Chapter 5 Basic Concept and Models of the K-views

"Heaven signifies night and day, cold and heat, times and seasons."
~ Sun Zhu

In this chapter, we introduce the concepts of "view" and "characteristic view". This view concept is quite different from those of gray-level co-occurrence matrix (GLCM) and local binary pattern (LBP). We emphasize on how to precisely describe the features of a texture and how to extract texture features directly from a sample patch (i.e. sub-image), and how to use these features to classify an image texture. The view concepts and related methods work with a group of pixels instead of single pixel. Three principles are used in this work for developing the model: 1) texture features from a view should carry as much information as possible for image classification; 2) the algorithm should be kept as simple as possible; and 3) the computational time to distinguish different texture classes should be the minimum. The view related concepts are suitable for textures that are generated by one or more basic local patterns and repeated in a periodic manner over some image region. The set of characteristic views is a powerful feature extraction and representation to describe an image texture. As different textures show different patterns, the patterns of a texture also show different views. If a set of characteristic views is properly defined, it is possible to use this set of characteristic views for texture classification. The K-views template is an algorithm that uses a number of characteristic views, denoted by K, for the classification of images. The K-views algorithm is suitable for classifying image textures that have basic local patterns repeated in a periodic manner. Several variations of the basic K-views model are given in chapters 6, 7 and 8.

5.1 View Concept and a Set of Views

Human beings are capable of using the texture in the interpretation of a photograph for the targets of interest. It is very natural for a machine to use the same feature of texture for the recognition. Researchers in image texture analysis have proposed many innovative methods for image texture classification. These methods can be divided into two major categories: the first category is based on the features with a high degree of spatial localization, which can use edge detection operations for recognition. The major problem with this approach is that it is difficult to distinguish the texture boundaries and the micro-edges located in the same texture. The second category is based on the discrimination function using several texture features. The classification accuracy in this category depends upon the discriminative power of the extracted texture features. Many methods have been developed to extract features either using statistical or other algorithms as discussed in chapter 2. In most cases, the feature is represented numerically by a vector, which is composed of real numbers (vector components) derived from a neighborhood of the corresponding class. In this chapter, we present a different framework for representing texture features for the classification. We describe how to extract texture features directly from a sample sub-image of a texture and how to use them to classify an image. As each texture class has a characteristic feature that can distinguish this class from others, this characteristic feature may show different "views". When we determine if a pixel belongs to a specific texture class during the classification, if the spatial neighborhood of the pixel (which is a small image patch) is considered, we may be able to look up for a set of the "views" of texture classes to determine the categorization of this image patch [7, 12]. In other words, one can observe that any local patch of a texture consists of only a few patterns, which are called views in this context.

In the following, we illustrate the concept of the view by using a simple image as shown in Figure 5.1 that contains two different texture classes. One texture class consists of parallel vertical lines. The other consists of two intersected sets of diagonal lines. These two arrangements show two different textures. This type of pattern structure can be taken as the basic element of measure for image textures. Figure 5.1 also shows that an image texture not only depends on the values of its composed pixels, but also on the spatial arrangement of those pixels. Please note that it is impossible for a per-pixel based classification method to classify this texture image correctly. In order to classify this image into two different texture classes, we can use a correlation matching method. The following steps are used to illustrate the correlation matching method.

The Correlation Matching Method:

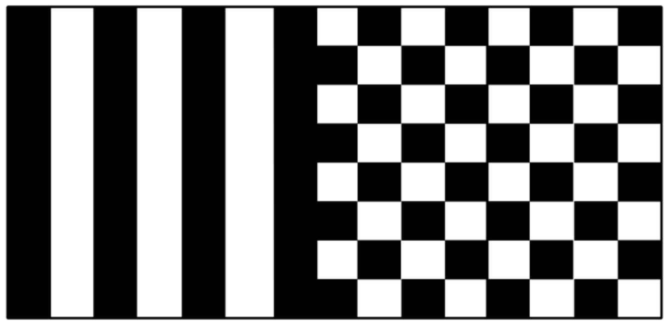
Step 1: Select randomly in the area of the texture class a sample sub-image for each texture class from the original image. An example of selected sample sub-images, sample 1 and sample 2, are shown in Figure 5.2(a) and (b). Note that the size of these two sample sub-images do not have to be the same.

