International Journal of Pattern Recognition and Artificial Intelligence
Vol. 16, No. 1 (2002) 1–25

A NEW METHOD FOR EXTRACTING PRIMITIVES OF REGULAR TEXTURES BASED ON WAVELET TRANSFORM*

KUEN-LONG LEE and LING-HWEI CHEN†

Department of Computer and Information Science, National Chiao Tung University, 1001 Ta Hsueh Rd., Hsinchu, Taiwan 30050, R.O.C.
†*lhchen@cc.nctu.edu.tw*

In this paper, a new method for extracting primitives of a regular texture via wavelet transform is provided. Each primitive is restricted to be a parallelogram. The main work of the proposed method is to extract the two displacement vectors that form the primitive of a regular texture. A contrast-based criterion is used to select appropriate subimages obtained from wavelet transform for detecting displacement vectors. An edge thresholding method is then performed on the subimages selected to locate edges. Based on these edges, Hough transform is applied to extract the displacement vectors. The proposed method is quite efficient and can get more accurate results for displacement vectors when comparing to traditional co-occurrence matrix based methods. Synthesized textures are provided to show the effectiveness of the proposed method.

Keywords: Regular texture; texture primitive; texture analysis and synthesis; displacement vector.

1. Introduction

Texture has long been an important topic in image processing (see Refs. 1, 2, 4, 6, 7, 10–13, 15, 17, 19–21, 26, 28, 29 and 31). Generally speaking, textures can be classified into two major categories, i.e. regular and irregular. To be more specific, irregular textures like cloud or grass, cannot be constructed by regularly arranged patterns; while regular textures like brick wall, are composed of structurally repeated similar patterns which can be modeled by texture primitives with two displacement vectors along which the texture primitives are formed. To represent irregular textures, many statistical approaches have been proposed, they use parameters to measure texture content in terms of smoothness, coarseness and regularity. For regular textures, however, it is believed that they are most efficiently represented by structural approaches through texture primitives and displacement vectors. Therefore, an efficient and effective method for extracting the texture primitives and displacement vectors is needed for regular textures.

*This research was supported in part by the National Science Council of R.O.C. under contract NSC-87-2213-E-009-060.
†Author for correspondence.

In the past, some approaches[14,16,23,30] have been proposed to extract the primitives and displacement vectors of regular textures. Co-occurrence matrix (CM) based methods have been widely used in the analysis of texture structure. Kim and Park[14] used the projection information computed from the co-occurrence matrix to extract the spatial arrangement of a regular texture. In this method, a texture primitive is assumed to be a parallelogram and the texture primitives are arranged along two periodicity angles of the parallelogram. To reduce the computational load, these two periodicity angles are first determined by projection information. For each angle, co-occurrence matrices for several distances are computed. Then, a chi-square test is performed on these co-occurrence matrices to obtain the corresponding repeating distance along that angle. Starovoitov *et al.*[27] adopted CM-based method as well. Several CM features are examined and joint features are used to extract the texture primitives. Basically, CM-based methods are computationally expensive. Thus, to lessen the computational load, coarse quantization of the repeating angles and distances is usually needed. Zucker and Terzopoulos[32] used only four angles, i.e. 0°, 45°, 90°, 135°. Kim and Park[14] used a step size of 10° from 0° to 170° to search for the angle. These will make the extracted displacement vectors less accurate.

There are also non-CM-based methods for texture primitives extraction. Matsuyama *et al.*[24] extracted the texture periodicity by detecting the peaks on the Fourier transform of a texture image. However, as the peaks in Fourier spectrum are not prominent, it is not easy to locate them. Lin *et al.*[18] used autocorrelation function to extract the periodicity of a regular texture. The autocorrelation function of a regular texture is first computed, a peak finding algorithm based on 2D Gaussian smoothing is then applied to detect the peaks on the autocorrelation function. Based on the locations of these peaks, the displacement vectors and texture primitives are then extracted by a generalized Hough transform.[3] However, computation of autocorrelation function is time-consuming. Besides, the displacement vectors extracted from this method do not match those observed by human perception.

To lessen the computational load and increase the accuracy of texture primitive extraction, in this paper, we will propose a texture periodicity extraction method based on wavelet transform.[9,22] Based on the definition of Conner and Harlow,[8] that the texture primitive forming a texture image is assumed to be a parallelgram that can be described by two displacement vectors, the proposed method applies the wavelet transform to a texture image to decompose it into subimages of different frequencies and orientations, i.e. low–low (LL), low–high (LH), high–low (HL) and high–high (HH) subimages. A contrast-based criterion is then used to select the appropriate subimages for detecting displacement vectors. The two displacement vectors of the texture primitive will be extracted from the LH and HL subimages selected. For each subimage, the edge points are located by a thresholding method. Based on these edge points, the Hough transform is then used to extract the angle and length of the displacement vector in each subimage. As the algorithm

for determining the displacement vectors only involves simple computation, it is quite efficient in terms of computational loading. In addition, a finer quantization step compared to traditional CM-based methods is used in the Hough trasnform procedure for displacement vector extraction, the displacement vectors extracted are more accurate. A texture is then synthesized by repeating the parallelogram along the displacement vectors, and the extracted primitives do match with those observed by human perception.

The rest of the paper is organized as follows. In Sec. 2, the proposed method is described in detail. The experimental results and discussion are presented in Sec. 3. Finally, in Sec. 4, we give a conclusion.

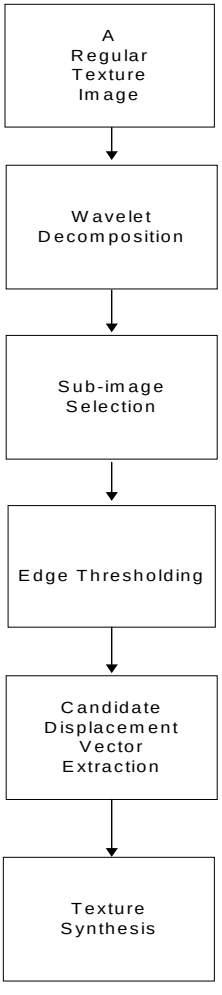


Fig. 1. The system block of the proposed method.

2. The Proposed Method

The system block of the proposed method is shown in Fig. 1. A regular texture image is first decomposed into subimages of different frequencies and directions via wavelet transform. A subimage selection method based on contrast is then used to select appropriate subimages for detecting displacement vectors. An edge thresholding method is then applied on the selected subimages to locate the edge pixels. Hough transform is performed to extract the candidates of the two displacement vectors from LH and HL subimages based on the edge pixels located, respectively. Finally, the extracted texture primitives are used to synthesize the texture. In the following, the proposed method will be described in detail.

2.1. *Wavelet decomposition*

In this paper, the pyramid-structured wavelet transform is adopted to decompose the texture image. Via the transform, the texture image is decomposed into four subimages, i.e. low–low (LL), low–high (LH), high–low (HL) and high–high (HH). The decomposition can be repeated on the LL subimage to produce the four subimages of the next level. For a certain decomposition level, the LH subimage and HL subimage actually correspond to the near-horizontal edge subimage and the near-vertical edge subimage, respectively. Figure 2 shows a two-level decomposition. Figure 3 shows the original image of D20, which is from Brodatz album,[5] and the result image after applying wavelet transform to four levels.

