Optimal Gabor filter design for texture segmentation

D. M. Tsai
Machine Vision Lab.
Department of Industrial Engineering and Management
Yuan-Ze University, Chung-Li, Taiwan, R.O.C.
E-mail: iedmtsai@saturn.yzu.edu.tw

1. INTRODUCTION

Gabor-filter based methods have been successfully applied for a variety of machine vision applications, such as texture segmentation [1-10], edge detection [11], object detection [12,13], image representation [14], and recognition of handwritten numerals [15]. In this paper we consider the problem of segmenting textured images using a single Gabor filter.

Texture segmentation involves accurately partitioning an image into differently textured regions. It requires simultaneous measurements in both the spatial and the spatial-frequency domains. Gabor filters are well recognized in the recent past as a joint spatial/spatial-frequency representation of textures. Daugman [16] has shown that Gabor filters have optimal joint localization in both the spatial and the spatial-frequency domains. In addition, they are bandpass filters, which are inspired by a multi-channel filtering theory for processing visual information in the early stages of the human visual system [17, 18].

A 2-D Gabor function is an oriented complex sinusoidal grating modulated by a 2-D Gaussian function. The parameters of the Gabor function are specified by the

frequency, the orientation of the sinusoid (or represented by the center frequency), and the scale of the Gaussian function. Local orientations and spatial frequencies explicit in Gabor filters are therefore used as the key features for texture processing. The input image is generally filtered by a family of Gabor filters tuned to several resolutions and orientations. However, it may not be computationally convenient or feasible to apply a large number of filters responding at multiple resolutions and orientations to an image.

Accurate segmentation only occurs if the parameters defining the Gabor filters are suitably chosen. Two main methods have been proposed in the literature for selecting Gabor filters for texture segmentation: the filter-bank approaches and the filter-design approaches [10]. In filter-bank approaches [1, 2, 6, 11, 13], the filter parameters are present *ad hoc* and are not necessarily optimal for a particular processing task. A family of Gabor filters is usually reported with frequency bandwidth in octaves and orientation bandwidth in 45 degrees. Therefore, the orientation parameters are generally selected at four directions: $0°$, $45°$, $90°$ and $135°$ [4, 11-13, 15]. The restriction to four orientations is made for computational saving. For an image array with a width of W pixels, the following values of frequency are typically employed [4, 12] :

$$1\sqrt{2}, 2\sqrt{2}, 4\sqrt{2}, \Lambda, (W/4)\sqrt{2} \quad \text{cycles image-width}^{-1}$$

The total number of Gabor filters in the bank is given by $4\log_2(W/2)$.

The scale parameter of the Gaussian function is generally selected intuitively and assumed to be a constant. Dunn et al. [19] have proposed guidelines on selecting values for the scale parameter. The problem with the filter-bank approach is that a

few values of the free parameters can result in a very large set of filters, which entail a tremendous number of convolutions.

Currently human intervention is required to assist in selecting the appropriate filter parameters for texture segmentation. Namuduri et al. [11] use a recursive filtering method to generate a family of Gabor filters and their responses starting from a high resolution and proceeding towards the lowest resolution in step of an octave. A method that generates the same responses in the reverse order is also proposed. Raghu and Yegnanarayana [1] present a Bayesian approach for the supervised segmentation of textured images. The texture features are extracted by filtering the given image using a filter bank. The selection of filters is done by visually inspecting the filtered images, so as to make out the filters that characterize different textures in the image. Van Hulle and Tollenaere [6] present a neural network approach for texture processing. The input to the neural networks is based on local energy maps obtained by filtering the textures with a bank of quadrature pair Gabor filters with different preferred orientations and spatial frequencies. Besides the large computational burden imposed by a large bank of filters, it produces an output feature vector with high dimensionality and, therefore, requires a complicated classifier for texture discrimination.

Some texture segmentation tasks may not require a large bank of filters for effective performance. Therefore, in filter-design approaches [3-5, 8-10] only one or a few filters for a particular application are designed in an effort to reduce the difficulties of filter-bank approaches. The selection of best filters is generally based on *a priori* knowledge of the textural properties derived from a spectral Fourier analysis of the entire image [3, 20]. The analysis is followed by a search for the

most significant spectral components and then the preferred frequencies and orientations of the Gabor filters. Teuner et al. [5] point out that as the Fourier transform is a global transformation, this approach makes an optimum extraction of important local features impossible.

Clark and Bovik [21] propose a supervised approach based on computing the discrete Fourier transform (DFT) of sample textured regions. The 2-D frequency component that differs most between regions is then selected as the center frequency of the Gabor filter. The performance of the DFT method is relatively poor since it does not consider filter bandwidth when determining the center frequency. Bovik et al. [3] select a set of channel filters for the segmentation of a textured image. A simple peak-finding algorithm applied to the image power spectrum is used to guide the choice of the filter center frequencies. The filter parameters are chosen using a limited amount of human intervention. For strongly oriented textures, the most significant spectral peak along the orientation is chosen. For periodic textures, the lower fundamental frequency is chosen. Finally, for nonoriented textures, the center frequencies are chosen from the two largest maxima. Teuner et al. [5] employ an iterative version of the pyramidal Gabor transform [22] to select a set of Gabor filters for unsupervised texture segmentation. The parameter selection and tuning are based on the detection of spectral "nonconforming" components, which are computed from a contrast function calculated for all spectral components at different hierarchical levels of the Gabor pyramid. Jain and Farrokhnia [4] and Jain et al. [12] propose a systematic filter selection scheme for texture segmentation and object recognition. It is based on an intuitive least squares error between the input image and the reconstruction of the input image from the filtered images.

Dunn et al. [8] strive to give Gabor filters that produce step signatures (discontinuity) in the filtered image so that texture boundaries of two distinct textures can be detected. The method involves an exhaustive search to find the center frequency, and image-segmentation error is minimized using measured output statistics and a Rician statistical model. Weldon et al. [9] consider the issue of designing a single Gabor filter to segment two textures. They establish a method for simultaneously computing output power of a large number of candidate Gabor filters for a particular texture. The measure of output power is then used to select the best Gabor filter that maximizes the ratio of output powers for the two filtered images. Weldon et al. [10] further present a method for the design of a single Gabor filter for multiple texture segmentation. In the method, Rician statistics of filtered textures at two different Gabor-filter envelope scales are used to generate probability density estimates for each filtered image. The Rician statistics associated with Gabor-filter output must be estimated over a range of Gabor-filter center frequencies at a rather intuitively selected Gaussian scale. The filter design is established by selecting the filter that provides the lowest image-segmentation error.

An explicit methodology has not been suggested for simultaneously selecting the best Gabor-filter parameters of frequency, orientation and scale. This paper considers the issue of designing a single Gabor filter for multi-texture segmentation using a systematic optimization algorithm. The selection objective for a best Gabor filter is based on the *Maxmin* principle that maximizes the minimum ratio of output responses of any two distinct textures. The *Maxmin* processing result makes the output responses between different textures well separated. Therefore, a simple thresholding scheme can be directly employed to partition the input image into differently textured regions. Since the Gabor transform is a nonlinear complex

function, we propose a stochastic optimization procedure based on the simulated annealing (SA) algorithm for effectively determining the Gabor-filter parameters. The SA algorithm has been demonstrated that it has the capability of escaping from the local optima. In order to improve the efficiency of the SA algorithm, the proposed optimization method incorporates the Hooke-Jeeves pattern search (PS) to the SA algorithm as the move generation mechanism. The proposed method can be viewed as an SA with a PS-based move strategy as well as being viewed as a modified PS with stochastic transition. The new proposed stochastic method, named SA/PS, allows the Gabor-filter parameters to be simultaneously determined.

This paper is organized as follows : Section 2 describes the Gabor filtering scheme, defines the output energy of filtered images, and discusses the *Maxmin* principle for the selection objective. Section 3 presents the proposed SA/PS optimization algorithm for determining the best Gabor-filter parameters. Section 4 demonstrates the experimental results for a variety of artificial and real textures. The paper is concluded in Section 5.

2. GABOR FILTER AND FILTER DESIGN

We begin with a brief overview of Gabor filters. The 1-D Gabor function was first defined by Gabor [23], and later extended to 2-D by Daugman [16]. A 2-D Gabor filter is an oriented complex sinusoidal grating modulated by a 2-D Gaussian function, which is given by

$$G_{\sigma,\,\phi,\theta}(x,y) = g_{\sigma}(x,y)\cdot \exp[2\pi j\phi(x\cos\theta + y\sin\theta)] \qquad (1)$$

where

$$g_\sigma = \frac{1}{2\pi\sigma^2}\exp[-(x^2+y^2)/2\sigma^2]$$

The frequency of the span-limited sinusoidal grating is given by ϕ and its orientation is specified as θ. $g_\sigma(x,y)$ is the Gaussian function with scale parameter σ. The parameters of a Gabor filter are therefore given by the frequency ϕ, the orientation θ and the scale σ. Note that we need only to consider θ in the interval $[0°, 180°]$. Symmetry makes the other directions redundant. This sets up the range constraint for θ in the SA/PS algorithm described in the next section.

