks on the clothing often have similar color occurrence vectors to the seed block, their D_{cv} values are small. However, the color occurrence vectors of the blocks on the background might be significantly different from that of the seed block; their D_{cv} values tend to be larger. We assume that the clothing to be segmented occupies a significant percentage in the image. Thus, the D_{cv} histogram has two main peaks or clusters, one peak formed from the blocks belonging to the background and the other from the blocks on the clothing. We can apply the 2-means algorithm to the D_{cv} histogram to separate the two clusters. Then, the mean of the cluster with smaller D_{cv} values is selected as the value of T_{cv1}, and the decision boundary of the two clusters is used for the threshold T_{cv2}. Similarly, the thresholds T_{ccm1} and T_{ccm2} can be determined from the D_{ccm} histogram.

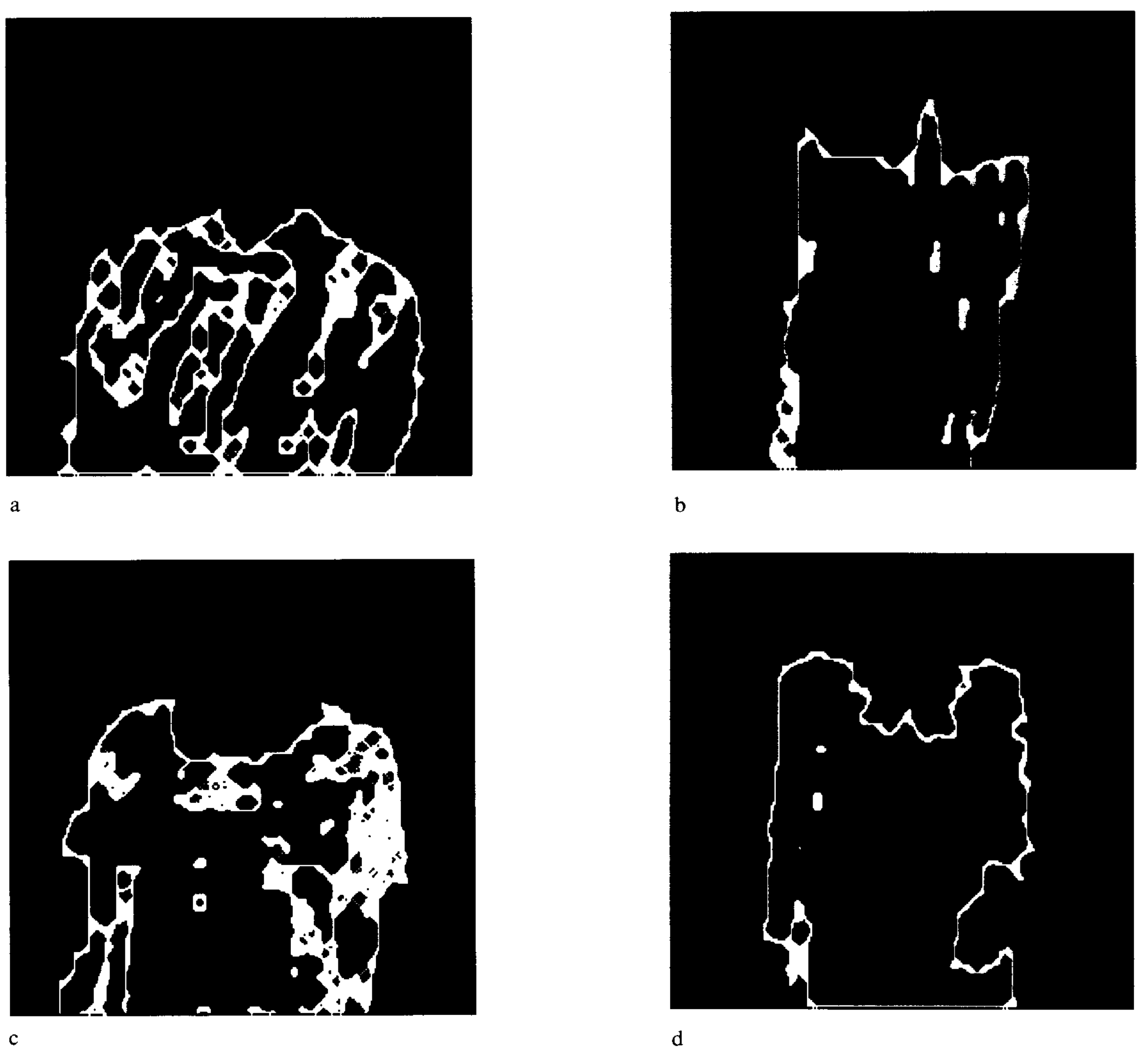


Fig. 15. Edge-linking results of images in Fig. 14.

2.3.3. Region growing algorithm

After locating the seed block, denoted by S, clothing segmentation is achieved by a region growing process. A queue is used in the process and initialized with the eight nearest neighboring blocks of S. Each block in the queue is a candidate for a clothing block. For each candidate block, say C, in the queue, if the values of D_{cv} and D_{ccm} between S and C are smaller than T_{cv1} and T_{ccm1}, respectively, C is accepted as a clothing block and its eight nearest neighboring blocks are inserted into the queue. But if the values of D_{cv} and D_{ccm} are larger than T_{cv2} and T_{ccm2}, respectively, the candidate block is very likely to be on the background, and so is rejected.