2.2. *Subimage selection*

As mentioned above, the two displacement vectors that form the texture primitive can be extracted from the LH or HL subimages. However, as the responses of the edges for some subimages are not prominent enough, not every subimage is a suitable candidate for calculating the displacement vector. Therefore, a subimage

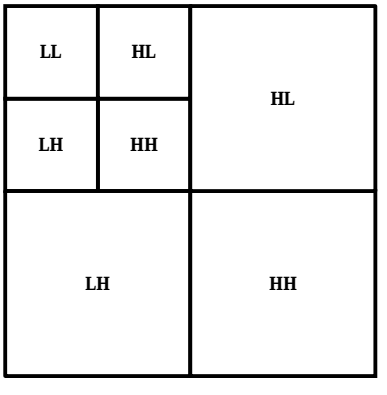
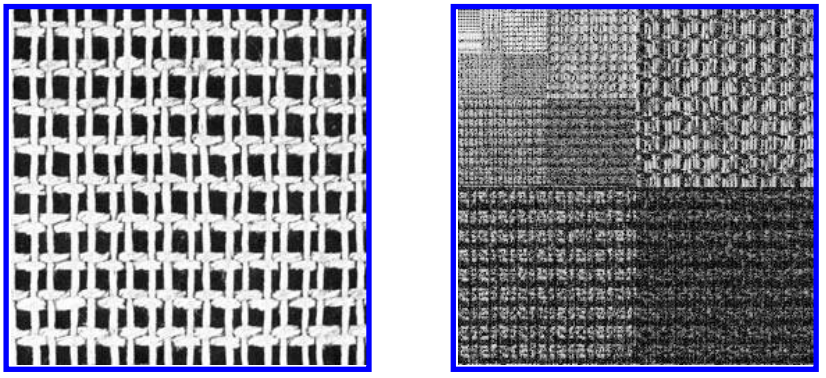


Fig. 2. A two-level decomposition representation of wavelet transform.



(a) (b)

Fig. 3. The original image and its wavelet transform. (a) The original D20 image. (b) The result image of applying wavelet transform on (a).

selection criterion is needed to select the appropriate subimages for displacement vector detection.

According to the fact that a subimage with more prominent edges will have greater contrast, we define the subimage contrast as the energy variance of a subimage and use it to select the appropriate subimages for displacement vector detection. For a subimage SI of size $M \times M$, the contrast is defined as

$$SI_c = \sum_{i=1}^{M} \sum_{j=1}^{M} (SI(i,j) - SI_m)^2 \tag{1}$$

where SI_m is the energy mean of SI, and is defined as

$$SI_m = \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} SI(i,j)\,. \tag{2}$$

For all LH or HL subimages, the subimage with the maximum contrast, and those with contrast above a minimum threshold, T_c, are selected as the representative for displacement vector detection. For a certain textured image with a strip primitive (see Fig. 4 as an example), there will be no subimage selected as the representative for LH or HL subimages. In this case, only one displacement vector is detected. When there are no subimages with contrasts above the minimum threshold, the textured image is rejected by the method for primitive detection. This case will be discussed in the experimental results section. The detail algorithm for subimage selection is shown in the following.

Algorithm* 1. *Subimage Selection

Input. The LH subimages, SLH^i, and the HL subimages, SHL^i, $i = 1, \ldots, L$, L is the maximum decomposition level.

Output. The representative LH subimage, SLH^{i^*}, and the representative HL subimage SHL^{j^*}, selected for displacement vector detection.

Step 1. (Select the representative subimage for LH subimages)

For $i = 1$ to L

Compute SLH_c^i by Eq. (1).

End for

$i^* = \arg\max_i SLH_c^i$

if $SLH_c^{i^*} < T_c$ then no subimage is selected as SLH^{i^*}.

Step 2. (Select the representative subimage for HL subimages)

For $j = 1$ to L

Compute SHL_c^j by Eq. (1).

End for

$j^* = \arg\max_j SLH_c^j$

if $SHL_c^{j^*} < T_c$ then no subimage is selected as SHL^{j^*}.

Step 3.

If no SLH^{i^*} and no SHL^{j^*} are selected then reject the input image for primitive detection.

***End of algorithm* 1**.

Figure 5 shows the representative LH and HL subimages selected for Fig. 3(b).

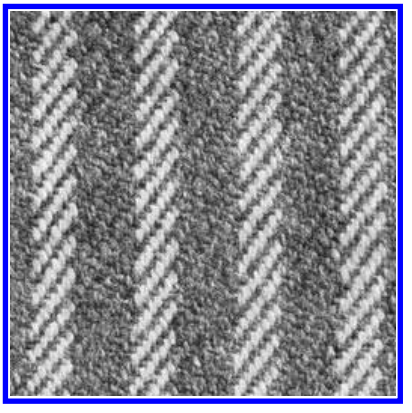


Fig. 4. A textured image (D11) with strip primitives.

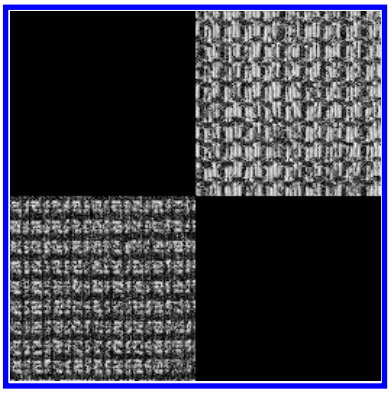


Fig. 5. The representative LH and HL subimages selected for Fig. 3(b).

2.3. *Edge thresholding*

As mentioned above, for any decomposition level, the LH subimage and HL subimage actually correspond to the near-horizontal edge subimage and the near-vertical edge subimage, respectively. It is observed in each subimage that the texture primitives are arranged along the direction of those lines formed by edge pixels. Thus, if we are able to locate these edge pixels, then the located edges can be used as a good base for extracting the two displacement vectors along which a texture primitive is repeated.

Assume that the transform coefficients in SLH^{i^*} or SHL^{j^*} subimage range from 0 to $L_{\max}$. Based on the fact that the edge pixels are of high values, we can locate these edge pixels through a threshold value l with the aggregated energy from $L_{\max}$ to l greater than $t\%$ of the total energy. In order to locate edges accurately, before thresholding, a smoothing and noise removing process is applied by using two types of windows that resemble the shapes of edges, i.e. a wide window for a LH subimage and a long window for an HL subimage. The thinning procedure for binary images presented in Ref. 25 is then applied to reduce the width of located edges to one pixel. Basically, the thinning procedure consists of iterations of two basic steps applied to the border points of the binary image, where a border point is any pixel with value 1 and having at least one 8-neighbor valued 0. The 8-neighborhood definition used in the thinning procedure is shown as Fig. 6.

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

Fig. 6. The neighborhood definition used by the thinning algorithm.

Then, the number of nonzero neighbors of p_1, $N(p_1)$, is defined as follows:

$$N(p_1) = \sum_{i=2}^{9} p_i \, .$$

And $S(p_1)$ is defined as the number of 0–1 transitions in the ordered sequence of $p_2, p_3, \ldots, p_8, p_9, p_2$. Two conditions are defined based on the above definitions.

***Condition* 1**:

(a) $2 \leq N(p_1) \leq 6$;
(b) $S(p_1) = 1$;
(c) $p_2 \cdot p_4 \cdot p_6 = 0$;
(d) $p_4 \cdot p_6 \cdot p_8 = 0$;

***Condition* 2**:

(a) $2 \leq N(p_1) \leq 6$;
(b) $S(p_1) = 1$;
(c) $p_2 \cdot p_4 \cdot p_8 = 0$;
(d) $p_2 \cdot p_6 \cdot p_8 = 0$;

Border pixels which satisfy these two conditions are then flagged for deletion. The two basic steps are then applied iteratively until no further points are deleted, at which time the thinning algorithm terminates with the thinned binary image. Now the binary image thinning algorithm and edge thresholding algorithm are presented as follows.