Step 2: Take a small image template (we can call this template a patch) from the original image being classified as shown in Figure 5.2(c) and (d). The small patch should be much smaller than any of the sample sub-image. Then, find the best match between this small image patch and the sample sub-images.

Step 3: If the best match occurs in sample sub-image **k**, classify all the pixels (in the original image) corresponding to this small image patch into class **k**. (or if the small image patch is regarded as a neighborhood of one pixel, classify only that pixel to class **k**). In Figure 5.1, K is set to 2.

Step 4: Repeat Steps 2 and 3 until the entire original image is classified.

In Steps 2 and 3, when we perform the classification of an image, a small patch is taken and look up a set of sub-images extracted from the sample image to determine which texture class this small patch belongs. These steps will be repeated for each pixel in a neighborhood of the patch size in the original image. In a sense, this operation is very similar to how we apply the spatial filter to an image.



(a)

0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	0	1	0	1	0	1	0	1
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	0	1	0	1	0	1	0	1
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	0	1	0	1	0	1	0	1
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	0	1	0	1	0	1	0	1

(b)

Figure 5.1

(a) an image shows two texture classes and (b) the corresponding pixel values.

0	1	0	1
0	1	0	1
0	1	0	1

(a)

0	1	0	1
1	0	1	0
0	1	0	1

(b)

0	1
0	1

(c)

0	1
1	0

(d)

Figure 5.2

(a) and (b) show sample sub-images with a size of 3 x 4 from two different textures, respectively, from Figure 5.1 and (c) and (d) small patches (size of 2 x 2) taken from (a) and (b), respectively.

The correlation matching method, which measures the similarity between a small image patch and a sample sub-image, can be defined in several ways. One simple method is to use Euclidean distance. Suppose that a small image patch taken from an original image which has a size of m × n. A sample sub-image with a size of M x N will contain (M - m + 1)*(N - n +1) small patches of size m × n with overlapping. This is illustrated in Figure 5.3.

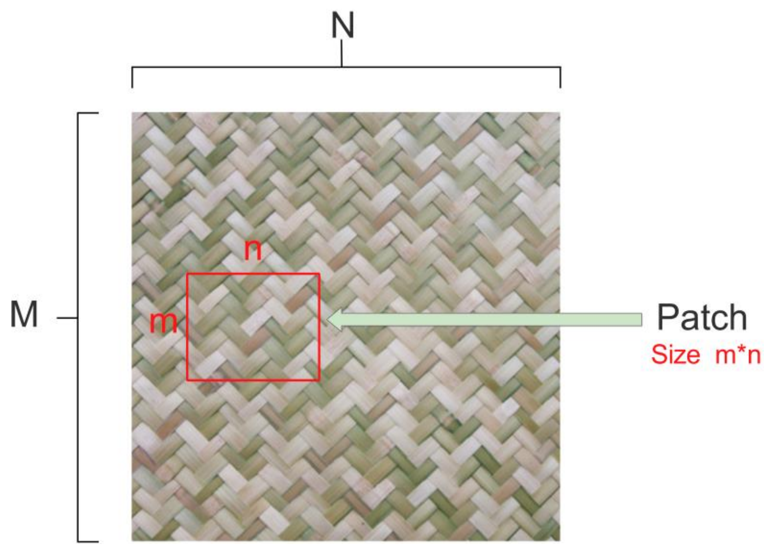


Figure 5.3

A sample sub-image with a size of M x N which contains (M - m + 1)*(N - n +1) small patches of size m × n with overlapping.