When either D_{cv} is larger than T_{cv2} or D_{ccm} larger than T_{ccm2}, the candidate block may be on the boundary of the clothing. Therefore, the candidate block contains only a portion of patterns on the seed block. To process the candidate block in this case, the concept of 'subfeature' is introduced. Color occurrence vector V_c of the candidate block is said to be a subfeature of color occurrence vector V_s of the seed block if $V_s(i) = 0$ implies $V_c(i) = 0$. Similarly, color co-occurrence matrix $\mathbf{O}_c$ of the candidate block is a subfeature of $\mathbf{O}_s$ of the seed block if $\mathbf{O}_s(i,j,d) = 0$ implies $\mathbf{O}_c(i,j,d) = 0$. Two exterior blocks, say E_1 and E_2, of the candidate block are defined, as Fig. 12 shows. Let the color occurrence vectors of E_1 and E_2 be $\mathbf{V}_{e1}$ and $\mathbf{V}_{e2}$, respectively, and the color co-occurrence matrices of E_1 and E_2 be $\mathbf{O}_{e1}$ and $\mathbf{O}_{e2}$, respectively. When both $\mathbf{V}_{e1}$ and $\mathbf{V}_{e2}$ are subfeatures of $\mathbf{V}_s$, and both $\mathbf{O}_{e1}$ and $\mathbf{O}_{e2}$ are subfeatures of $\mathbf{O}_s$, the clothing and its background have similar-looking texture and color information. Hence, it is very likely that the candidate block is on the background, and it is rejected. But when the above subfeature conditions are not satisfied, the clothing and its background are dissimilar in colors and textures. Therefore, the candidate block

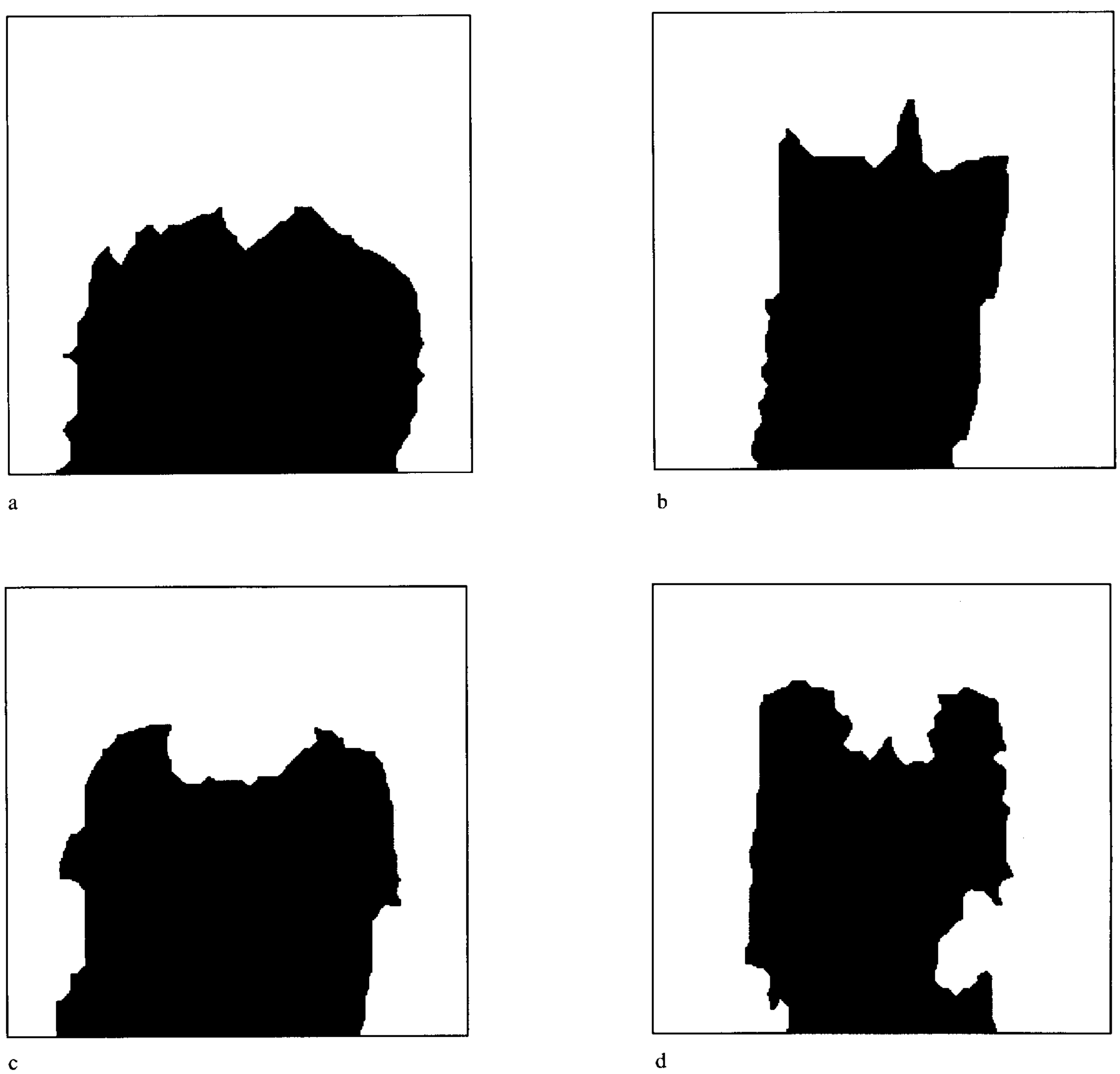


Fig. 16. Background-searching results of images in Fig. 15.

seems to lie on the boundary of the clothing. That is, a part of it is on the clothing and the remainder on the background. In this case, the candidate block is split into four equal-sized blocks for further checking.