The Gabor filter $G_{\sigma,\phi,\theta}(x,y)$ forms complex valued function. Decomposing $G_{\sigma,\phi,\theta}(x,y)$ into real and imaginary parts gives

$$G_{\sigma,\phi,\theta}(x,y) = R_{\sigma,\phi,\theta}(x,y) + jI_{\sigma,\phi,\theta}(x,y) \tag{2}$$

where

$$R_{\sigma,\phi,\theta}(x,y) = g_\sigma(x,y)\cdot\cos[2\pi\phi(x\cos\theta + y\sin\theta)]$$

$$I_{\sigma,\phi,\theta}(x,y) = g_\sigma(x,y)\cdot\sin[2\pi\phi(x\cos\theta + y\sin\theta)]$$

Gabor-filtered output of an image $f(x,y)$ is obtained by the convolution of the image with the Gabor function $G_{\sigma,\phi,\theta}(x,y)$. Given a neighborhood window of size $W\times W$ for W=2k+1, the discrete convolutions of $f(x,y)$ with respective real and imaginary components of $G_{\sigma,\phi,\theta}(x,y)$ are

$$C_R(x,y|\sigma,\phi,\theta) = \sum_{\lambda=-k}^{k}\sum_{m=-k}^{k} f(x+\lambda, y+m)\cdot R_{\sigma,\phi,\theta}(\lambda,m) \tag{3.a}$$

and

$$C_I(x,y|\sigma,\phi,\theta) = \sum_{\lambda=-k}^{k} \sum_{m=-k}^{k} f(x+\lambda, y+m) \cdot I_{\sigma,\phi,\theta}(\lambda, m) \qquad (3.b)$$

Define the energy $E(x,y|\sigma,\phi,\theta)$ at (x,y) as

$$E(x,y|\sigma,\phi,\theta) = C_R^2(x,y|\sigma,\phi,\theta) + C_I^2(x,y|\sigma,\phi,\theta) \qquad (4)$$

In the conventional Gabor-filter design approaches, the best filter parameters are generally selected so that the corresponding energy is a maximum for each specific texture. Here we consider the design of a single Gabor filter to segment multiple textures based on a *Maxmin* principle. Since our segmentation scheme is a supervised one, let us assume that an input image I consists of *N* different textural classes C_i, $i = 1,2,\Lambda\ ,N$. Let $I(C_i)$, a subimage of I with the size equal to the neighborhood window $W \times W$, be a training sample of the class C_i. Denote $E_i(x,y|\sigma,\phi,\theta)$ by the energy of texture class C_i, as defined in eq. (4), for a given filter parameter set (σ,ϕ,θ). The selection objective for the optimal Gabor-filter parameters is based on the *Maxmin* principle that maximizes the minimum energy ratio of any two distinct textures C_i and C_j, $i \neq j$. Therefore, the optimal filter parameter values $(\sigma^*,\phi^*,\theta^*)$ are selected by

$$(\sigma^*,\phi^*,\theta^*) = \arg[\underset{(\sigma,\phi,\theta)}{Maxmin}\{\frac{E_i(x,y|\sigma,\phi,\theta)}{E_j(x,y|\sigma,\phi,\theta)} \,|\, i = 1,2,\Lambda\ ,N-1; j = i+1, i+2,\Lambda\ ,N\}] \qquad (5)$$

For a bipartite textured image containing two texture classes C_1 and C_2, the above *Maxmin* selection objective is simply equal to maximizing the ratio of $E_1(x,y|\sigma,\phi,\theta)$ to $E_2(x,y|\sigma,\phi,\theta)$, i.e.,

$$\underset{(\sigma,\phi,\theta)}{Max}\ \{\frac{E_1(x,y|\sigma,\phi,\theta)}{E_2(x,y|\sigma,\phi,\theta)}\}$$

Let

$$E^* = \underset{(\sigma,\phi,\theta)}{Maxmin}\{\frac{E_i(x,y|\sigma,\phi,\theta)}{E_j(x,y|\sigma,\phi,\theta)} | \, i = 1,2,\Lambda\ ,N-1; j = i+1, i+2,\Lambda\ ,N\} \tag{6}$$

For an image containing *N* different textures, the *Maxmin* principle of eq. (6) results in energy magnitude in decreasing order as the texture class number increases, i.e.,

$$E_1(x,y|\sigma^*,\phi^*,\theta^*) > E_2(x,y|\sigma^*,\phi^*,\theta^*) > \Lambda\ > E_N(x,y|\sigma^*,\phi^*,\theta^*)$$

Also, the energy ratio of any two texture classes C_i and C_j, $i \neq j$, is at least E^*. A large value of E^* will result in better discrimination between any two distinct textures. By observing the energy distribution in the 1-D histogram of a multi-textured image, we can employ a simple multi-threshold scheme [24, 25] to select *N*-1 proper energy thresholds $T_1, T_2, \Lambda\ , T_{N-1}$. A pixel (x, y) in the textured image is assigned to texture class C_i if

$$T_{i-1} < E(x,y|\sigma^*,\phi^*,\theta^*) \leq T_i\ ,\ \ i = 1,2,\Lambda\ ,N$$

where $T_0 = 0$ and $T_N = \infty$. Therefore, the complicated texture segmentation problem in 2-D domain is converted to a simple thresholding problem in 1-D domain.

3. THE SA/PS OPTIMIZATION ALGORITHM

In this section, we discuss the optimization algorithm that determines the best parameter values of a single Gabor filter for multi-texture segmentation. Based upon the previous discussion, the optimal model for multi-texture segmentation can be formulated as

$$Max\,min\{\frac{E_i(x,y|\sigma,\phi,\theta)}{E_j(x,y|\sigma,\phi,\theta)} | \, i = 1,2,\Lambda\ ,N-1\ ;\ \ j = i+1, i+2,\Lambda\ ,N\}$$

subject to

$$0° \le \theta < 180° \tag{7.a}$$

$$\phi_{\min} \le \phi \le \phi_{\max} \tag{7.b}$$

$$\sigma_{\min} \le \sigma \le \sigma_{\max} \tag{7.c}$$

The constraints (7.a), (7.b) and (7.c) specify the possible ranges for parameters θ, ϕ, and σ, respectively. $\phi_{\min}$ and $\phi_{\max}$ are the minimum and maximum values of ϕ. Likewise, $\sigma_{\min}$ and $\sigma_{\max}$ are the minimum and maximum values of σ. The constraints on ϕ and σ are not mandatory, but they will increase the search efficiency of the proposed optimization algorithm. In our experiments, the range of ϕ is between 1 and *W* (the width of the window), and no restriction is placed on σ.

The model formulated above is a nonlinear, constrained programming problem with multiple continuous variables σ, ϕ, and θ. The proposed optimization algorithm is an approach combining the SA algorithm and the PS algorithm. The SA algorithm is well-known for its capability of escaping from the local optima. However, it is not efficient with respect to the number of iterations. The PS algorithm has been widely used and considered to be efficient in the area of nonlinear programming [26], but it also more inclined to terminate on the local optima. The proposed SA/PS optimization algorithm embeds the PS into the SA algorithm as the move generation mechanism.

The simulated annealing (SA) algorithm [27, 28] is a stochastic search technique, which has been designed for guiding search procedure to escape from the trap of local optimality. The search procedure in the SA algorithm performs with respect to a transition probability which is determined by the control temperature and the change in objective function. The SA algorithm is a variation of neighborhood search, which can move "downhill" with respect to gains, replacing the current solution with

lower gains. By allowing a move to a worse solution in a controlled situation, the SA algorithm can escape from a local optimum and potentially find a more promising "uphill" path. Further, the "downhill" moves are carefully controlled by the temperature parameter. When the temperature is high, the probability of the "downhill" move is high. With a gradual decreasing of temperature, the probability of the "downhill" move becomes small. The general procedure of the SA algorithm for maximizing an objective $E(X)$ is described as follows :

Initialize the solution X, and the temperature Γ

While Γ is not frozen,

Do the following loop *M* times :

Begin

Pick a neighboring solution X' of X

by the move generation mechanism.

Let $\Delta E = E(X') - E(X)$

If $\Delta E > 0$ (uphill move), set $X' = X$

If $\Delta E \le 0$ (downhill move), then

set $X' = X$ with probability $e^{\Delta E / \Gamma}$.