If we use V_o to denote a small patch taken from an original image and V_i (i = 1 to (M - m + 1)*(N - n +1)) to denote a small image patch taken from a sample sub-image representing a textural class and d_i denote the Euclidean distance between V and V_i. We assume that both V_o and V_i have the same size. The similarity which is measured by the Euclidean distance, say $S_{V_oV_i}$, between V_o and a patch V_i, can be expressed by the following equation:

$$S_{V_oV_i} = \min\{d_1, d_2, d_3, \ldots, d_k, \ldots, d_{(M-m+1)*(N-n+1)}\} \quad (5.1)$$

Please note that Equation 5.1 will be repeated for each sample sub-image taken from a textural class. The textural class i (from 1 to C) to which V_o belongs can be determined by Equation 5.2.

$$S_{V_i} = \min\{S_{V_1}, S_{V_2}, \ldots, S_{V_i}, \ldots, S_{V_C}\} \quad (5.2)$$

where C is the total number of textural classes.

This method can accurately classify an image such as the one shown in Figure 5.1. If the small image patch has a size of 2 × 2, this method can classify this image well into 2 texture classes. Lee and Philpot described a similar method in their work [8].

Here, we call each small image patch, V, a "view". There are two different views in Figure 5.2(a), if the view size is 2 × 2: $\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$ and $\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$. In total, there are six views in this sample sub-image (i.e. (3 – 2 + 1) * (4 – 2 + 1)). Among a total of six views, there are four identical views of $\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$ and two identical views of $\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$. A view is of a size of m × n, where m > 0 and n > 0. When m = 1 and n = 1, a view is just a single pixel.

For a simple representation, a view with a size of m by n can be denoted by a vector:

$$(x_{11}, x_{12}, \ldots, x_{1n};\ x_{21}, x_{22}, \ldots, x_{2n};\ \ldots;\ x_{m1}, x_{m2}, \ldots, x_{mn})^T \quad (5.3)$$

where T is the transpose. In Equation 5.3, a group of values separated from others by semicolons corresponds to a row in the view. Each value in this group corresponds to a pixel value in the view. Hence $\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$ can be expressed by $(1, 0; 1, 0)^T$. All of these views from a sample sub-image form a set called a *view set*. A view set is an exemplar set for an image texture class.

Now it is clear that the correlation matching method is in fact a method to compare a view with those in the view sets formed from different sample sub-images (i.e. textures).

5.2 Set of Characteristic Views and the K-views Template Algorithm (K-views-T)

Although the correlation matching method can accurately classify a textural image, it is a computational intensive algorithm. To have a representative set of views, a large sample sub-image has to be chosen. A large sample sub-image indicates that it will increase the computation time. In fact, it is not necessary to compare a view (from an image being classified) with the entire set of views extracted from a sample sub-image in the matching process because this original view set may contain several identical views, and some views are very similar. It will be more efficient if a representative set of views can be chosen from an original set of views which has similar or identical views. In doing so, it will not affect the matching result. If a small representative set of views can be derived from the entire large view set of a sample sub-image, the computation time for the matching method will be dramatically reduced. This representative set of views derived from an original set of views will be called the set of **characteristic views**. A view in the set of characteristic views is called the **characteristic view** and denoted by V_{cs}.

An original set of views (abbreviation *view set* and denoted by V_s) can be formulated as:

$$V_s = D@V_{cs} \qquad (5.4)$$

where V_s denotes an original set of views, V_{cs} a set of all different views from a sample sub-image, and D the relative frequency (or distribution) of the views (i.e. elements) in the set of different views (V_{cs}). The notation @ is the operator that we have chosen to relate the datagram to the frequency of each characteristic view in V_{cs}. The datagram will be explained later.

For example, with the sample sub-image in Figure 5.2(a) and the view with a size of 2 × 2, the *view set*, V_s, of sample sub-image is

$$\left\{\begin{pmatrix}0 & 1\\0 & 1\end{pmatrix},\begin{pmatrix}1 & 0\\1 & 0\end{pmatrix},\begin{pmatrix}0 & 1\\0 & 1\end{pmatrix},\begin{pmatrix}0 & 1\\0 & 1\end{pmatrix},\begin{pmatrix}1 & 0\\1 & 0\end{pmatrix},\begin{pmatrix}0 & 1\\0 & 1\end{pmatrix}\right\} \qquad (5.5)$$

or {(0, 1; 0, 1), (1, 0; 1,0), (0, 1; 0, 1), (0, 1; 0, 1), (1, 0; 1,0), (0, 1; 0, 1)} (5.6)

please note that the transpose T for each vector in Equation 5.6 is omitted for the simplicity. Similarly, notation T will be omitted in the following vector representations.