Algorithm* 2. *Binary Image Thinning

Input. A binary image $B(x, y), x = 1, \ldots, N, y = 1, \ldots, N$ with $B(x, y) = 0$ or 1.

Output. The thinned $B(x, y)$.

Repeat
// Flag border pixels satisfying condition 1 and delete flagged pixels
For each $B(x, y)$ which is a border point do
If $B(x, y)$ satisfies condition 1 then
Flag $B(x, y)$ as deleted.
End if
End for
Delete all $B(x, y)$ marked for deletion.
// Flag border pixels satisfying condition 2 and delete flagged pixels

For each $B(x, y)$ which is a border point do

If $B(x, y)$ satisfies condition 2 then

Flag $B(x, y)$ as deleted.

End if

End for

Delete all $B(x, y)$ marked for deletion.

Until no border pixels are marked for deletion.

***End of algorithm* 2**.

Algorithm* 3. *Edge thresholding

Input. A subimage (SLH^{i^*} or SHL^{j^*}) $I(x, y), x = 1, \ldots, N, y = 1, \ldots, N$ with $0 \leq I(x, y) \leq L_{\max}$.

Output. Edge pixels of the input subimage.

Step 1. (Smoothing-noise removing)

For each pixel $I(x, y)$

Create a new pixel, $I'(x, y)$ as

$$I'(x, y) = \frac{1}{hs} \sum_{(a,b) \in W x,y} I(a, b)\,,$$

where $W_{x,y}$ is the $h \times s$ window centered at (x, y) with $s > h$ if the input image is a LH subimage; $h > s$ if the input image is an HL subimage.

Step 2. Evaluate the histogram of $H(i), i = 1, \ldots, L_{\max}$ of I'.

Step 3. Compute the energy E of I' as:

$$E = \sum_{x=1}^{N} \sum_{y=1}^{N} I'(x, y)\,.$$

Step 4. Let CE_j be the cumulative energy of histogram $H(i)$ from j to $L_{\max}$ and be evaluated as follows:

$$CE_j = \sum_{i=j}^{L_{\max}} H(i)\,.$$

Determine the largest threshold l which satisfies the following criterion:

$$l = \max\{j | CE_j/E > t\%\}\,.$$

Step 5. Use l to threshold $I'(x, y)$ as follows:

If $I'(x, y) < l$ then $I'(x, y) = 0$.

Else $I'(x, y) = 1$.

End if

Step 6. Perform thinning on $I'(x, y)$.

***End of algorithm* 3**.

After applying Algorithm 3, nonzero $I'(x, y)$ s correspond to edge pixels in the subimage. In our experiment, the window size of s and h are set as 3×7 for LH subimages, and 7×3 for HL subimages for all levels of decomposition. Figure 7 shows the thresholding result of Fig. 5.

2.4. *Candidate displacement vector extraction*

In this step, based on the edge points located, the Hough transform[25] is used to extract the angles and lengths of two displacement vectors of the texture primitive from SLH^{i^*} and SHL^{j^*} subimages, respectively. Here, we use the following equation to represent a line:

$$x \cos\theta + y \sin\theta = \rho\,. \tag{3}$$

Through the Hough transform (the details can be found in Ref. 25), the angle θ^* associated with the maximum accumulating value plus 90° is considered as the angle of the candidate for one displacement vector of the texture primitive in that subimage. Then, we extract those ρ_s with the accumulating value of (θ^*, ρ_s) greater than

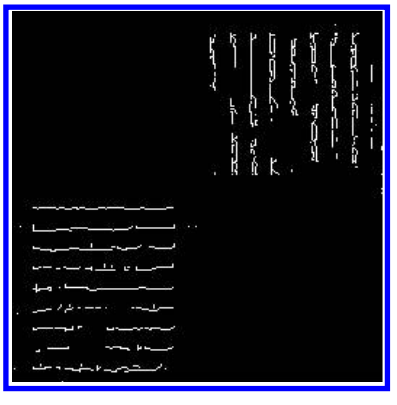


Fig. 7. The result image after applying edge thresholding to Fig. 5.

zero, and compute the distances among these ρ_s. Let the distance with maximum frequency be denoted as Pd^* and (θ^*, Pd^*) is called a parameter pair.

By conducting the Hough transform on SLH^{i^*} and SHL^{j^*} subimages respectively, two parameter pairs (θ_1^*, Pd_1^*) and (θ_2^*, Pd_2^*) can be extracted. A pictorial example is shown in Fig. 8. From Fig. 8, it can be easily shown that:

$$\angle DEF = \angle CDG = \angle CAB, \text{ and } \angle CAB = |\theta_1^* - \theta_2^*|\,,$$

$$Pd_1^* = D_2 \times \sin(\angle DEF) = D_2 \times \sin(|\theta_1^* - \theta_2^*|)\,.$$

Therefore, the repeating distances D_1 and D_2 along the directions of $\theta_1^* + \pi/2$ and $\theta_2^* + \pi/2$ can be derived as follows:

$$D_1 = \frac{Pd_2^*}{\sin(|\theta_1^* - \theta_2^*|)}\,,$$

$$D_2 = \frac{Pd_1^*}{\sin(|\theta_1^* - \theta_2^*|)}\,.$$

$(\theta_i^* + \pi/2, D_i)$, $i = 1, 2, -\pi < \theta_i^* < \pi$, are then used to form the displacement vectors of subimages. The Parameter Pair Calculation algorithm and Displacement Vectors Detection algorithm are presented as follows.

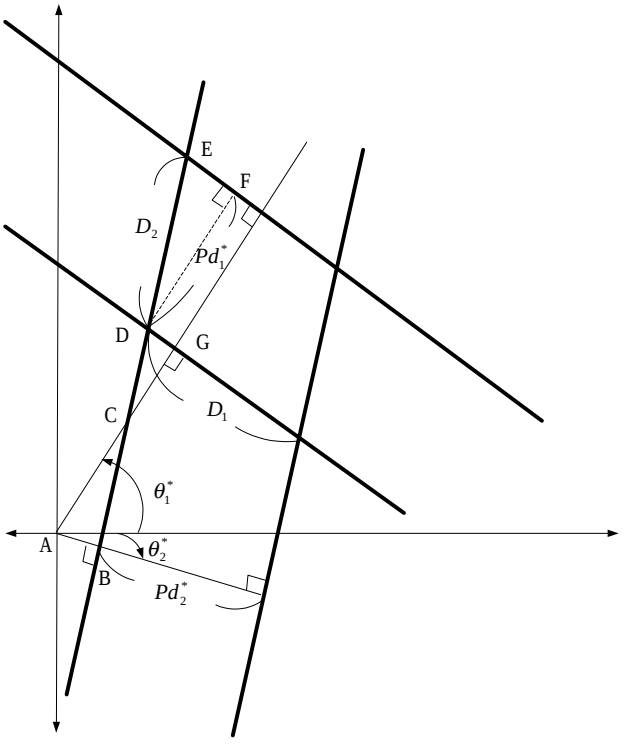


Fig. 8. A pictorial example for two parameter pairs extracted.