For conditions not discussed above, the candidate block may be on the boundary of the clothing, but its texture and color information are not destroyed seriously by shadows, highlights or folds. It is split into four subblocks. These subblocks and the eight nearest neighboring blocks of the candidate block are all inserted into the queue for further checking.

Note that when the candidate block is of a smaller size than the seed block, a distinct comparison criterion should be used. In this case, the candidate block may contain only a portion of patterns on the seed block. We also use the above subfeature checking method to test if the smaller candidate block falls on the clothing. When $\mathbf{V}_c$ and $\mathbf{O}_c$ of the candidate block are subfeatures of $\mathbf{V}_s$ and $\mathbf{O}_s$ of the seed block, respectively, the candidate block is accepted as a clothing block. Otherwise, it is split for further checking. The major steps of the region growing process are given.as follows:

Algorithm: Region growing method.

Input. (1) A quantized image; (2) the seed block; (3) thresholds T_{cv1}, T_{cv2}, T_{ccm1} and T_{ccm2}.

Output. A set of clothing blocks.

Step 1. Compute the color vector, $\mathbf{V}_s$, and the color co-occurrence matrix, $\mathbf{O}_s$, of the seed block.

Step 2. Initialize a queue, Q, with the eight nearest neighboring blocks of the seed block.

Step 3. If Q is not empty, take out a candidate block from Q. Otherwise, **stop**.

Step 4. Compute the color vector, $\mathbf{V}_c$, and the color co-occurrence matrix, $\mathbf{Q}_c$, of the candidate block.

Step 5. If the candidate block and the seed block are of unequal size, go to Step 11.

Step 6. Compute D_{cv} of $\mathbf{V}_s$ and $\mathbf{V}_c$ and D_{ccm} of $\mathbf{O}_s$ and $\mathbf{O}_c$.

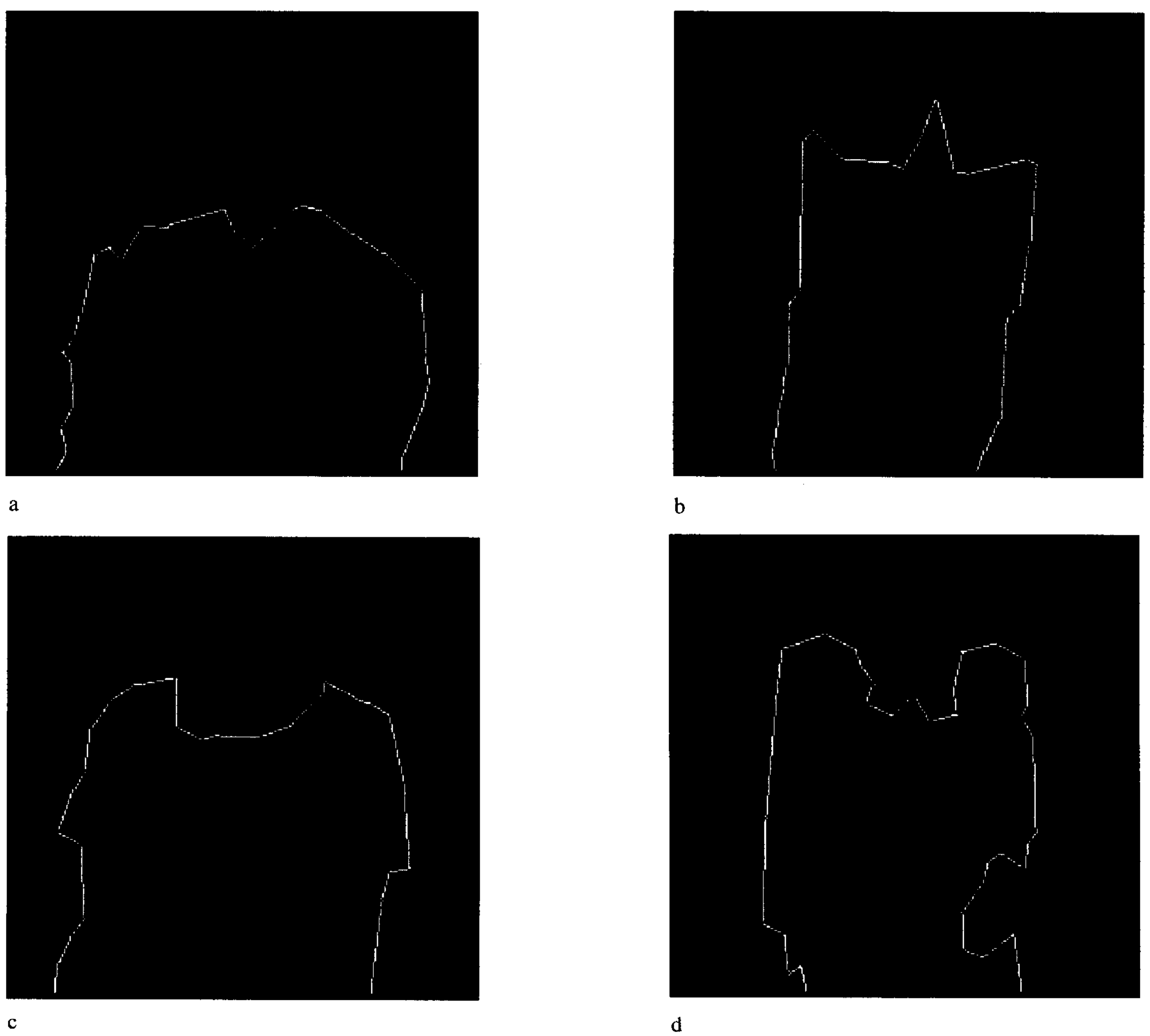


Fig. 17. Boundary-smoothing results of images in Fig. 16.