End

Set $\Gamma = c \cdot \Gamma$ (lower temperature), $0 < c < 1$

Return X

To reduce the computational cost of the SA algorithm, the proposed method incorporates the PS algorithm into the SA as the move generation mechanism to expedite the search procedure. The Hooke-Jeeves pattern search [29] proceeds according to a series of exploratory moves and pattern moves. The exploratory

moves examine the local behavior of a function and seek to locate the direction of any steep hills that might be present. The pattern moves utilize the information generated in the exploration to step rapidly along the hills.

The exploration starts from an initial point using the specified step size in each coordinate direction. If the adjacent move is accepted according to the objective function, the step is considered successful. Otherwise, the step is retracted and replaced by a step in the opposite direction, which in turn is retained depending upon whether it succeeds or fails. When all coordinates (variables) have been investigated, the exploratory move is completed. If a pattern direction exists after the completed exploratory move, the search procedure proceeds to the pattern move. In the case of a successful pattern move, the line search along the pattern direction is conducted until there is no further acceptable move. The step sizes of the variables are adjusted if the vector of pattern direction is equal to zero, i.e. no move is accepted in the completed exploratory move with the current step sizes. The step sizes will be increased first until the user-specified limit is reached. If this fails, the step sizes will be decreased, and the exploratory moves are repeated.

An application of the SA approach needs first to define four basic components of the algorithm [28]. The four basic components in the proposed SA/PS optimization algorithm are stated as follows :

(1) Configuration : a legal configuration is one combination of variables, i.e., an $X = (\sigma, \phi, \theta)$ related to the Gabor-filter model.

(2) Move set : all X's obtained from the PS algorithm are elements of the move set. The moves in the SA/PS algorithm are determined by the PS algorithm which adjusts the step sizes and decides the search directions.

(3) Gain function : the objective function of gains $E(X)$ is defined as eq. (6), which is the energy ratio of any two textures based on a *Maxmin* principle.

(4) Cooling schedule : we carried out a simple geometric cooling schedule [30]. After a specified number of moves (*M*) are completed, the temperature (Γ) is replaced by the old temperature multiplied by a constant *c*, called the cooling ratio, for $0 < c < 1$. The process is judged to be frozen when the procedure has made *K* consecutive loops of *M* moves with no change in the current best solution. The geometric cooling schedule is proposed here to reduce the computational requirements. A number of other cooling schedules have been discussed by Van Laarhoven and Aarts [31].

We can now formally present the proposed SA/PS algorithm in a general form. The following symbols are the notation used in the algorithm. The digits shown in the parentheses following each symbol are the parameter values used in the experiments.

Notations

m	number of decision variables
X	$\{x_1, x_2, \Lambda\ , x_m\}$, a set of decision variables ($X = \{\sigma, \phi, \theta\}$)
$x_{\min}(j)$	the lower bound of x_j
$x_{\max}(j)$	the upper bound of x_j
X'	a neighboring solution of X
X^0	initial solution of X, $\{x_1^0, x_2^0, \Lambda\ , x_m^0\}$
S	vector of step sizes, $\{s_1, s_2, \Lambda\ , s_m\}$
S^0	vector of initial step sizes, $\{s_1^0, s_2^0, \Lambda\ , s_m^0\}$

	$(s_1^0 = 0.5,\ s_2^0 = 1.0,\ s_3^0 = 1.0)$
ΔX	vector of pattern direction, $\{\Delta x_1, \Delta x_2, \Lambda\ , \Delta x_m\}$
n_I	a counter for number of step size increment
I	maximum number of step size increment allowed $(I = 150)$
n_D	a counter for number of step size decrement
r_I	increasing rate of step sizes, $r_I > 1$ $(r_I = 1.5)$
r_D	decreasing rate of step sizes, $0 < r_D < 1$ $(r_D = 0.8)$
n_M	a counter for number of search points at a temperature level
M	specified number of search points at a temperature level
n_K	a counter for checking frozen state achieved
K	specified maximum number of n_K
c	cooling ratio, a constant, $0 < c < 1$ $(c = 0.95)$
Γ	control temperature
Γ^0	initial control temperature

The procedure

Step 1. (Initialize the search procedure)

(a) Get an initial solution X^0, an initial temperature Γ^0, and initial step sizes S^0.

(b) Set $X = X^0$, $\Gamma = \Gamma^0$, $S = S^0$

$n_K = 0$, $n_M = 0$, $n_I = 0$

improve = FALSE, *frozen = FALSE*

Step 2. (Exploratory move)

Set $\Delta X = 0$, $X' = X$.

For j=*1* to m

(a) Set $x'_j = x_j + s_j$, $\Delta E = E(X') - E(X)$

Let $\Delta E = -\infty$ if $x'_j > x_{\max}(j)$.

If $\Delta E > 0$ (uphill move), then

set $X' = X$, $\Delta x_j = s_j$, *improve* = *TRUE*.

If $\Delta E \le 0$ (downhill move), then

set $X' = X$, $\Delta x_j = s_j$, *improve* = *TRUE* with probability $e^{\Delta E / \Gamma}$.

(b) Perform sub-procedure CHECK.

If *frozen* = *TRUE*, go to step 5.

(c) If $\Delta x_j \neq 0$, return to (a). Otherwise,

Set $x'_j = x_j - s_j$, $\Delta E = E(X') - E(X)$.

Let $\Delta E = -\infty$ if $x'_j < x_{\min}(j)$.

If $\Delta E > 0$ (uphill move), then

set $X = X'$, $\Delta x_j = -s_j$, *improve* = *TRUE*.

If $\Delta E \le 0$ (downhill move), then

set $X = X'$, $\Delta x_j = -s_j$, *improve* = *TRUE* with probability $e^{\Delta E / \Gamma}$.

(d) Perform sub-procedure CHECK

If *frozen* = *TRUE*, go to step 5.

Step 3. (Check pattern direction found and adjust step size)

(a) If $\Delta x_j \neq 0$, go to step 4.

(b) If $n_I \le I$, then

set $n_I = n_I + 1$,

$S = (r_I)^{n_I} \cdot S^0$ (increase step sizes),

go to step 2.

If $n_I > I$, then

set $n_D = n_D + 1$,

$S = (r_D)^{n_D} \cdot S^0$ (decrease step sizes),

go to step 2.

Step 4. (Pattern move)

(a) Set $X' = X + \Delta X$, $\Delta E = E(X') - E(X)$.

Let $\Delta E = -\infty$ if X' violates the constraints.

If $\Delta E > 0$ (uphill move), then

set $X = X'$, *improve* = *TRUE*.

If $\Delta E \le 0$ (downhill move), then

set $X = X'$, *improve* = *TRUE* with probability $e^{\Delta E / \Gamma}$.

(b) Perform sub-procedure CHECK

If *frozen* = *TRUE*, go to step 5.

(c) If $X' = X$, return to (a) (continue pattern move).

Otherwise, return to step 2 with *X*.

Step 5. (Termination)

Deliver *X* and terminate the search.

Sub-procedure CHECK (Check improvement and lower control temperature) :

Step 1 Set $n_M = n_M + 1$.

If $n_M < M$, then return.

Step 2 (Check improvement in *M* moves)

(a) Set $n_M = 0$.

If *improve* = *TRUE*, then set $n_K = 0$.

Otherwise, set $n_K = n_K + 1$.

(b) If $n_K = K$ (frozen state achieved), set *frozen* = *TRUE*.

Otherwise, set *frozen* = *FALSE*.

Step 3. (Lower control temperature)

Set $\Gamma = c \cdot \Gamma$, $0 < c < 1$.

Step 4. Return.

The proposed SA/PS algorithm has several desirable characteristics, including capability of escaping from the local optima, ease of implementation algorithmically, robustness for dealing with complicated nonlinear problems, and no restrictive assumptions about objective function, constraint set and parameter set.

4. EXPERIMENTS AND DISCUSSION

4.1. Experimental results

In this section we present the experimental results for evaluating the validity of the proposed Gabor-filter design algorithm for multi-texture segmentation. The algorithm is tested on a number of artificial and real textures containing bipartite, tripartite and quadripartite regions. All input images are 512×480 pixels wide with 8-bit gray levels. Only the inner 400×400 pixels are used as effective image regions for energy measurements so that the effect of image boundaries can be eliminated.

The size of the neighborhood windows is selected to be 65×65 pixels. During the training process, a subimage of size 65×65 pixels for each texture class, as shown in Figure 1(a), is arbitrarily selected to determine the best filter parameters. Figure 1(a) shows an artificial texture containing bipartite regions, one with horizontal line pattern, and the other with diagonal line pattern. Figure 2(a) presents a real bipartite textured image containing two different textile fabrics. Figure 3(a) shows another real bipartite textured image containing two sandpapers of different grains. Table 1(a) summarizes the trained energy values and the filter parameter values determined by the SA/PS algorithm for the bipartite textured images in Figures 1(a), 2(a) and 3(a). Figures 1(b), 2(b) and 3(b) show the segmentation results as binary images for Figures 1(a), 2(a) and 3(a), respectively. Figures 1(c), 2(c) and 3(c) depict the corresponding energy histograms for the textured images in Figures 1(a), 2(a) and 3(a). From each of the energy histograms, it can be seen that a bipartite textured image results in two groups (peaks) in the energy distribution, and a threshold can be easily identified at the valley of the distribution to segment the filtered image.