In this view set, there are four views of the pattern (0, 1; 0, 1), and two views of the pattern (1, 0; 1,0). The distribution can be represented as

$$V_s = (4, 2)@\{(0, 1; 0, 1), (1, 0; 1,0)\} \qquad (5.7)$$

where (4, 2) is the datagram, {(0, 1; 0, 1) (1, 0; 1, 0)} is the set of characteristic views. If we choose to use only one characteristic view, V_s can be further simplified as:

$$V_s = (6)@((0.3, 0.7; 0.3, 0.7)) \tag{5.8}$$

where the value 6 is the total number of views in the view set V_s, 0.3 is the average value of the distribution in the first component of all vectors in Equation 5.6, 0.7 is the average value of the distribution in the second component, and so on. If the size of the sample sub-images is getting larger, the average of the distribution is closer to 0.5 in this case, then $Vs \approx (6)@\{(0.5, 0.5; 0.5, 0.5)\}$. Similarly,

the View Set of the sample sub-image in Figure 5.2

$= \{(0, 1; 1, 0), (1, 0; 0, 1), (0, 1; 1, 0), (1, 0; 0, 1), (0, 1; 1, 0), (1, 0; 0, 1)\}$

$= \{3@(0, 1; 1, 0), 3@(1, 0; 0, 1)\}$

$= (3, 3)@\{(0, 1; 1, 0), (1, 0; 0, 1)\}$

$$= (6)@\{(0.5, 0.5; 0.5, 0.5)\} \tag{5.9}$$

The correlation matching method now will use the set of characteristic views to classify an image texture. We call this method the **K-views Template Algorithm (K-views-T)**, as the characteristic view is similar to a "template". The procedure of the K-views template algorithm based on the modification of correlation matching method is described in the following.

The K-views Template Algorithm (K-views-T)

Step 1: Select randomly a sample sub-image in the area of the textures for each textural class from the original image. In other words, N sample sub-images will be selected for N textural classes. The size of each sub-image can be different.

Step 2: Extract a view set, V_s from each sample sub-image.

Step 3: Determine the value of K for each view set, and derive K-views for each characteristic view set from each sample sub-image using the K-means algorithm or fuzzy C-means algorithm. The number of views, K, may vary for each texture class (i.e. sample sub-image).

Step 4: In the matching process, each view (a small image patch), say *V*, of the original image being classified will be compared with each characteristic view in each set of the characteristic views and find a best match (a high correlation). Please note that the size of each view is the same.

Step 5: If the best matching characteristic view belongs to the characteristic view set j, classify all pixels in the view V to class j where $j = 1, \ldots, N$. (If the view is regarded as a neighborhood of one pixel, classify only that pixel to class j).

Step 6: Repeat Steps 4 and 5 for each view in the original image being classified.

The procedure sketched above leaves two parameters undefined. One is the size of the view and the other is the number of the characteristic views, K. If a specific pattern repeated in the texture class frequently, the size of the view can be small. Otherwise, the size of the view should be large. The larger the size of the view is, the more information about the texture the view can carry. The number of the characteristic views should depend on both the texture structure in the image and the similarity between texture classes. The smaller K of the characteristic views is, the less powerful the characteristic views can describe for the texture. For the image shown in Figure 5.1, if K is set to 1, the classification result will not be satisfied.

Deriving the representative characteristic views from the view set of a sample sub-image is a simple and straightforward process. As long as the number of characteristic views (K) is determined, most clustering methods described in chapter 3 can be used to select a representative set of characteristic views and obtain K cluster centers [9]. It can be proved that the cluster centers derived from these methods are those representative characteristic views of the view set by minimizing the objective function.