Step 7. If $D_{cv} \leq T_{cv1}$ and $D_{ccm} \leq T_{ccm1}$, accept the candidate block as a clothing block, insert into Q its nearest neighboring blocks which are not yet put into Q, and go to Step 3.

Step 8. If $D_{cv} > T_{cv2}$ and $D_{ccm} > T_{ccm2}$, reject the candidate block and go to Step 3.

Step 9. If either $D_{cv} > T_{cv2}$ or $D_{ccm} > T_{ccm2}$:

Step 9.1. Find the exterior blocks of the candidate block, E_1 and E_2, and compute $\mathbf{V}_{e1}$, $\mathbf{O}_{e1}$, $\mathbf{V}_{e2}$ and $\mathbf{O}_{e2}$

Step 9.2. If $\mathbf{V}_{e1} \subseteq \mathbf{V}_s$, $\mathbf{V}_{e2} \subseteq \mathbf{V}_s$, $\mathbf{O}_{e1} \subseteq \mathbf{O}_s$ and $\mathbf{O}_{e1} \subseteq \mathbf{O}_s$, reject the candidate block. Otherwise, split the candidate block to four subblocks and insert the four subblocks into Q.

Step 9.3. Go to Step 3.

Step 10. Split the candidate block. Insert into Q its four subblocks and its nearest neighboring blocks which are not yet put into Q. Go to Step 3.

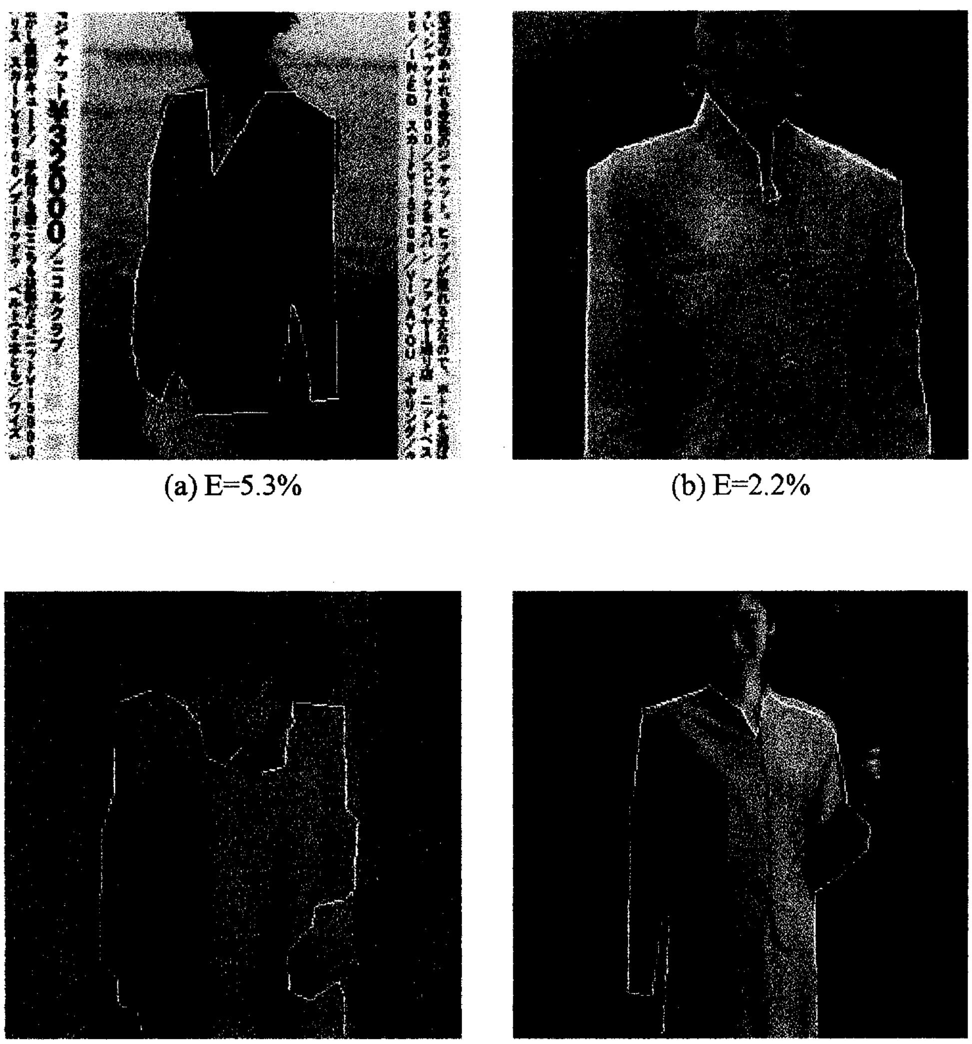
Step 11. If $\mathbf{V}_c \subseteq \mathbf{V}_s$ and $\mathbf{O}_c \subseteq \mathbf{O}_s$, accept the candidate block as a clothing block and go to Step 3.

Step 12. If the candidate block is not of size 1×1, split the block and insert its four subblocks into Q. Otherwise, reject this pixel block. Go to Step 3.