For three-texture segmentation, Figures 4(a) and 5(a) present tripartite textured images of real textile fabrics and sandpapers, respectively. The segmentation results are represented by trilevel images as seen in Figures 4(b) and 5(b). The corresponding energy histograms and the selected thresholds are shown in Figures 4(c) and 5(c). It shows three obvious peaks in each energy histogram.

For four-texture segmentation, Figure 6(a) demonstrates a quadripartite textured image containing four different textile fabrics. Figure 7(a) shows another quadripartite textured image containing four different machined surfaces (milled and shaped specimens with specific surface roughness). The segmentation results of the

quadripartite textured images are presented in Figures 6(b) and 7(b). The energy histograms for Figures 6(a) and 7(a) are shows in Figures 6(c) and 7(c), respectively. They show four groups in the distribution, each corresponding to one class of a specific texture in the image. The trained energy values and filter parameter values determined by the SA/PS algorithm for the tripartite textured images in Figures 4(a) and 5(a), and the quadripartite textured images in Figures 6(a) and 7(a) are summarized in Tables 1(b) and 1(c), respectively.

Observing from Figures 1 through Figure 7, we can see that a single Gabor filter derived from the SA/PS algorithm can effectively convert the textured image into a simple energy histogram that groups the energy distribution according to the number of texture classes in the image. More elaborate classification techniques such as Bayes, clustering and neural networks can be employed, but a simple threshold has directly demonstrated the effectiveness of the proposed method. The bipartite textured images yield very good segmentation results in terms of presence of noise and smoothness of texture boundaries. Minor noise is generated for tripartite and quadripartite textured images. The classification errors are typically confined to the vicinity of the texture boundary due to the untrained texture patterns around the boundary. Away from the boundary, the filter-output energy is highly discriminating. We can concluded that the proposed single Gabor-filter design approach is robust for segmenting images containing a few textures. Its performance will degrade gracefully as the number of texture classes in an image increases.

There are a few factors that may affect the effectiveness of the segmentation results, including the neighborhood window size, the order of texture classes appeared in the objective function, and the parameter setup in the SA/PS algorithm. They are

separately discussed in the following subsections.

4.2. Effect of window sizes

The selection of a proper neighborhood window size must be large enough to contain the local, periodic, spatial arrangement of intensity for all texture classes in question. Too small a window size causes insufficient representation of texture information, whereas too large a window size increases the computational burden. In this experiment, we vary the window size from 65×65, 53×53, 41×41 to 25×25 pixels to study the impact of window size on segmentation. Figure 8(a) shows a bipartite textured image containing two textile fabrics. Figures 8(b), 8(c), 8(d) and 8(e) present the segmentation results from window sizes 65×65, 53×53, 41×41 and 25×25, respectively. Window sizes 65×65 and 53×53 generate similar segmentation results. However, as the window sizes are reduced to 41×41 and 25×25, the segmentation results in zigzag boundary at the transit of two texture patterns. Away from the texture boundaries, the filter-output energy is still highly discriminating. Minor noise appears for the over-reduced window of 25×25. Based on the segmentation results in Figure 8, it reveals that a larger neighborhood window size causes the filter-output energy to be less sensitive to window-position perturbation, and reducing output variation.

4.3. Order of texture classes in the objective function

The objective function of eq.(6) maximizes the minimum energy ratio of any two textures C_i and C_j, $i \neq j$. Let E_i and E_j be the measured energies for

respective texture classes C_i and C_j. If the energy ratio in eq.(6) is defined by E_i/E_j, then we expect that the resulting energy values will have $E_i > E_j$. Alternately, if the energy ratio is given by E_j/E_i, then we expect that $E_j > E_i$. The question now is whether the order of texture classes present in the objective function (i.e., the position of E_i in the numerator or denominator with respect to any other E_j) can affect the effectiveness of the SA/PS algorithm.

In this experiment, the bipartite textured image as shown in Figure 2(a) is used for evaluation. In the figure, the inner fabric, denoted by E_1, has finer texture, and the outer fabric, denoted by E_2, has coarser texture. The SA/PS search results from individual objective function $Max\{E_1/E_2\}$ and $Max\{E_2/E_1\}$ are summarized in Tables 2(a) and 2(b), respectively. We have conducted 10 runs of the SA/PS procedure with different random number sequences for each specific objective function. It can be seen from both Tables 2(a) and 2(b) that all resulting energy values in the numerator are larger than those in the denominator. Taking the fine texture in the numerator results in large distance between E_1 and E_2, and the SA/PS algorithm converges to similar and successful parameter values for 9 out of 10 runs, as marked by " * " in Table 2(a). However, taking the coarse texture in the numerator only results in 5 successful searches out of a total of 10 runs, as seen in Table 2(b). The similar results are also observed for the tripartite textured image shown in Figure 4(a). Therefor, it can be concluded that the textures C_1, C_2, Λ, C_N present in the objective function of eq.(6) should be ordered such that C_i has finer texture than C_j if $i < j$. This forms the energy ratio E_i/E_j for any two textures C_i and C_j in the objective function.

4.4. SA/PS parameter setup

To evaluate the impact of the starting point $(\sigma^0, \phi^0, \theta^0)$ on the SA/PS solution, we arbitrarily select two starting points $(\sigma^0, \phi^0, \theta^0)$ = (1.0, 1.0, 1.0) and (3.0, 10.0, 50.0) to test the bipartite textured image shown in Figure 2(a). Five runs are carried out for each starting point with different random number sequences. Table 3 presents the trained energy values and the resulting parameter values. It shows that the SA/PS algorithm converges to similar parameter values for both starting points. The trained energy values for the fine texture (E_1) and the coarse texture (E_2) are around 15 and 0.1, respectively, using either starting point.

In the SA/PS algorithm, the solution quality is generally determined by the initial control temperature Γ^0, and the specified maximum number *K* for the frozen state of temperature. Generally, the larger the parameter values Γ^0 and *K* we use in the SA/PS algorithm, the better the resulting solution is close to the global optimum. The cost for using large values of Γ^0 and *K* is the increment of computational time. In this experimental study, we have examined two levels of Γ^0 at 100 and 10,000, and two levels of *K* with *K*=10 and *K*=20 for the three bipartite textured images shown in Figures 1(a), 2(a) and 3(a), and the two tripartite textured images in Figures 4(a) and 5(a). We have found that temperature parameter Γ^0=100, and frozen-state parameter *K*=10 are sufficient to generate successful Gabor parameters for texture segmentation. As a rule of thumb, the larger the number of texture classes presents

in an image for segmentation, the larger the values of Γ^0 and K should be used in the SA/PS algorithm.

5. CONCLUSION

In this paper, a systematic optimization algorithm has been presented for the design of a single Gabor filter for multi-texture segmentation. The proposed optimization algorithm is a stochastic search technique based on the simulated annealing (SA) procedure. It embeds the pattern search (PS) into the SA procedure as the move generation mechanism to accelerate the search. The selection objective for a best Gabor filter is based on the *Maxmin* principle that maximizes the minmum energy ratio of any two distinct textures. This objective makes the energy responses between different texture classes well separated. Therefore, a simple thresholding scheme can be directly applied to partition an input image into different textured regions. This approach converts a complicated texture segmentation problem in 2-D domain to a simple thresholding problem in 1-D domain.

The experiments on bipartite, tripartite and quadripartite textured images have demonstrated the effectiveness of the proposed method. While a large set of Gabor filters could be used in the segmentation, the experimental results show that the proposed method is sufficiently powerful to achieve multi-texture segmentation using only a single Gabor filter. Because of the generality and flexibility of the proposed SA/PS algorithm, the proposed objective function can be easily modified to different objectives without any revision of the search procedure. For instance, the Gaussian function $g_\sigma(x, y)$ used in the Gabor function can be asymmetric, i.e.,

$$g(x,y) = \exp\left[\frac{1}{2}(x/\sigma_x) + \frac{1}{2}(y/\sigma_y)\right]$$

where σ_x and σ_y determine the scales of the Gaussian along the respective axes. This results in four Gabor-filter parameters (σ_x, σ_y, ϕ, θ). In the most general case, $g(x,y)$ can be any reasonable window function. The proposed SA/PS algorithm can be directly applied to any window function or other types of filters.