Algorithm* 4. *Parameter Pair Calculation

Input. The edge pixels of a subimage. Let the set of edge pixels be denoted as $E(x_s, y_s), s = 1, \ldots, S$. S is the number of edge pixels.

Output. The parameter pair (θ^*, Pd^*) of the subimage.

Step 1. Let $\theta_i = (i-1)^*\Delta\theta$, $i = 1, \ldots, M$, represent the quantized values of θ, and $\rho_j = (j-1)^*\Delta\rho$, $j = 1, \ldots, R$, denote the quantized values of ρ. $\Delta\theta$ and $\Delta\rho$ are the step size of θ and ρ, respectively. Let $AC(i, j)$ be defined as the accumulator corresponding to parameter (θ_i, ρ_j), and C_k as the accumulator of candidate length of the displacement vector.

For each edge point $E(x_s, y_s)$

For each θ_i

Compute the corresponding ρ with $\rho = x_s \cos\theta_i + y_s \sin\theta_i$.

Let $j = \arg\min_i |\rho_i - \rho|$.

$AC(i, j) = AC(i, j) + 1$.

End for

End for

Let $m = \arg\max_i(AC(i, j))$, and set $\theta^* = \theta_m$.

Step 2. For each ρ_i with $AC(m, i) > 0$

For all ρ_j with $AC(m, j) > 0$ and $j > i$

$c_{j-i} = c_{j-i} + 1$

End for

End for

$k = \arg\max_i c_i$.

$Pd^* = k \cdot \Delta\rho$.

***End of algorithm* 4.**

Algorithm* 5. *Displacement Vectors Detection

Input. The edge-thresholded images.

Output. The displacement vectors (θ_i^*, d_i^*), $i = 1, 2$ of the texture primitive.

Step 1. Perform parameter pair calculation algorithm (Algorithm 4) on LH and HL subimages respectively, to obtain two parameter pairs (θ_i^*, Pd_i^*), $i = 1, 2$.

Step 2. Calculate the repeating distances of D_i and θ_i^*, $i = 1, 2$ as follows:

$$D_1 = \frac{Pd_2^*}{\sin(|\theta_1^* - \theta_2^*|)},$$

$$D_2 = \frac{Pd_1^*}{\sin(|\theta_1^* - \theta_2^*|)}.$$

Step 3. Output $(\theta_i^* + \pi/2, D_i)$, $i = 1, 2$ as the displacement vectors of the texture primitive.

***End of algorithm* 5**.

In our experiments, the quantization steps of θ and ρ of Parameter Pair Calculation algorithm are both set as 1.

2.5. *Texture synthesis*

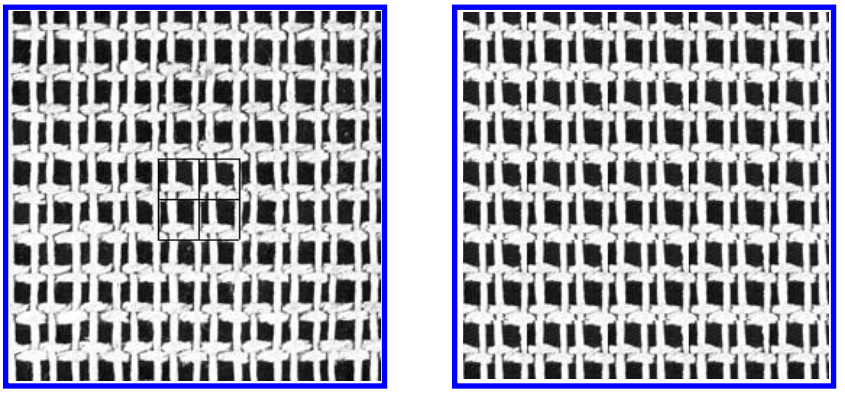
In order to show the effectiveness of the proposed method, the extracted primitive is used to synthesize a texture image, which will be compared with the original image. We synthesize the texture by determining a center point CP from which the extracted displacement vectors are used to reconstruct the texture. To determine CP, an AND operation is performed on the LH and HL subimages obtained after the edge thresholding step. Let the result be denoted as f. We then start searching for CP from the center of f and set the first object pixel encountered as CP.

A representative unit which is centered at CP and has the size of four texture primitives is then extracted. Figure 9 shows the synthesis process of D20 image. Figure 9(a) shows D20 with the representative unit marked by black line segments. Using the representative unit, the synthesis process starts from pasting the representative unit centered at CP. To paste the next unit, the next center point is computed, and the pasting procedure is repeated row by row in upward and downward directions respectively until the whole image is filled. The synthesized texture of D20 is shown in Fig. 9(b). The synthesis algorithm is described as follows:

Algorithm* 6. *Texture Synthesis

Input. The two displacement vectors ν_1 and ν_2 constituting a texture primitive.

Output. The synthesized texture image.



(a) (b)

Fig. 9. The synthesis result of D20 image. (a) D20 with the synthesis unit marked by black line segments. (b) Synthesized texture of D20.

Step 1. Let $I'_{LH}(x,y), I'_{HL}(x,y), x = 1,\ldots,N, y = 1,\ldots,N$ denote the thresholded LH image and HL image, respectively.

$f = AND(I'_{LH}, I'_{HL})$

Step 2. Search the object pixel o which is the nearest neighbor of

$(N/2, N/2)$ with $f(o) = 1$.

$CP = o$

Step 3. Take the parallelogram centered at CP with the size of four texture primitives as the representative unit P. The structure of P is shown in Fig. 10.

$MCP = CP$

Step 4. (Pasting rows upwardly)

c $= MCP$

While P centered at c is within the area of the original image

Begin

Paste P at c

c = c $- 2\nu_1$

end

c $= MCP + 2\nu_1$

While P centered at c is within the area of the original image

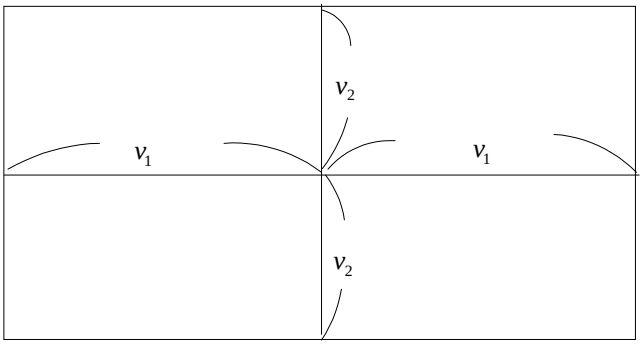


Fig. 10. The structure of a representative unit.

Begin

Paste P at c.

$\mathrm{c} = \mathrm{c} + 2\nu_1$

end

$MCP = MCP - 2\nu_2$

if P centered at MCP is within the area of the original image then go to step 4.

Step 5. (Pasting rows downwardly)

$MCP = CP + 2\nu_2$

$\mathrm{c} = MCP$

While P centered at c is within the area of the original image

Begin

Paste P at c.

$\mathrm{c} = \mathrm{c} - 2\nu_1$

end

$\mathrm{c} = MCP + 2\nu_1$

While P centered at c is within the area of the original image

Begin

Paste P at c.

$\mathrm{c} = \mathrm{c} + 2\nu_1$

end

$MCP = MCP + 2\nu_2$

if P centered at MCP is within the area of the original image then go to step 5.