Region-growing results of the quantized images in Fig. 8 are shown in Fig. 13. In the figure, the detected clothing blocks are displayed in white (1) with the background in black (0). Such binary images will be used as input in the post-processing phase.

2.4. Post-processing

After region growing, the boundary of the set of clothing blocks in the resulting image is often jagged. Moreover, there are blocks which are on the clothing but which are classified to the background part in the region growing process, as shown in Fig. 13. Therefore, post-processing is necessary to obtain a smooth



(a) E=5.3% (b) E=2.2%

(c) E=8.2% (d) E=3.0%

Fig. 18. Segmentation results of non-textured clothing. (a) $E = 5.3\%$, (b) $E = 2.2\%$, (c) $E = 8.2\%$, (d) $E = 3.0\%$.

boundary of the desired clothing. There are four steps in post-processing, including mainly edge detection, edge linking, background search and boundary smoothing.

In post-processing, a simple edge detector is first applied to the resulting binary image of region growing. If one of the 8-neighbors of a white pixel is black, then the white pixel is regarded as an edge point by the edge detector. Fig. 14 shows the edge-detecting results of images in Fig. 13. From the figure, it is found that edges on the clothing boundary may not be linked well. Therefore an edge linker is applied next. In edge linking [30], each pair of edge pixels in a 7×7 window is linked by a line segment. Fig. 15 shows the edge-linking results of images in Fig. 14. After edge linking, what we obtain is also a binary image, where 1 denotes an edge pixel and 0 otherwise. Because it is difficult to search the clothing region from the linked edge pixels, we search the background region directly. The output of background searching is also a binary image, where 1 denotes a pixel on the background and 0 otherwise. Fig. 16 shows the background-searching results of images in Fig. 15. The detailed procedure is described as follows, where the initial input point can be selected as a pixel of a rejected block in the region growing process which is far from the seed block:

Algorithm: Background search.
Input. (1) $N_x \times N_y$ binary image I_1; (2) an initial point on the background.
Output. (2) An $N_x \times N_y$ binary image I_2.

Step 1. Push the initial point into a stack, S. Initialize the output image, I_2, each pixel of which is assigned a zero value.
Step 2. If S is not empty, pop out a point, (x_q, y_q), from S. Otherwise, **stop**.
Step 3. If (x_q, y_q) on I_2 has been set as a background point, i.e. assigned the value 1, go to Step 2.

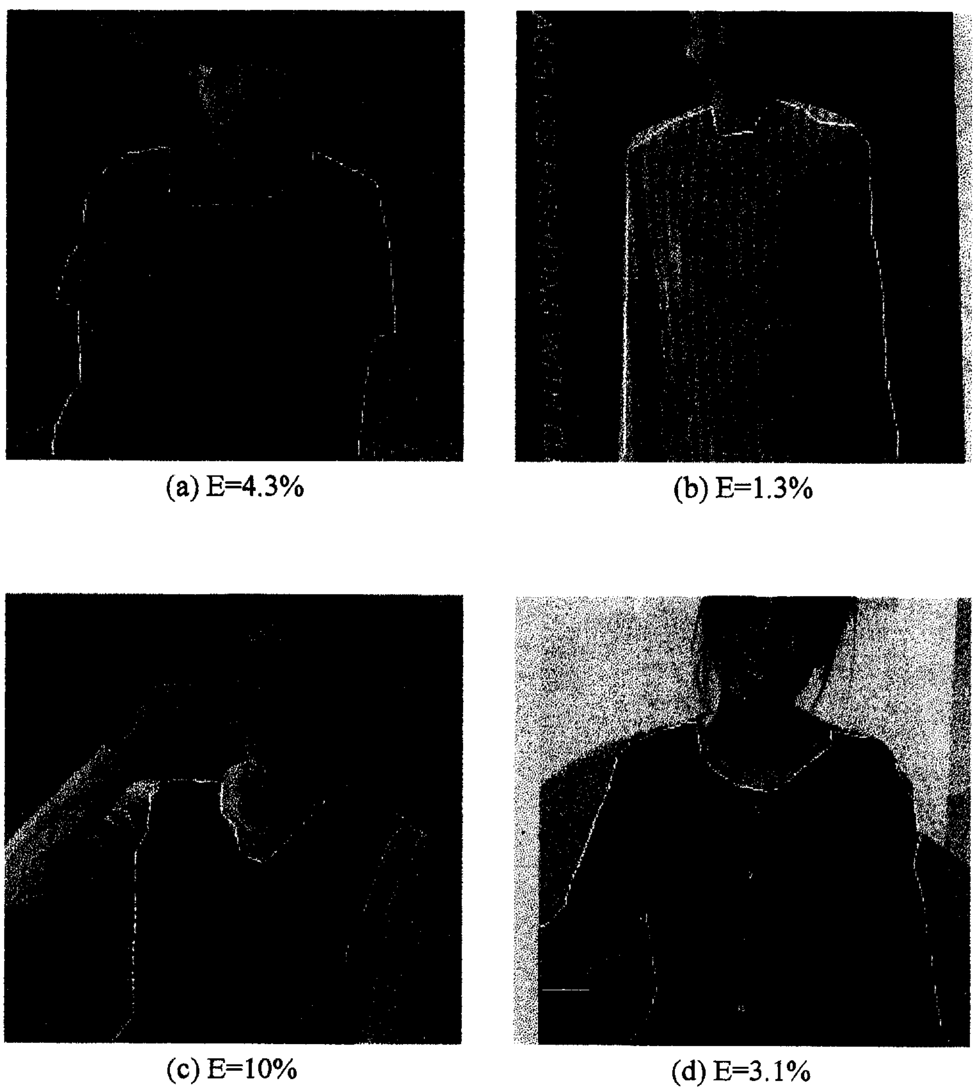


Fig. 19. Segmentation results of textured clothing. (a) $E = 4.3\%$, (b) $E = 1.3\%$, (c) $E = 10\%$, (d) $E = 3.1\%$.