The performance of the proposed method degrades gracefully with increasing number of texture classes in an image. When an image contains numerous textures, this leads to on going research on the design of a best set of multiple Gabor filters to segment multiple textures using the SA/PS algorithm.

REFERENCES

1. P. P. Raghu and B. Yegnanarayana, Segmentation of Gabor-filtered textures using deterministic relaxation, IEEE Trans. Image Processing 5, 1625-1636 (1996).

2. T. R. Reed and H. Wechsler, Segmentation of textured images and Gestalt organization using spatial/spatial-frequency representations, IEEE Trans. Pattern Analysis Mach. Intell. 12, 1-12 (1990).

3. A. C. Bovik, M. Clark and W. S. Geisler, Multichannel texture analysis using localized spatial filters, IEEE Trans. Pattern Analysis Mach. Intell. 12, 55-73 (1990).

4. A. K. Jain and F. Farrokhnia, Unsupervised texture segmentation using Gabor filters, Pattern Recognition 24, 1167-1186 (1991).

5. A. Teuner, O. Pichler and B. J. Hosticka, Unsupervised texture segmentation of images using tuned matched Gabor filters, IEEE Trans. Image Processing 4, 863-870 (1995).

6. M. M. Van Hulle and T. Tollenaere, A modular artificial neural network for texture processing, Neural Networks 6, 7-32 (1993).

7. M. Clark and A. C. Bovik, Texture segmentation using Gabor modulation/ demodulation, Pattern Recognition Letters 6, 261-267 (1987).

8. D. Dunn and W. E. Higgins, Optimal Gabor filters for texture segmentation, IEEE Trans. Image Processing 4, 947-964 (1995).

9. T. P. Weldon, W. E. Higgins and D. F. Dunn, Efficient Gabor filter design for texture segmentation, Pattern Recognition 29, 2005-2015 (1996).

10. T. P. Weldon, W. E. Higgins and D. F. Dunn, Gabor filter design for multiple texture segmentation, Optical Engineering 35, 2852-2863 (1996).

11. K. R. Namuduri, R. Mehrotra and N. Ranganathan, Efficient computation of Gabor filter based multiresolution responses, Pattern Recognition 27, 925-938 (1994).

12. A. K. Jain, N. K. Ratha and S. Lakshmanan, Object detection using Gabor filters, Pattern Recognition 30, 295-309 (1997).

13. D. P. Casasent, J. -S. Smokelin and A. Ye, Wavelet and Gabor transforms for detection, Optical Engineering 31, 1893-1898 (1992).

14. M. Porat and Y. Y. Zeevi, The generalized Gabor scheme of image representation in biological and machine vision, IEEE Trans. Pattern Analysis Mach. Intell. 10, 452-468 (1988).

15. Y. Hamamoto, S. Uchimura, M. Watanabe, T. Yasuda, Y. Mitani and S. Tomita, A Gabor filter-based method for recognizing handwritten numerals, Pattern Recognition 31, 395-400 (1998).

16. J. G. Daugman, Uncertainty relation for resolution in space, spatial-frequency, and orientation optimized by two-dimensional visual cortical filters, J. Opt. Soc. Amer. 2, 1160-1169 (1985).

17. R. L. De Valois, D. G. Albrecht and L. G. Thorell, Spatial-frequency selectivity of cells in macaque visual cortex, Vision Research 22, 545-559 (1982).

18. J. Beck, A. Sutter and R. Ivry, Spatial frequency channels and perceptual grouping in texture segregation, Comput. Vision, Graphics, Image Processing 37, 299-325 (1987).

19. D. Dunn, W. Higgins and J. Wakeley, Texture segmentation using 2-D Gabor elementary function, IEEE Trans. Pattern Analysis Mach. Intell. 16, 130-149 (1994).

20. A. C. Bovik, Analysis of multichannel narrow-band filters for image texture segmentation, IEEE Trans. Signal Processing 39, 2025-2043 (1991).

21. M. Clark and A. Bovik, Experiments in segmenting texton patterns using localized spatial filters, Pattern Recognition 22, 707-717 (1989).

22. M. Porat and Y. Y. Zeevi, Localized texture processing in vision : analysis and synthesis in the Gaborian space, IEEE Trans. Biomed. Eng. 36, 115-129 (1989).

23. D. Gabor, Theory of communication, J. Inst. Elec. Eng. 93, 429-457 (1946).

24. D. -M. Tsai and Y. -H Chen, A fast histogram-clustering approach for multi-level thresholding, Pattern Recognition Letters 13, 245-252 (1992).

25. D. -M. Tsai, A fast thresholding selection procedure for multimodal and unimodal histograms, Pattern Recognition Letters 16, 653-666 (1995).

26. J. P. Ignizio, Goal Programming and Extensions, Lexington Books, Boston, Mass. (1976).

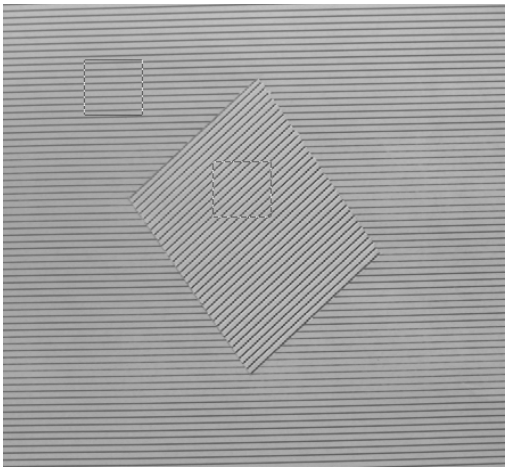
27. S. Kirkpatrick, C. D. Gelatt, Jr. and M. P. Vecchi, Optimization by simulated annealing, Science 220, 671-680 (1983).

28. R. A. Rutenbar, Simulated annealing algorithm : an overview, IEEE Circuits and Device Magazine 5, 19-26 (1989).

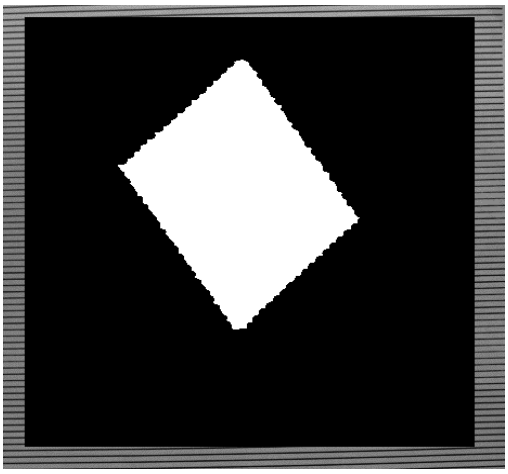
29. R. Hooke and T. A. Jeeves, A direct search solution of numerical and statistical problems, J. ACM 8, 212-219 (1961).

30. N. E. Collins, R. W. Egless and B. L. Golden, Simulated annealing-an annotated bibliography, American Journal of Mathematical and Management Science 8, 209-307 (1988).

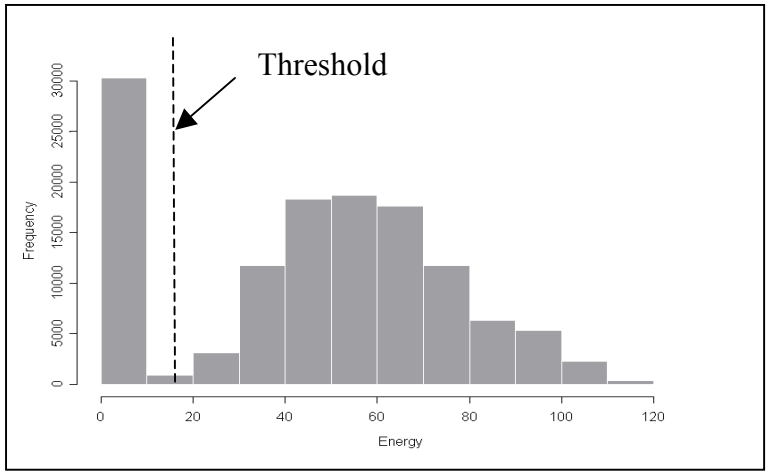
31. P. J. M. Van Laarhoven and E. H. L. Aarts, Simulated Annealing : Theory and Applications, Reidel Dordrecht, Holland (1987).