***End of algorithm* 6**.

3. Experimental Results

Several regular texture images from Brodatz album are used to test the proposed method. All images used have 256×256 pixels and are gray-scale ones. Pyramid-structured wavelet transform with four-level decomposition using Daubechies 6-tap filter are used in all experiments. To demonstrate the effectiveness of the proposed method, textured images with the following characteristics are chosen in the experiments: (1) textured images with rectangular type of primitives, (2) textured images with primitives represented by diagonal displacement vectors, (3) textured images with primitives containing curvilinear objects, (4) textured images with strip primitives.

Figure 11 shows the extraction and synthesis results of applying the proposed method to a D1 texture image from Brodatz album in sequence. Figure 11(b) shows the wavelet-transformed image. As shown in Fig. 11(b), the LH and HL subimages of decomposition level two have greater contrast when compared to the LH and HL subimages of the other decomposition levels. Extracted vertical and horizontal edges of HL and LH subimages are shown in Fig. 11(c). T_c, the minimum threshold for selecting a subimage is an empirical value and set as 2460 in our experiments. It is shown that the LH and HL subimages of decomposition level two are correctly selected, and the corresponding edges are thresholded. The contrast values of all subimages of Fig. 11(b) are shown in Table 1. Figure 11(d) illustrates that the texture primitives extracted match with those observed by human perception. Figure 11(e) shows the synthesized texture image which is quite similar to Fig. 11(a). Figure 11(f) shows the experimental result of D1 by applying the method proposed in Ref. 18, the primitives extracted do not match with human perception.

Table 1. The contrast values of all subimages of D1.

Decomposition Level	LH	HL
1	2307.7	3177.1
2	3187.6	4141.4
3	2512.1	3120.8
4	1963.2	2219.8
$T_c = 2460$		

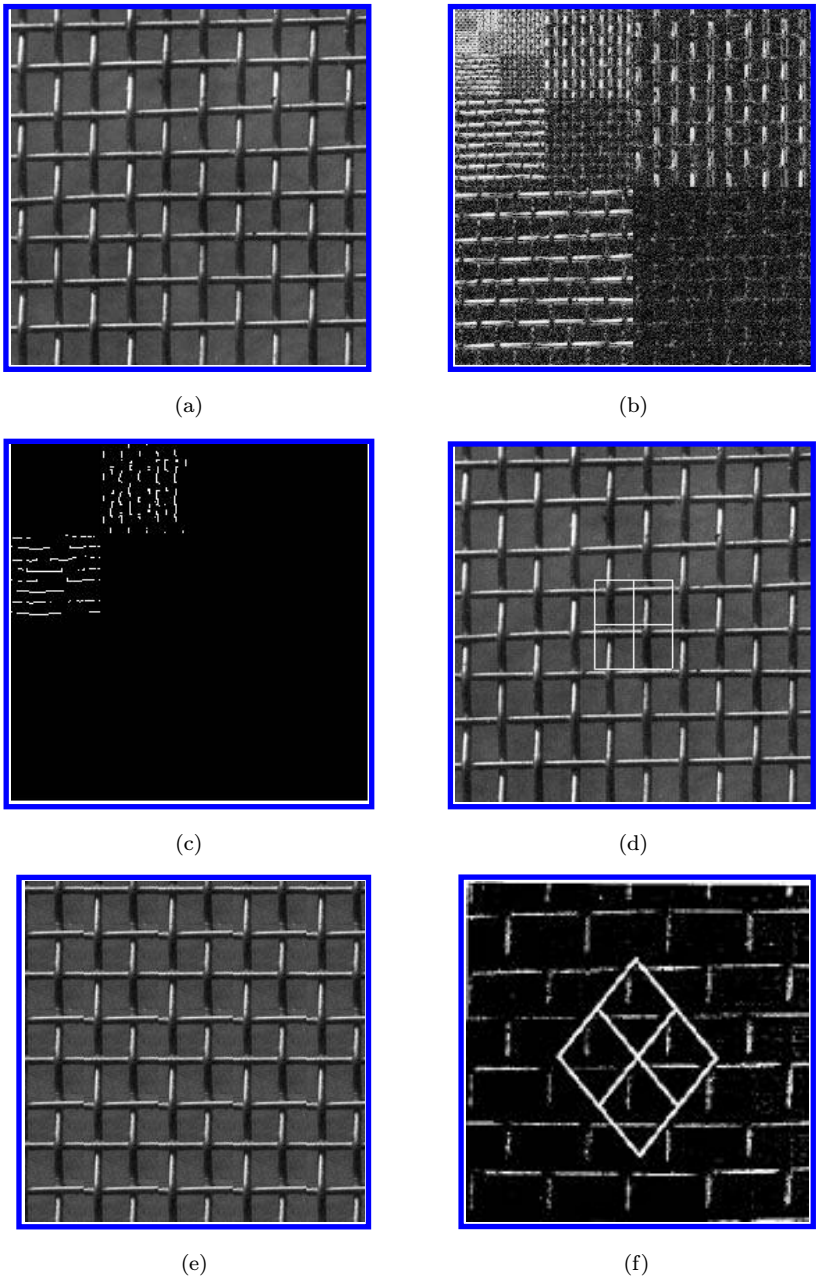


Fig. 11. The extracted primitives and synthesized image for a D1 texture image. (a) A D1 textured image from Brodatz album. (b) The result image of applying wavelet transform on (a). (c) The edge-thresholded result of the representative LH and HL subimages. (d) The original image with four primitives marked. (e) The synthesized image. (f) The primitives extracted by applying the method proposed in Ref. 18.