Step 4. Let $x = x_q + 1$.

Step 5. If $x < N_x$ and (x, x_q) on I_1 is a background point, set (x, y_q) on I_2 as a background point. Otherwise, go to Step 7.

Step 6. If $(x, y_q + 1)$ on I_1 is a background point but $(x, y_q + 1)$ on I_2 is not, push pixel $(x, y_q + 1)$ into S. If $(x, y_q - 1)$ on I_1 is a background point and $(x, y_q - 1)$ on I_2 is not, push pixel $(x, y_q - 1)$ into S. Increase x by 1 and go to Step 5.

Step 7. Let $x = x_q - 1$.

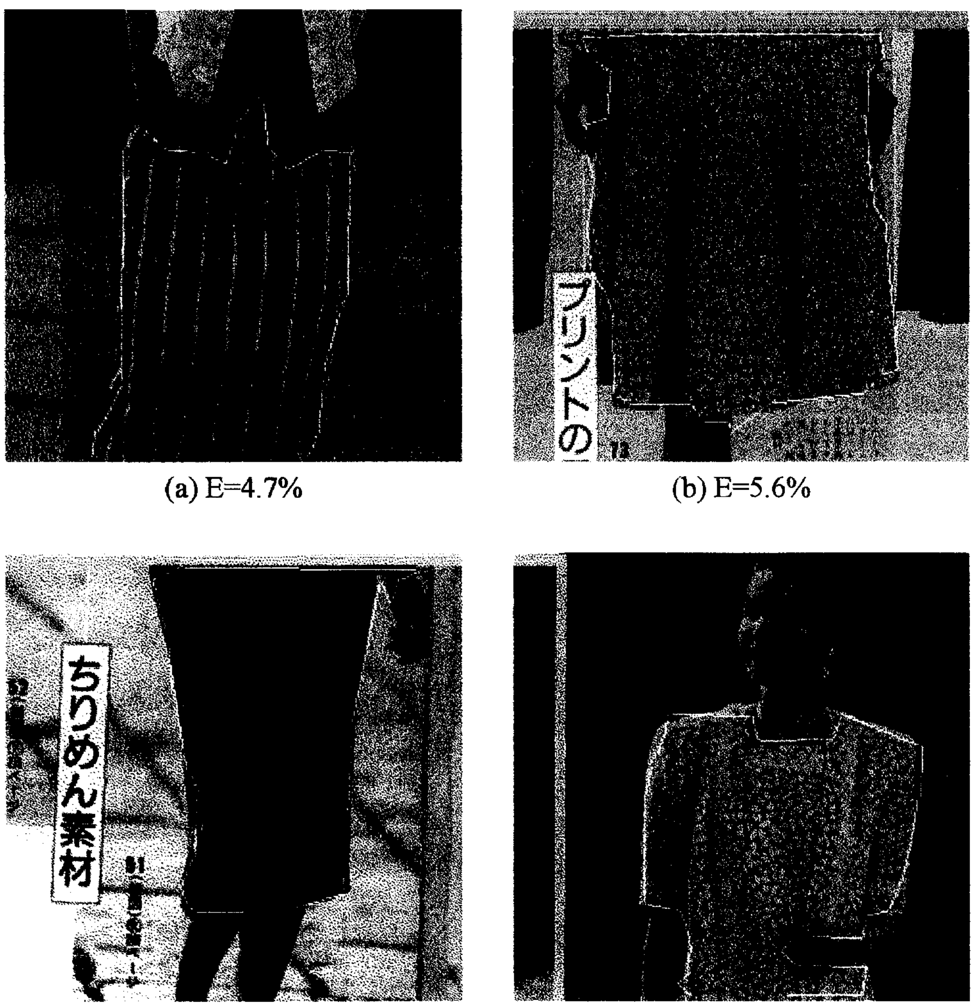
Step 8. If $x > 0$ and (x, y_q) on I_1 is a background point, then set (x, y_q) on I_2 as a background point. Otherwise, go to Step 2.

Step 9. If $(x, y_q + 1)$ on I_1 is a background point but $(x, y_q + 1)$ on I_2 is not, push pixel $(x, y_q + 1)$ into S. If $(x, y_q - 1)$ on I_1 is a background point and $(x, y_q - 1)$ on I_2 is not, push pixel $(x, y_q - 1)$ into S. Decrease x by 1 and go to Step 8.

After extracting the background region, the remainder on the image is the clothing part. To find the clothing region, the edge detection operator is applied again. The region which contains the seed point is the clothing part which we wish to separate from its background. Therefore, only the edges of this region are of interest. To obtain the edges of the clothing region, we search outwards from the seed point to find an edge pixel. Then a boundary tracing method [25] starting from the edge pixel is used to extract the overall clothing boundary. Since the boundary obtained is jagged, line fitting [31] is performed finally on the edge pixels. The boundary-smooth results of images in Fig. 16 are shown in Fig. 17.