(a)

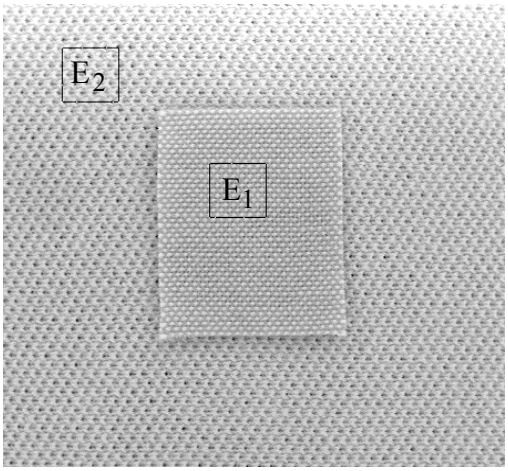


(b)

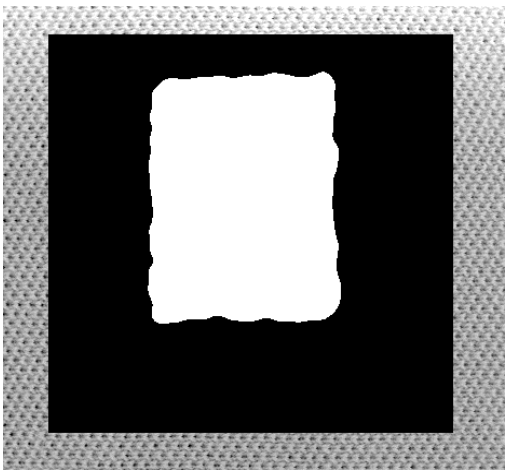


(c)

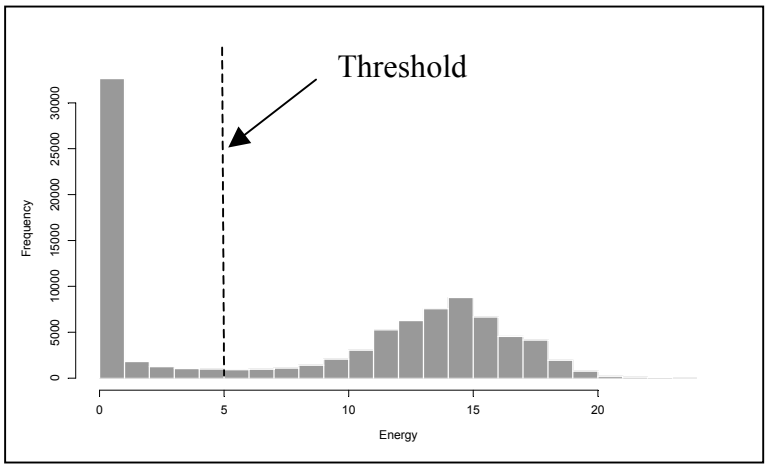
Figure 1. An artificial bipartite textured image of line patterns : (a) the original image ; the square frames shown in the image are the subimages used for training ; (b) the segmentation result shown as a binary image ; (c) the energy histogram.



(a)

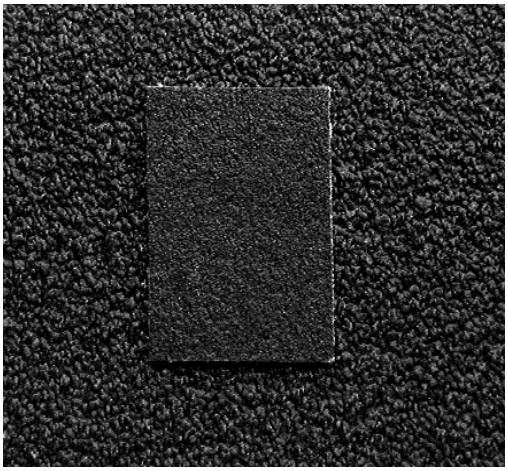


(b)

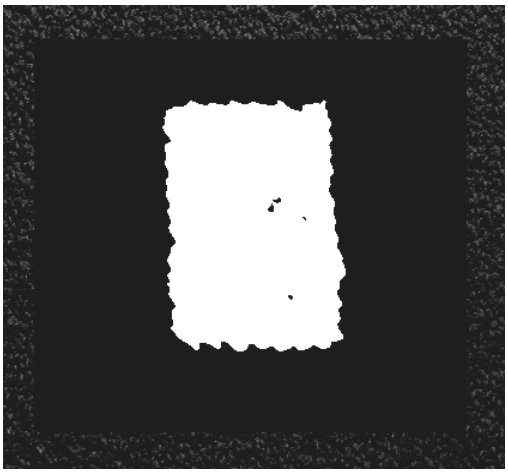


(c)

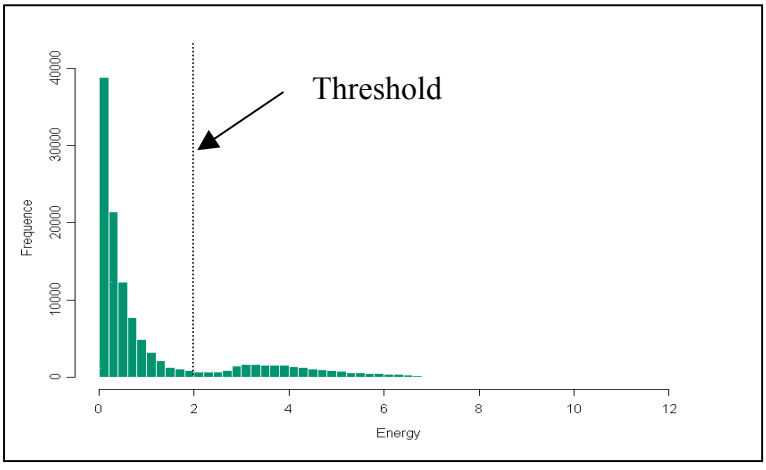
Figure 2. A real bipartite textured image of textite fabrics : (a) the original image ; the square frames shown in the image are the subimages used for training ; (b) the segmentation result ; (c) the energy histogram.



(a)

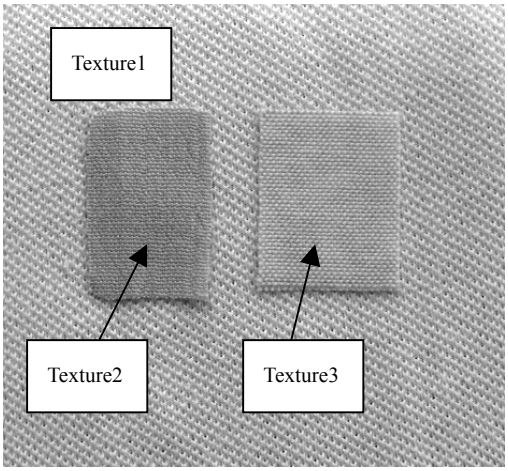


(b)

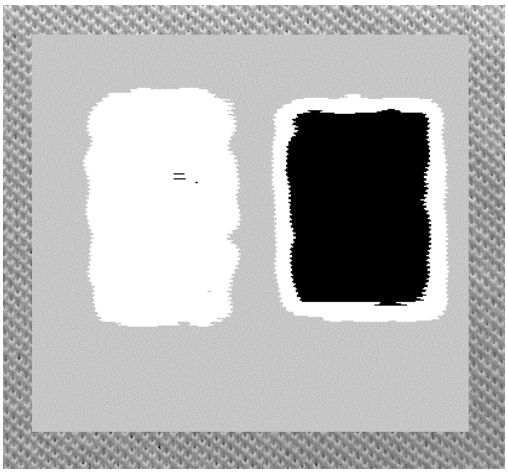


(c)

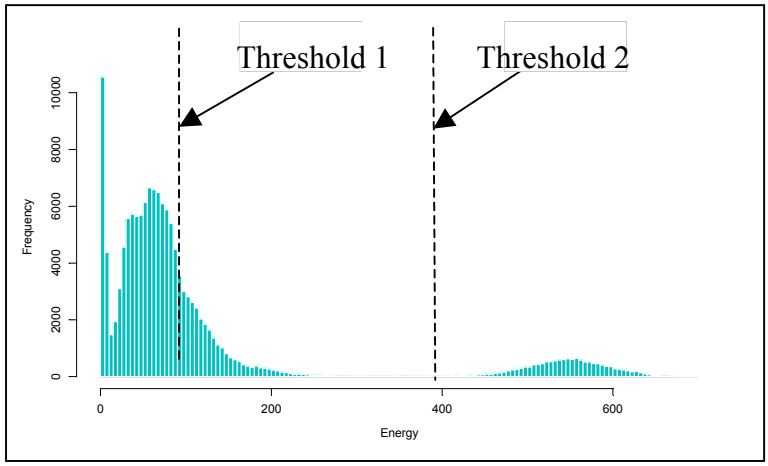
Figure 3. A real bipartite image of sandpapers : (a) the original image ; (b) the segmentation result ; (c) the energy histogram.