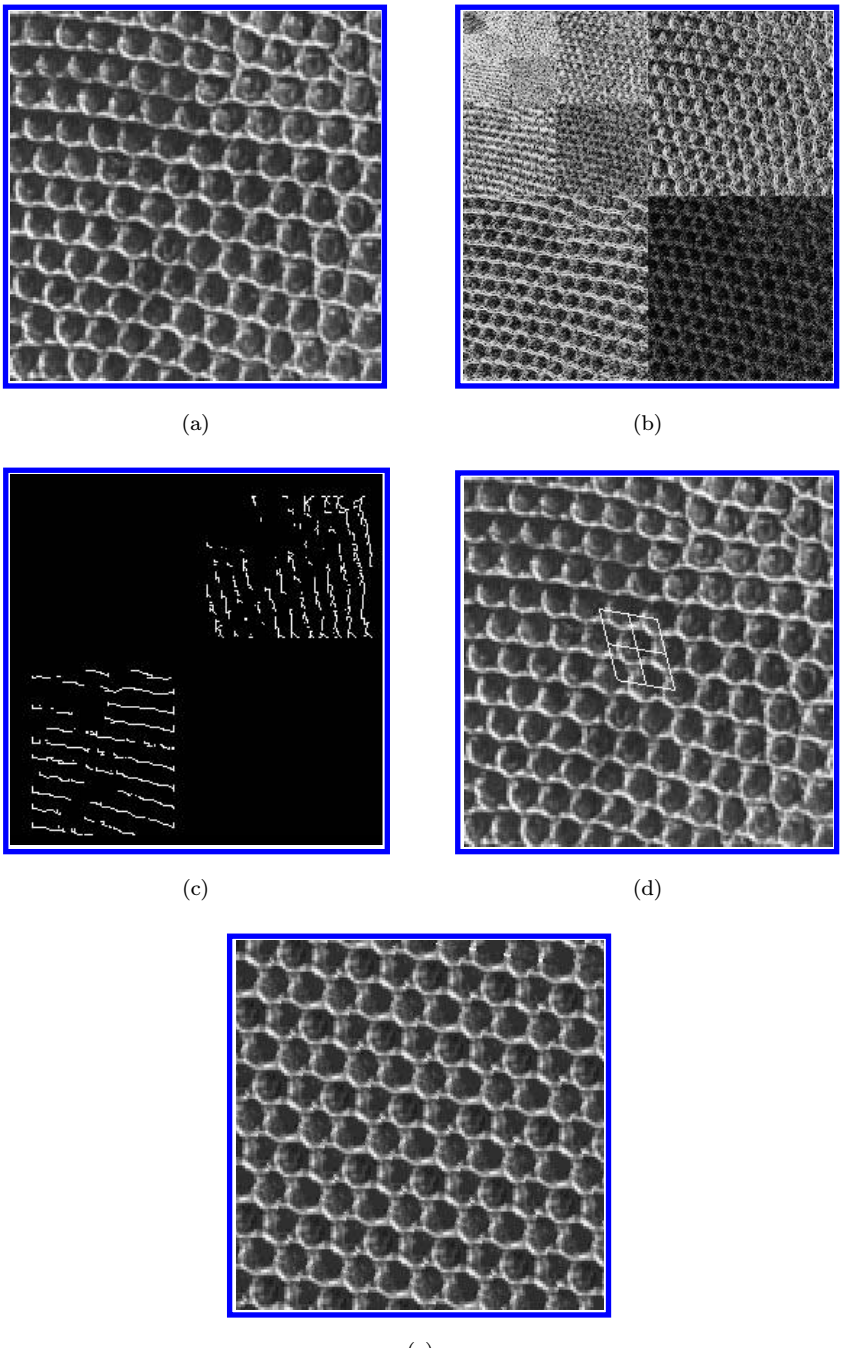


Fig. 12. The extraction and synthesis result of a D3 image. (a) A D3 image. (b) The wavelet-transformed image. (c) The edge-thresholded result of the representative LH and HL subimages. (d) The original image with four primitives marked. (e) The synthesized image.

Figure 12 shows another example with texture primitives formed by diagonal displacement vectors. The LH and HL subimages of decomposition level one have the most prominent edges and are correctly selected for displacement vectors calculation. The selected subimages and extracted edges are shown in Fig. 12(c). It is worth mentioning that there are some irregular texture patterns near the corners of the original image. As four texture primitives near the center part of the image are used to synthesize the texture, the synthesized texture image is more regular compared to the original image. The contrast values of all subimages are shown in Table 2.

Figure 13 shows a D102 image from Brodatz Album. The texture primitives contain circular objects. Although most of the extracted edges shown in Fig. 13(c) are broken, the displacement vectors are still extracted correctly. As shown in Figs. 13(d) and 13(e), the texture primitives are correctly extracted and the synthesized textured image is quite similar to the original image shown in Fig. 13(a). The contrast values of all subimages are shown in Table 3.

Figure 14 shows the processing result of a D11 image from Brodatz album. As shown in Fig. 14(b), there are no prominent edges for most subimages. Figure 14(c) shows that only one HL subimage is selected as SHL^{j^*} at decomposition level 3. Table 4 shows the contrast values of all subimages of D11. As the minimum threshold T_c is set as 2460 in our experiments, only the HL subimage of decomposition level three is selected for displacement vector calculation. Since only one displacement vector with angle of 90, ν_1, is detected, the texture primitive is considered as a strip shape as shown in Fig. 14(d). The final synthesized image is shown in Fig. 14(e).

Table 2. The contrast values of all subimages of D3.

Decomposition Level	LH	HL
1	3704.5	3213.9
2	3104.1	2616.1
3	2258.5	2176.0
4	1890.3	2124.4
$T_c = 2460$		

Table 3. The contrast values of all subimages of a D102 image.

Decomposition Level	LH	HL
1	3948.4	3647.5
2	2373.1	2423.9
3	1808.6	1956.1
4	1960.1	1717.8
$T_c = 2460$		

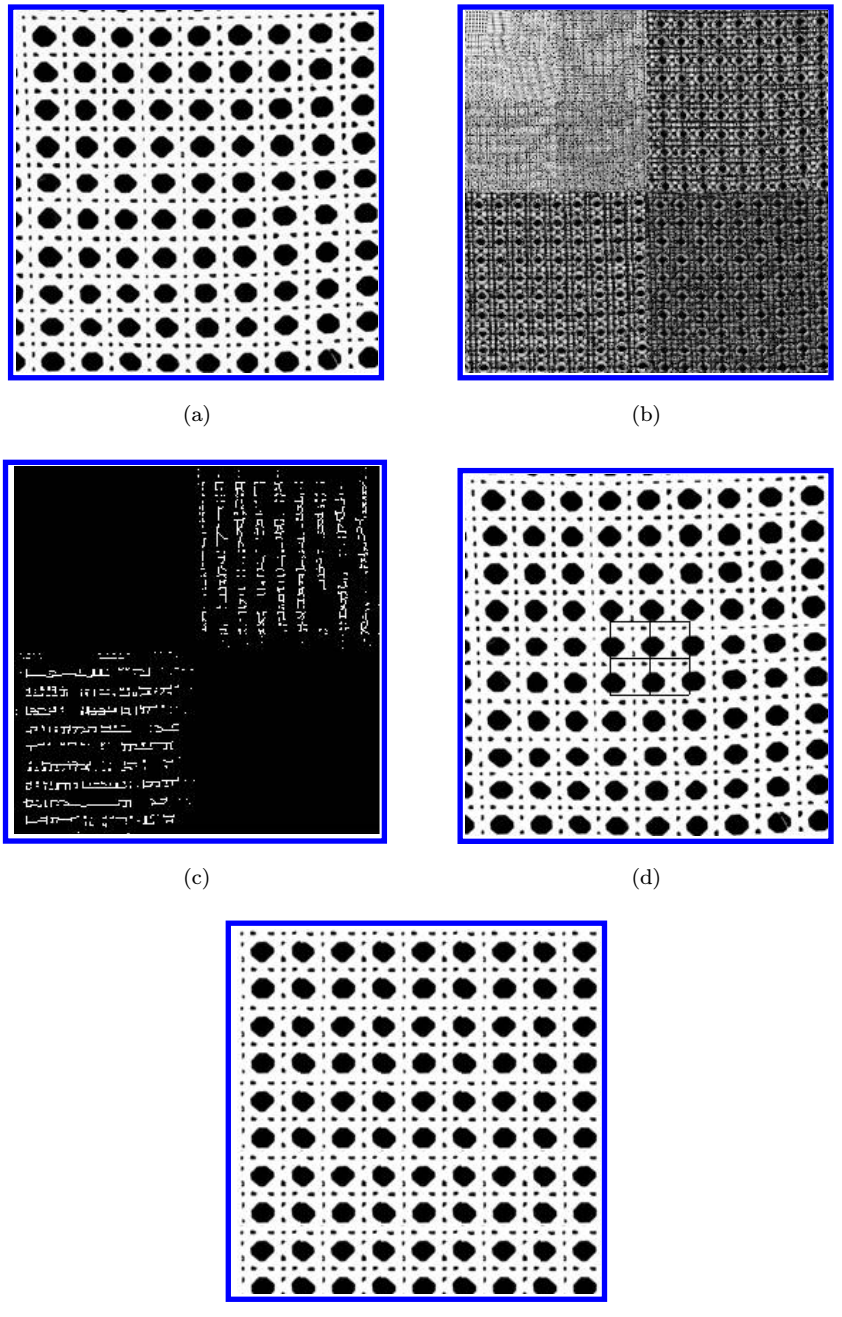


Fig. 13. The extracted primitives and synthesized image for a D102 image. (a) A D102 image from Brodatz album. (b) The result image of applying wavelet transform on (a). (c) The edge-thresholded result of the representative LH and HL subimages. (d) The original image with four primitives marked. (e) The synthesized image.