3. Experimental results

The proposed approach has been implemented in the C language on a 33 MHz 486 PC with a



(a) E=4.7% (b) E=5.6%

(c) E=3.8% (d) E=8.6%

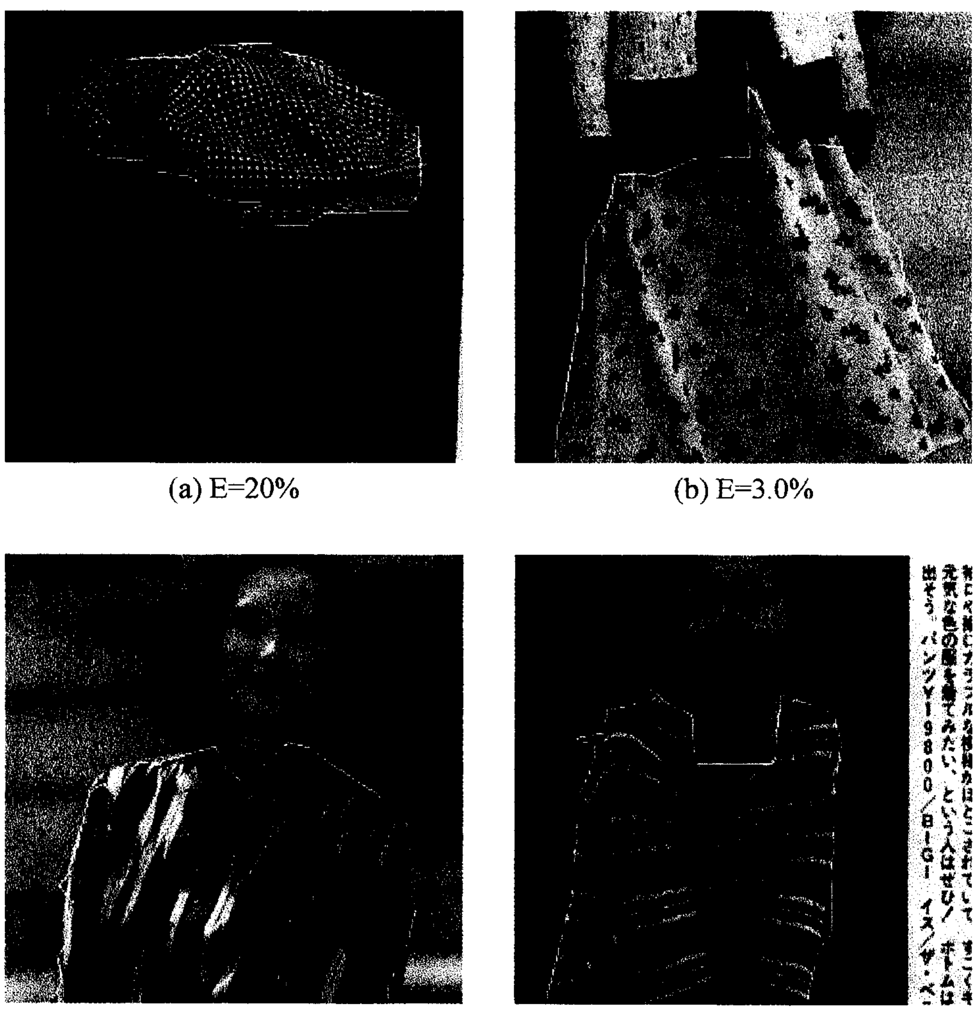
Fig. 20. Segmentation results of clothing with larger pattern prints. (a) $E = 4.7\%$, (b) $E = 5.6\%$, (c) $E = 3.8\%$, (d) $E = 8.6\%$.

TARGA+ developers kit. Various model images are obtained by scanning pictures in some magazines, such as LADY BOUTIQUE, JUNIE and NON-NO. To evaluate the experimental results quantitatively, an evaluation function is defined. Let A be the area of the clothing region whose boundary is specified by hand, and B be that whose boundary is found by the proposed segmentation approach. The evaluation function is defined as

$$E(A,B)=\frac{(A\cup B)-(A\cap B)}{A\cap B}.$$

A large value of the evaluation function means that the manually segmented result is significantly different from the result obtained by the proposed method.

Several experimental results are shown in Figs. 18–21. Fig. 18 shows segmentation results of four non-textured clothing items. The shadows and highlights on the clothing are evident. Fig. 19 shows the segmentation results of four textured clothing items. In these images, shadows, highlights and folds are apparent on the clothing. Fig. 20 shows the segmentation results of four textured clothing items with larger patterns. Fig. 21 gives the segmentation results of four textured clothing items with serious folds. The value of the evaluation function for each segmented result is given in the figures. The average value of the evaluation functions in Figs. 18–21 is about 5.6%. It takes about one and a half minutes for each 256 × 256 image to separate the desired clothing from the background. From these figures, some defects can be found in the segmentation results, which still remain to be solved. First, when the clothing to be segmented and its background have similar hue attributes, some blocks on the background will be mistaken for the clothing blocks. In addition, when there are serious shadows or highlights on the clothing, the blocks on the clothing may be misclassified.



(a) E=20%

(b) E=3.0%

(c) E=3.1%

(d) E=2.6%

Fig. 21. Segmentation results of clothing with serious folds. (a) $E=20\%$, (b) $E=3.0\%$, (c) $E=3.1\%$, (d) $E=2.6\%$.

4. Conclusion

In this paper, we have proposed a color texture segmentation method for a computer-aided fashion design system. A color quantization method based on the circular hue histogram has been applied to reduce the number of colors in the image, and to reduce the effects of shadows and highlights on the image. The color occurrence vector and color co-occurrence matrix have been used as color texture features of each image block. These two color texture features are tolerant to fold and orientation variations on the textures. Using these two features, blocks on the clothing can be separated from the background using a region growing method. Experimental results have shown the feasibility of the proposed approach.