If the size of the view is selected appropriately, the K-views template method can have an optimal performance in texture classification. However, if the size of the view is very small, different view sets may have some views in common. Hence, different sets of characteristic views may also have some same or similar characteristic views. For example, if the size of the view is 1×2 for the sample sub-images shown in Figure 5.2 (a) and (b), we will have the following set of views:

the set of views for sample sub-image-1

$$= \{(0, 1), (1, 0), (0, 1), (0, 1), (1, 0), (0, 1), (0, 1), (1, 0), (0, 1)\}$$

$$= (6, 3)@\{(0,1), (1, 0)\}, \qquad (5.10)$$

and

the set of views for sample sub-image-2

$$= \{(0, 1), (1, 0), (0, 1), (1, 0), (0, 1), (1, 0), (0, 1), (1, 0), (0, 1)\}$$

$$= (5, 4)@\{(0,1), (1, 0)\} \qquad (5.11)$$

These two sets of characteristic views for two different textural samples are almost the same. In this situation, we cannot totally depend on these sets of characteristic views in order to obtain a correct classification. One solution to this problem is to increase the size of the view. However, before we do that, we should ask one question: are the sample sub-images really good representatives for textures? Is it likely that a sample sub-image contains a part that is similar to another texture class? This leads us to a second solution. The information from the distribution in the datagram can be used. Suppose that two characteristic views in two different sets are identical (or similar). If 40% of sample sub-image-1 can be classified by this characteristic view and only 5% of the sample sub-image-2 can be classified by the same characteristic view, we can just remove this characteristic view from the second set of characteristic views.

In our experiments, the size of the view is set to between 10 × 10 and 15 × 15. If a view size of 10 x 10 is chosen, a sample sub-image of size 50 × 50 has at most 41*41 different views. It is unlikely that these sets of characteristic views for different texture classes will have an identical characteristic view. In the empirical study, it shows that some of views coming from sample sub-image class i may be classified into class j. In such a situation, the third solution is to increase K of the characteristic views in a set so that more characteristic views can be used to accurately describe the characteristic of a texture class. In summary, three solutions to this problem can be used: 1) the first solution is to increase the size of the view, 2) the second solution is to use both the information from the distribution in the datagram and the set of characteristic views, and 3) the third solution is to increase the number of characteristic views in each set for a texture class.

Besides the above three solutions, there exists another solution from the perspective of the histogram point of view. Assume that there are two sets of characteristic views for two textural classes and arranged as histograms (we will call this datagram [12]) shown in Figure 5.4: class 1 is (1, 1, 2, 7, 2)@$(V_4, V_5, V_6, V_7, V_8)$ and class 2 is (1, 3, 6, 3, 1)@$(V_1, V_2, V_3, V_4, V_5)$. Characteristic view 7 appears frequently in texture class 1 and only occurs in class 1. Similarly, characteristic views 2 and 3 appear frequently in texture class 2 and only occurs in class 2. If we take an image patch containing characteristic view 7, we can say that this image patch belongs to texture class 1 since characteristic view 7 is the most distinguished view in this set. If it contains characteristic view 2 or 3, the entire patch will be classified into class 2 because these two characteristic views are the prominent features in this set. If the image patch consists of characteristic views 3 and 7, or, 2 and 7, that indicates that the patch is in the boundary of these two textural classes.