(a)

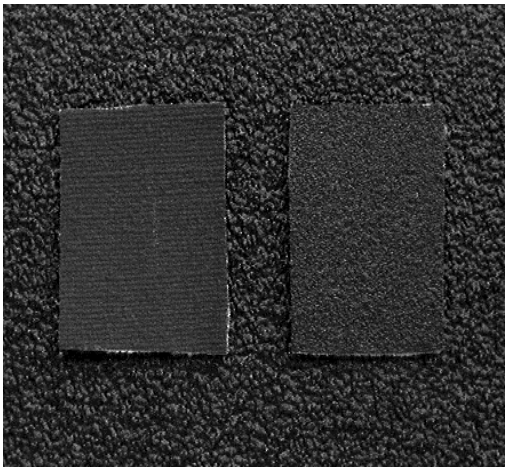


(b)

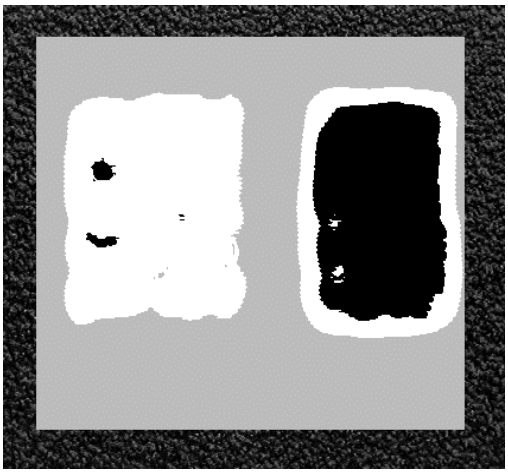


(c)

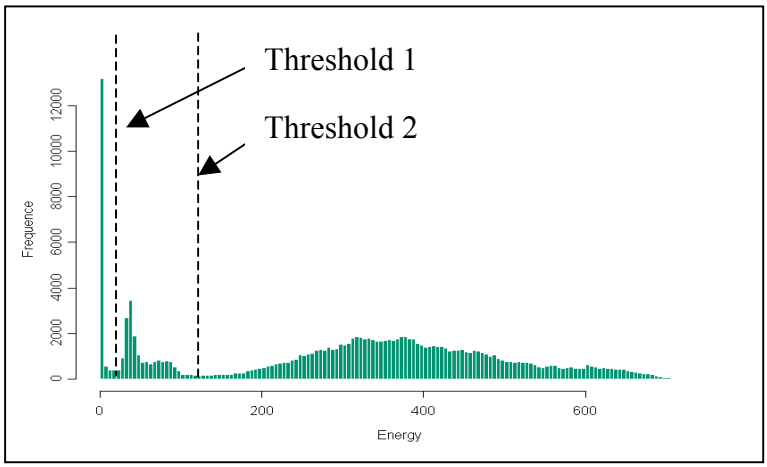
Figure 4. A tripartite textured image containing three different textile fabrics : (a) the original image ; (b) the segmentation result ; (c) the energy histogram.



(a)

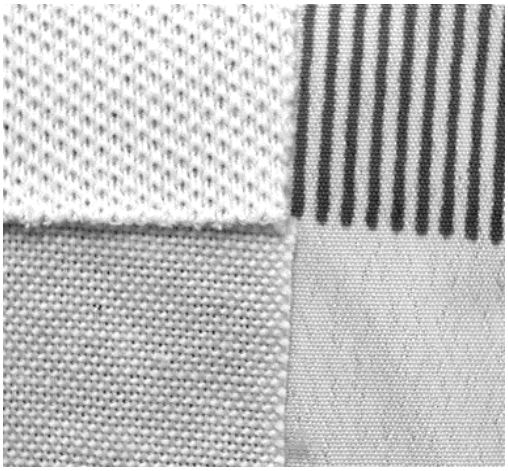


(b)

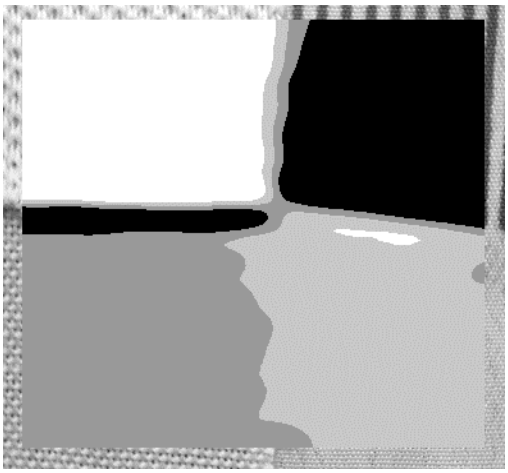


(c)

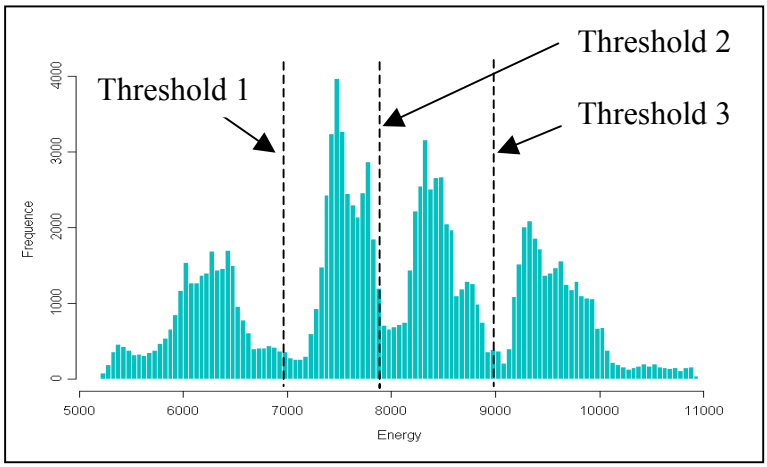
Figure 5. A tripartite textured image containing sandpapers of three different grains : (a) the original image ; (b) the segmentation result ; (c) the energy histogram.



(a)

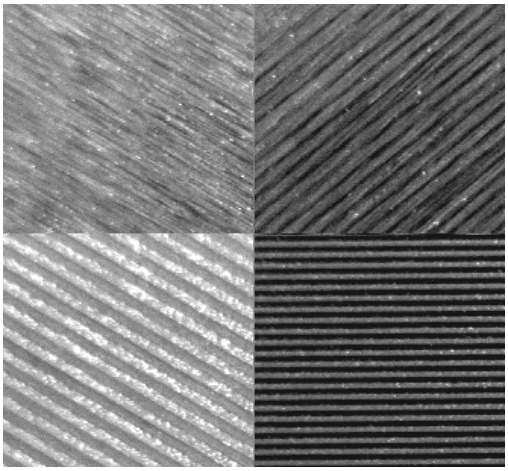


(b)

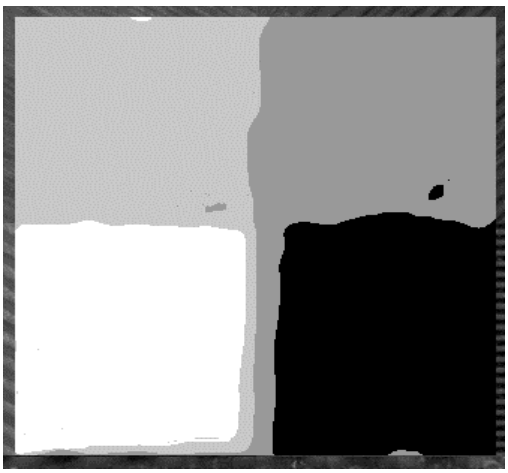


(c)

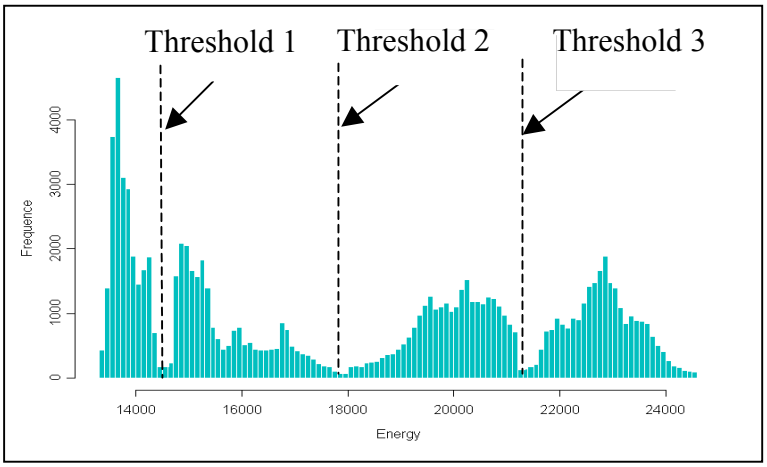
Figure 6. A quadripartite textured image containing four different textite fabrics : (a) the original image ; (b) the segmentation result ; (c) the energy histogram.