Further research may be directed to the following topics. First, improve the speed by parallelizing critical sections of the proposed approach. Second, achieve better performance by resolving the serious shadow/highlight problem and by designing more texture features.

Acknowledgements

This work was supported partially by National Science Council, Republic of China under Grant NSC84-2213-E-007-020.

References

[1] Y. Ohta, T. Kanade and T. Sakai, Color information for region segmentation. Computer Graphics and Image Processing, 13(2) (1980) 224–241.
[2] S. Tominaga, Color image segmentation using three perceptual attributes. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Miami Beach, FL, (1986) pp. 628–630.
[3] S. Tominaga, Color classification of natural color images. Color Research and Application, 17(4) (1992) 230–239.
[4] A. Sarabi and J.K. Aggarwal, Segmentation of chromatic images. Pattern Recognition, 13(6) (1981) 417–427.
[5] M. Celenk, A color clustering technique for image segmentation. Computer Vision, Graphics, and Image Processing, 52(2) (1990) 145–170.
[6] A.P. Witkin, Scale-space filter: a new approach to multi-scale description. In S. Ullman and W. Richards (eds.), Image Understanding, Ablex, NJ, 1984, pp. 79–95.
[7] M.J. Carlotto, Histogram analysis using a scale-space approach. IEEE Trans. Pattern Analysis and Machine Intelligence, 9(1) (1987) 121–129.
[8] C.L. Huang, T.Y. Cheng and C.C. Chen, Color images' segmentation using scale space filter and Markov random field. Pattern Recognition, 25(10) (1992) 1217–1229.
[9] Y.M. Lim and S.U. Lee, On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques. Pattern Recognition, 23(9) (1990) 935–952.
[10] J. Liu and Y.H. Yang, Multiresolution color image segmentation. IEEE Trans. Pattern Analysis and Machine Intelligence, 16(7) (1994) 689–700.
[11] I. Andreadis, M.A. Browne and J.A. Swift, Image pixel classification by chromaticity analysis. Pattern Recognition Lett., 11(1) (1990) 51–58.
[12] G.J. Klinker, S.A. Shafer and T. Kanade, A physical approach to color image understanding. Int. J. Computer Vision, 4(1) (1990) 7–38.
[13] S.A. Shafer, Using color to separate reflection components. Color Research and Application, 10(4) (1985) 210–218.
[14] G.J. Klinker, S.A. Shafer and T. Kanade, Using a color reflection model to separate highlights from object color. Proc. Int. Conf. Computer Vision, London, UK, 1987, pp. 145–150.
[15] G.J. Klinker, S.A. Shafer and T. Kanade, The measurement of highlights in color images. Int. J. Computer Vision, 2(1) (1988) 7–32.
[16] J. Scharcanski, J.K. Hovis and H.C. Shen, Representing the color aspect of texture images. Pattern Recognition Lett., 15(2) (1994) 191–197.
[17] J. Scharcanski, H.C. Shen and A.P. Alves da Silva, Colour quantization for colour texture analysis. IEEE Proc. Part E: Computers and Digital Techniques, 140(2) (1993) 109–114.
[18] D.K. Panjwani and G. Healey, Unsupervised segmentation of textured color images using markov random field models. Proc. Int. IEEE Conf. Computer Vision and Pattern Recognition, New York, USA, 1993, pp. 776–777.
[19] T. Caelli and D. Reye, On the classification of image regions by colour texture and shape. Pattern Recognition, 26(4) (1993) 461–470.
[20] J.M. Kasson and W. Plouffe, An analysis of selected computer interchange color spaces. ACM Trans. Graphics, 11(4) (1992) 373–405.
[21] H. Levkowitz and G.T. Herman, GLHS: A generalized lightness, hue, and saturation color model. Computer Vision, Graphics, and Image Processing, 55(4) (1993) 271–285.
[22] R.W.G. Hunt, The Reproduction of Colour in Photography, Printing and Television, Fountain, UK, 1987.
[23] G. Wyszecki and W.S. Stiles, Color Science Concepts and Methods, Quantitative Data and Formulae, Wiley-Interscience, USA, 1982.
[24] A.K. Jain, Fundamentals of Digital Image Processing, Prentice-Hall, NJ, 1989.
[25] R.C. Gonzalez and R.E. Woods, Digital Image Processing, Addison-Wesley, Reading, MA, 1992.
[26] Y.F. Li and D.C. Tseng, Color image segmentation using circular histogram thresholding. Proc. IPPR Conf. Computer Vision, Graphics, and Image Processing, Taiwan, ROC, 1994, pp. 175–183.
[27] R.M. Haralick, K. Shanmugan and I. Dinstein, Textural features for images classification. IEEE Trans. System, Man, and Cybernetics, 3(6) (1973) 610–621.
[28] D.C. He, L. Wang and J. Guibert, Texture discrimination based on an optimal utilization of texture features. Pattern Recognition, 21(2) (1987) 141–146.
[29] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
[30] S.M. Kao, T.K. Ning, C.W. Wu and H. Peng, The identification of the focused object in an image with defocused background. Proc. IPPP Conf. Computer Vision, Graphics, and Image Processing, Taiwan, ROC, 1994, pp. 1–4.
[31] J. Sklansky and V. Gonzalez, Fast polygonal approximation of digitized curves. Pattern Recognition, 12(5) (1980) 327–331.