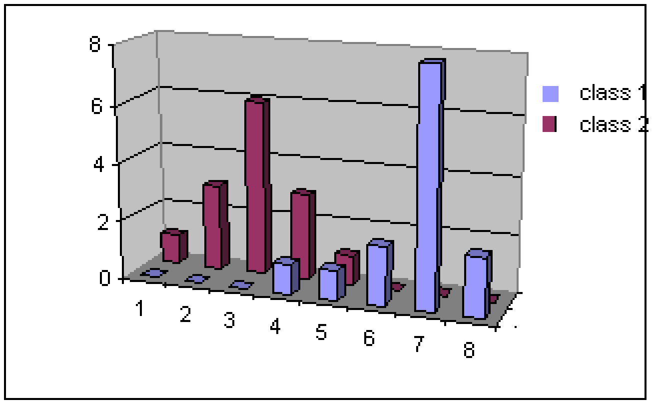


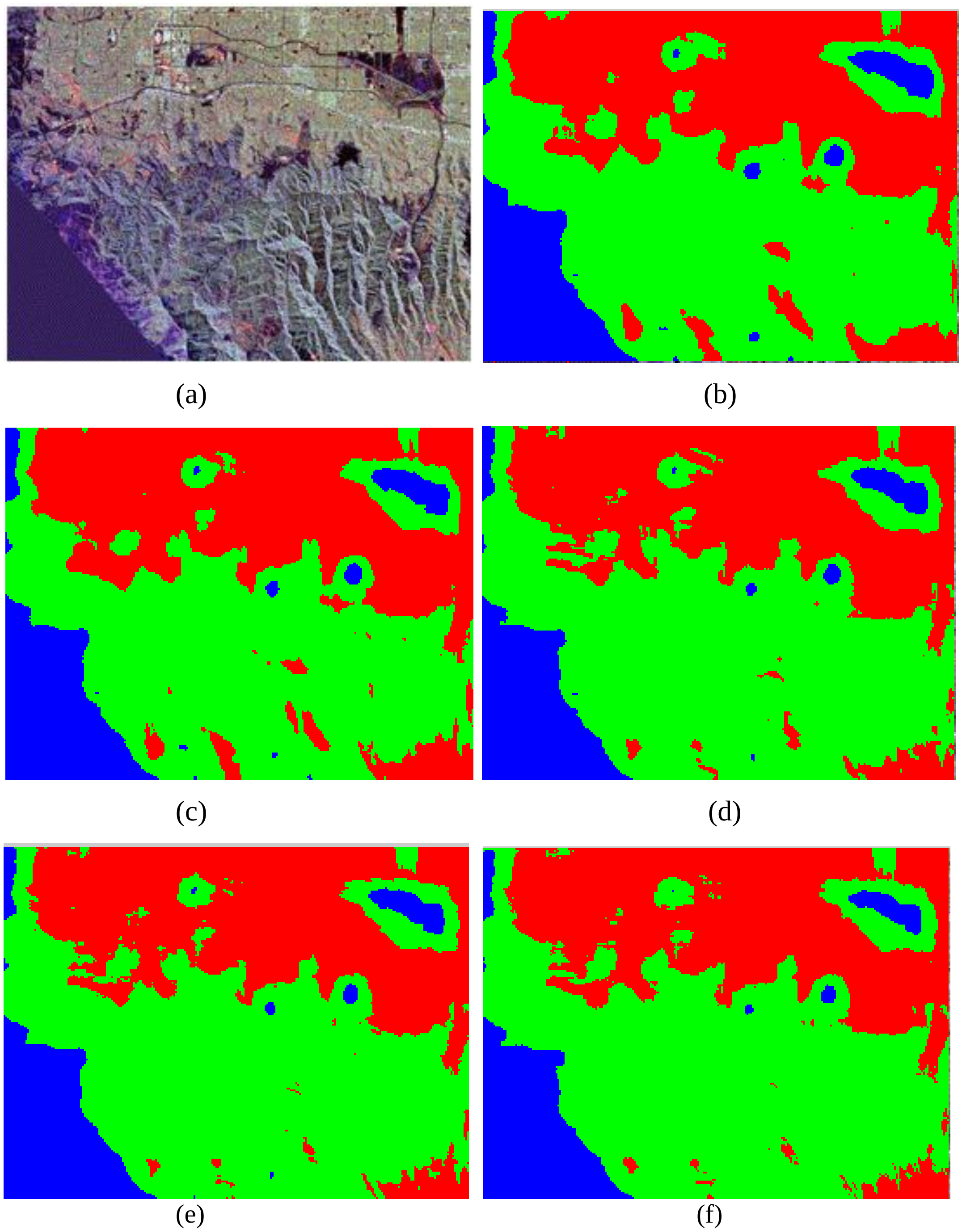
Figure 5.4
Two textural classes have some characteristic views in common. As shown in Equation 5.3, i.e. $V_s = D@V_{cs}$, y-coordinate represents D and x-coordinate represents V_{cs} which has five views $(V_4, V_5, V_6, V_7, V_8)$ for class 1 and $(V_1, V_2, V_3, V_4, V_5)$ for class 2 in the histogram.