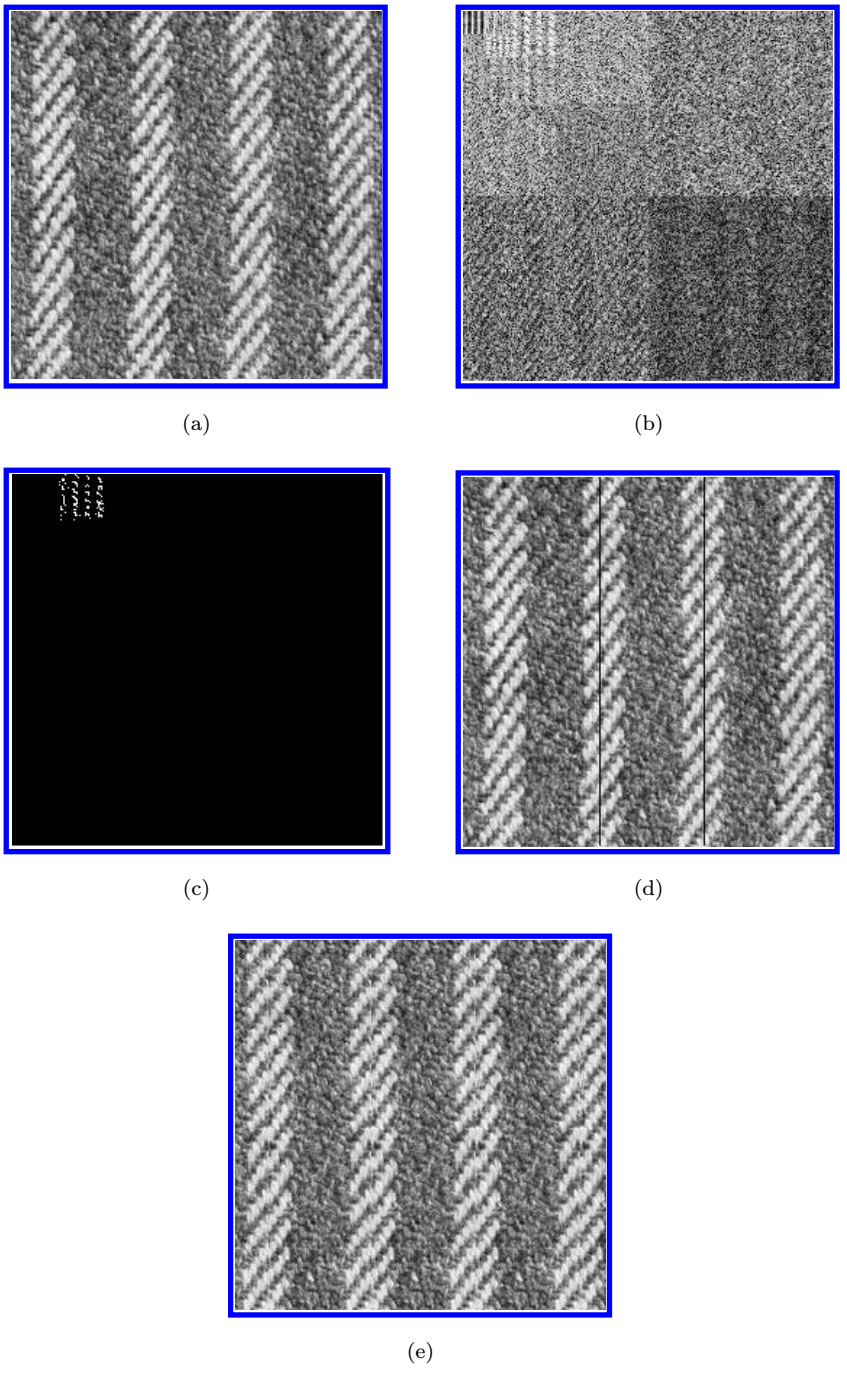
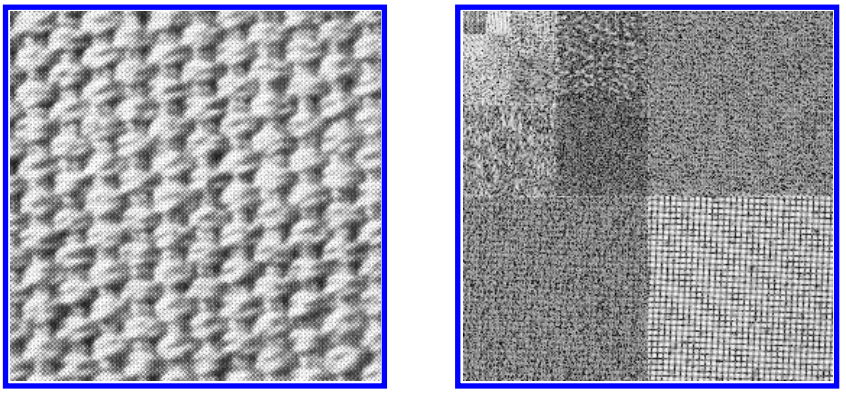


Fig. 14. The extraction and synthesis result of a D11 image. (a) A D11 image. (b) The wavelet-transformed image. (c) The edge-thresholded result of the representative LH and HL subimages. (d) The original image with the primitive marked. (e) The synthesized texture image.

Table 4. The contrast values of all subimages of D11.

Decomposition Level	LH	HL
1	2274.4	2389.2
2	2424.9	2457.5
3	2107.4	2501.1
4	1938.9	2390.7
$T_c = 2460$		

Figure 15(a) shows a D77 image of Brodatz album, and Fig. 15(b) shows the result of wavelet transform. It is obvious that there are no prominent edges for displacement vector detection in any subimages, therefore this image is rejected by the proposed method for texture primitive detection. Table 5 shows the contrast values for all subimages. All contrast values are far below the minimum threshold T_c set. For images with no prominent edges, the proposed method is not suitable. To treat this problem, when an image is rejected, other methods such as those in Ref. 18 might be adopted to cope with this kind of textured images, but computational



(a) (b)

Fig. 15. The wavelet transform result of a D77 image. (a) A D77 image. (b) The wavelet-transformed image.

Table 5. The contrast values of all subimages of D77.

Decomposition Level	LH	HL
1	2257.1	2127.1
2	2176.2	2084.6
3	2253.7	2228.8
4	1525.7	1844.9
$T_c = 2460$		

complexity will increase. For most regular textured images with prominent edges, the proposed method is a better choice in terms of computational efficiency.

4. Conclusion

In this paper, we propose a new method for extracting texture primitives based on wavelet transform. A subimage selection method is provided to select subimages with prominent edges for displacement vector detection. An edge thresholding method is then conducted on selected LH and HL subimages to locate edges. By using Hough transform for line detection, the displacement vectors of a texture primitive are extracted. The operations used in the proposed method are simple and are thus efficient as well. Besides, finer quantization for angles or distances extraction is used, thus the extracted primitives are quite accurate. In addition, the texture primitives extracted match with those observed by human perception. Textured images with no prominent edges will be rejected by the proposed method. In that case, other methods based on auto-correlation function could be used, computational complexity is increased. Textured images with different characteristics are chosen to demonstrate the effectiveness of the proposed method. The proposed method can be used in texture applications like synthesis, classification and data compression.