(a)

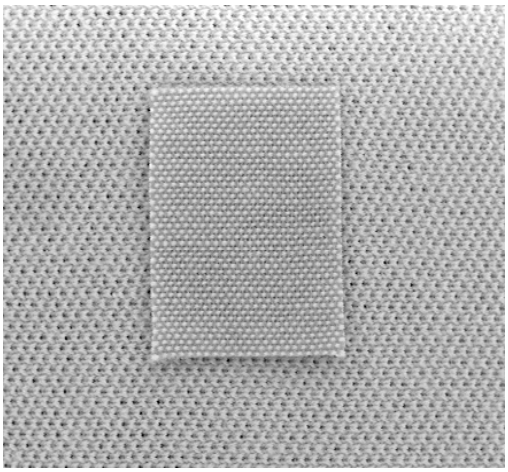


(b)

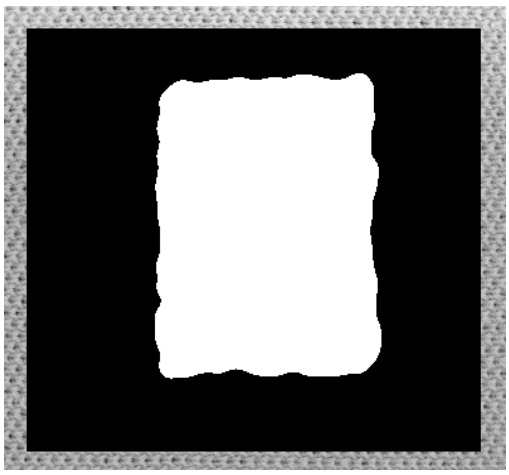


(c)

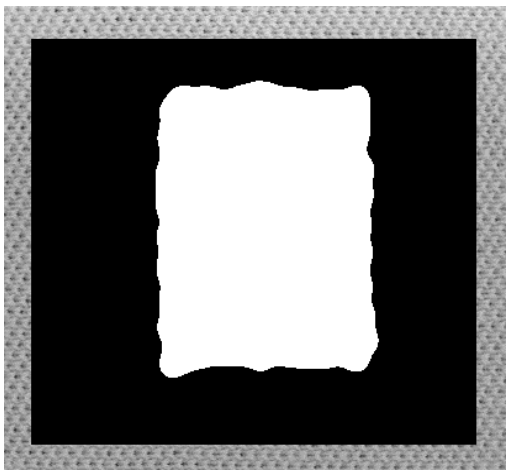
Figure 7. A guadpartite textured image containing four different machined surfaces : (a) the original image ; (b) the segmentation result ; (c) the energy histogram.



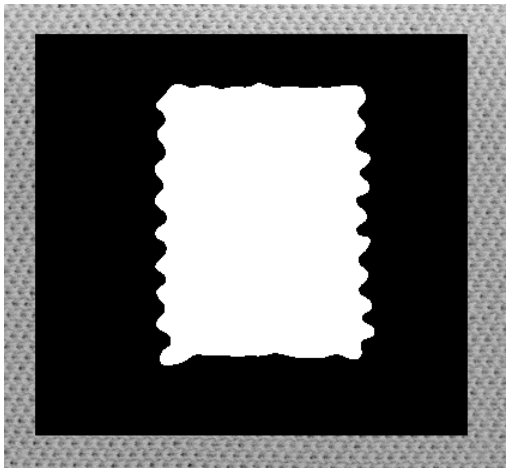
(a)



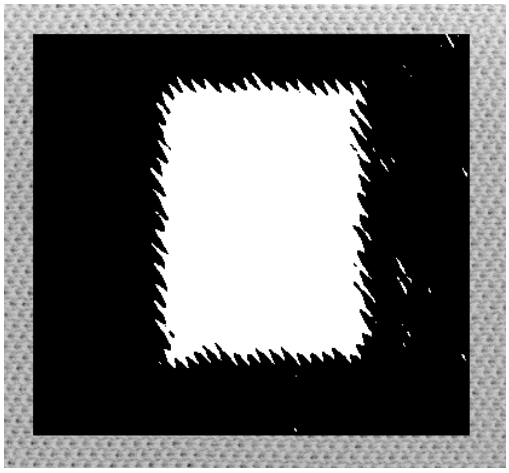
(b) 65×65



(c) 53×53



(d) 41×41



(e) 25×25

Figure 8. Segmentation with various windows sizes : (a) the test example of a bipartite fabric image ; (b) result from size 65 × 65 ; (c) result from size 53 ×53 ; (d) result from size 41×41 ; (e) result from size 25×25.

Table 1. The trained energy values and filter parameter values for : (a) bipartite textured images ; (b) tripartite textured images ; (c) quadripartite textured images.

(a)

Textured image	Energy		Filter parameters (σ,ϕ,θ)
	E_1	E_2	
Line patterns (Fig. 1(a))	70.82	0.10	(2.57, 6.99, 177.56)
Textile fabrics (Fig. 2(a))	18.67	0.10	(6.62, 28.24, 107.60)
Sandpaper (Fig. 3(a))	15.57	0.10	(9.18, 61.01, 179.32)

(b)

Textured image	Energy			Filter parameters (σ,ϕ,θ)
	E_1	E_2	E_3	
Textile fabrics (Fig. 4(a))	19.91	8.10	0.00	(9.23, 3.00, 5.00)
Sandpaper (Fig. 5(a))	21.26	10.27	0.00	(10.45, 6.01, 2.89)

(c)

Textured image	Energy				Filter parameters (σ,ϕ,θ)
	E_1	E_2	E_3	E_4	
Textile fabrics (Fig. 6(a))	9685.3	8634.8	7505.6	6365.4	(12.0, 1.0, 1.0)
Machined surfaces (Fig. 7(a))	22752.1	19952.7	15159.5	13900.2	(7.5, 19.0, 90.0)

Table 2. The SA/PS search results for the bipartite textured image shown in Figure 2(a) using the objective function : (a) $\max\{E_1/E_2\}$; (b) $\max\{E_2/E_1\}$. (The run marked by " * " represents a successful training.)

(a) Objective function : $\max\{E_1/E_2\}$

Run number	Energy		Filter parameters (σ,ϕ,θ)
	E_1 (Fine texture)	E_2 (Coarse texture)	
*1	25.66	0.10	(7.27, 17.47, 56.81)
*2	19.41	0.10	(7.15, 17.99, 56.04)
*3	18.93	0.10	(6.97, 18.67, 57.40)
*4	16.62	0.10	(7.38, 18.28, 56.48)
*5	19.15	0.10	(7.32, 18.29, 57.79)
*6	22.18	0.10	(7.43, 17.52, 55.61)
*7	23.68	0.10	(7.04, 18.99, 60.00)
*8	19.61	0.10	(7.88, 19.23, 57.00)
9	5.19	0.10	(1.78, 70.04, 83.99)
*10	25.20	0.10	(7.36, 18.84, 56.99)

(b) Objective function : $\max\{E_2/E_1\}$

Run number	Energy		Filter parameters (σ,ϕ,θ)
	E_1 (Fine texture)	E_2 (Coarse texture)	
*1	0.10	18.67	(6.61, 28.23, 107.59)
2	1.05	4.13	(2.15, 57.19, 106.00)
*3	0.10	24.16	(6.33, 29.21, 106.00)
*4	0.10	19.18	(6.26, 29.12, 106.25)
5	0.12	3.24	(1.15, 15.80, 118.69)
*6	0.10	23.16	(6.33, 29.12, 106.00)
7	2.32	4.25	(2.15, 57.19, 106.00)
8	0.08	3.26	(1.07, 41.62, 127.88)
*9	0.10	21.21	(6.61, 28.01, 105.94)
10	0.84	4.20	(2.15, 57.19, 106.00)

Table 3. The resulting energy and filter parameter values of the SA/PS algorithm using different starting points for the bipartite textured image in Figure 2(a).

Starting point $(\sigma^0, \phi^0, \theta^0)$	Run	Energy		Filter parameters (σ, ϕ, θ)
		E_1	E_2	
(1.0, 1.0, 1.0)	1	19.46	0.10	(10.64, 12.00, 1.04)
	2	17.79	0.10	(11.25, 12.14, 1.43)
	3	13.32	0.10	(10.82, 12.14, 1.83)
	4	13.47	0.10	(10.64, 12.00, 1.04)
	5	14.65	0.10	(11.15, 11.00, 1.00)
(3.0, 10.0, 50.0)	1	14.68	0.10	(12.00, 12.49, 2.80)
	2	15.92	0.10	(12.00, 12.53, 2.82)
	3	14.73	0.10	(11.87, 12.00, 2.70)
	4	17.83	0.10	(11.62, 12.00, 2.27)
	5	16.47	0.10	(12.00, 12.45, 2.96)