Acknowledgments

The authors would like to express their appreciation to the reviewers whose invaluable comments have made the presentation of this paper more precise.

References

1. R. Bajcsy, "Computer description of texture surface," *Proc. 1973 Int. Conf. Artificial Intelligence*, Stanford, CA, 1973, pp. 572–579.
2. R. Bajcsy and L. Lieberman, "Texture gradient as a depth cue," *Comput. Graph. Imag. Proc.* **5**, 1 (1976) 52–67.
3. D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Patt. Recogn.* **13**, 2 (1981) 111–122.
4. A. C. Bovik, M. Clark and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. Patt. Anal. Mach. Intell.* **12**, 1 (1990) 55–73.
5. P. Brodatz, *Textures — A Photographic for Artists and Designers*, Dover, NY, 1966.
6. T. Chang and C. C. J. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *IEEE Trans. Imag. Process.* **2**, 4 (1993) 429–442.
7. J. L. Chen and Kundu, "Unsupervised texture segmentation using multi-channel decomposition and hidden Markov models," *IEEE Trans. Imag. Process.* **4**, 5 (1995) 603–620.
8. R. W. Conners and C. A. Harlow, "Towards a structural textural analyzer based on statistical methods," *Comput. Graph. Imag. Process.* **12**, 3 (1980) 224–256.
9. I. Daubechies, *Ten Lectures on Wavelets*, CBMS series, SIAM, Philadelphia.
10. R. M. Haralick *et al.*, "Texture features for image classification," *IEEE Trans. Syst. Man Cybern.* **SMC-3**, 6 (1973) 610–621.

11. R. M. Haralick, "Statistical and structural approaches to texture," *Proc. 4th Int. Joint Conf. Pattern Recognition*, 1979, pp. 45–60.
12. D. C. He and L. Wang, "Textural filters based on the texture spectrum," *Patt. Recogn.* **24**, 12 (1991) 1187–1195.
13. A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Patt. Recogn.* **24**, 12 (1991) 1167–1186.
14. H. B. Kim and R. H. Park, "Extracting spatial arrangement of spectral textures using projection information," *Patt. Recogn.* **25**, 3 (1992) 237–247.
15. K. L. Laws, "Rapid texture identification," *Proc. SPIE* **238** (1980) 376–380.
16. J. G. Leu and W. G. Wee, "Detecting the spatial structure of natural textures based on shape analysis," *Comput. Vis. Graph. Imag. Process.* **31** (1985) 67–88.
17. S. Z. Li, *Markov Random Field Modeling in Computer Vision*, Springer-Verlag, NY, 1995.
18. H. C. Lin, L. L. Wang and S. N. Yang, "Extracting periodicity of a regular texture based on autocorrelation functions," *Patt. Recogn. Lett.* **18** (1997) 433–443.
19. C. S. Lu and P. C. Chung, *Wold Features for Unsupervised Texture Segmentation*, *14th Int. Conf. Pattern Recognition*, Vol. II, Special Topic in Computer Vision, Australia, 1988, pp. 1689–1693.
20. C. S. Lu, P. C. Chung and C. F. Chen, "Unsupervised texture segmentation via wavelet transform," *Patt. Recogn.* **30**, 5 (1997) 729–742.
21. S. Y. Lu and K. S. Fu, "A syntactic approach to texture analysis," *Comput. Graph. Imag. Proc.* **7**, 3 (1978) 303–330.
22. S. G. Mallat, "A theory of multiresolution signal decomposition: the wavelet transform," *IEEE Trans. Patt. Anal. Mach. Intell.* **11**, 7 (1989) 674–693.
23. T. Matsuyama, K. Saburi and M. Nagao, "A structural analyzer for regularly arranged textures," *Comput. Graph. Imag. Process.* **18** (1982) 259–278.
24. T. Matsuyama, S. I. Miura and M. Nagao, "Structural analysis of natural textures by Fourier transformation," *Comput. Vis. Graph. Imag. Process.* **24**, 3 (1983) 347–362.
25. C. G. Rafael and E. W. Richard, *Digital Image Processing*, Addison Wesley, 1992.
26. E. Salari and Z. Ling, "Texture segmentation using hierarchical wavelet decomposition," *Patt. Recogn.* **28**, 12 (1995) 1819–1824.
27. V. V. Starovoitov, S. Y. Jeong and R. H. Park, "Investigation of texture periodicity extraction," *SPIE Proc. Visual Communications and Image Processing*, Vol. 2501, 1995, pp. 870–881.
28. A. Teuner, O. Pichler and B. J. Hosticka, "Unsupervised texture segmentation of images using tuned matched Gabor filters," *IEEE Trans. Imag. Process.* **4**, 6 (1995) 863–870.
29. Tomita *et al.*, "Description of texture by a structural analysis," *IEEE Trans. Patt. Anal. Mach. Intell.* **PAMI-4**, 2 (1982) 183–191.
30. F. M. Vilnrotter, R. Nevatia and K. E. Price, "Structural analysis of natural textures," *IEEE Trans. Patt. Anal. Mach. Intell.* **PAMI-8** (1986) 79–89.
31. W. R. Wu and S. C. Wei, "Rotational and gray scale transform invariant texture classification using spiral re-sampling, sub-band decomposition, and hidden Markov model," *IEEE Trans. Imag. Process.* **5**, 10 (1996) 1423–1435.
32. S. W. Zucker and D. Terzopoulos, "Finding structure in co-occurrence matrices for texture analysis," *Comput. Graph. Imag. Process.* **12**, 3 (1980) 286–308.

Kuen-Long Lee received both the B.S. and M.S. degrees in computer and information science from National Chiao Tung University, Taiwan, in 1988 and 1990, respectively. He is currently a Ph.D. student in the Department of Computer and Information Science at National Chiao Tung University.

His research interests include image processing, pattern recognition, texture analysis and image retrieval.

Ling-Hwei Chen received the B.S. degree in mathematics and M.S. degree in applied mathematics from National Tsing Hua University, Hsinchu, Taiwan in 1975 and 1977, respectively, and the Ph.D. in computer engineering from National Chiao Tung University, Hsinchu, Taiwan in 1987.

From August 1977 to April 1979, she worked as a research assistant in the Chung-Shan Institute of Science and Technology, Taoyan, Taiwan, and then as a research associate in the Electronic Research and Service Organization, Industry Technology Research Institute, Hsinchu, Taiwan. From March 1981 to August 1983, she worked as an engineer in the Institute of Information Industry, Taipei, Taiwan. She is now a Professor in the Department of Computer and Information Science at the National Chiao Tung University.

Her current research interests include image processing, pattern recognition, document processing, image compression and image cryptography.

This article has been cited by:

1. Rocio A. Lizarraga-Morales, Raul E. Sanchez-Yanez, Victor Ayala-Ramirez. 2013. Fast texel size estimation in visual texture using homogeneity cues. *Pattern Recognition Letters* **34**:4, 414-422. [CrossRef]
2. Roman M. Palenichka, Marek B. Zaremba, Rokia Missaoui. 2006. Multiscale model-based feature extraction in structural texture images. *Journal of Electronic Imaging* **15**:2, 023013. [CrossRef]