In a gray-scale image, if the view size is set to 1 × 1, there will be at most 256 different characteristic views. The possible values are 0, 1, 2, 3, …, and 255. If two texture classes have the same set of characteristic views, the classification should totally depends on the view distribution in the datagram instead of the set of characteristic views. As defined in Equation 5.3, our texture classification model has been focusing on V_{cs}, instead of D. This is in contrast with most other texture classification algorithms.

The K-views template algorithm is suitable for textures that have periodically repeating patterns. For image texture that possesses some random structures, the K-views template may not be effective to distinguish them. In this situation, both the datagrams and sets of characteristic views can be used in the image texture classification algorithms. Characteristic views are powerful and simple to describe the characteristics of textures. Datagram is also a good feature for image classification [12]. The algorithm, which uses the datagram, will be discussed in chapter 6.

5.3 Experimental Results Using the K-views-T Algorithm

The K-views template algorithm was tested on satellite images in the experiments [7]. Figure 5.5(a) shows an original image that contains three texture classes, which are smooth land, ocean and mountain. Figure 5.5(b) shows the classified result using one characteristic view for each texture class. From Figure 5.5(c) to 5.5(f), the same K value was used for all the three texture classes. The value K was set to 2, 5, 10, and 20, respectively, for Figure 5.5(c), 5.5(d), 5.5(e), and 5.5(f). If a larger K is used for a set of characteristic views, the better the classification result one can achieve. Each texture class does not necessarily have the same number of characteristic views. This means that K may vary for each set of characteristic views of a texture class. For example, the ocean class and the smooth land can have much fewer characteristic views than that of the mountain class. Figure 5.6 shows five sets with a varying number of characteristic views (i.e. K-views) for the mountain class. We can see that the more number of characteristic views used in a set, the set of characteristic views is more close to the real image.



(a) (b) (c) (d) (e) (f)

Figure 5.5
Classification results using the K-views template algorithm. We assume that there are three texture classes in the image. (a) an original image, and the results of (b), (c), (d), (e) and (f) are obtained using the same K value (i.e. same number of characteristic views for each class) for all the three texture classes. The value K was set to 1, 2, 5, 10, and 20 for (b), (c), (d), (e), and (f), respectively.

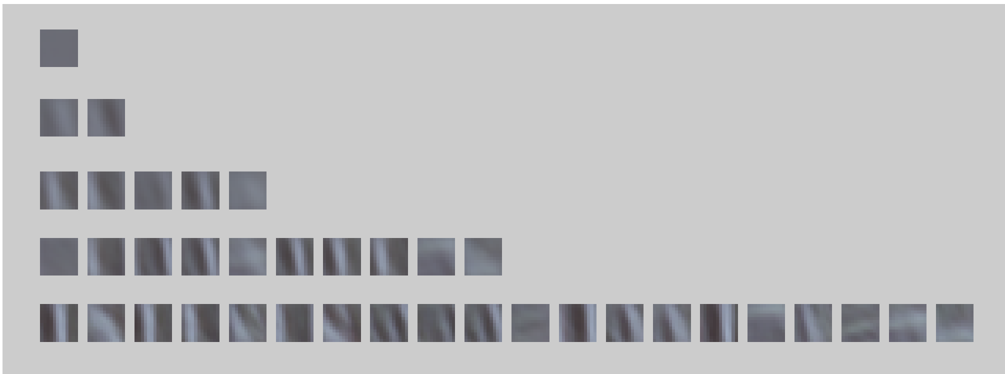
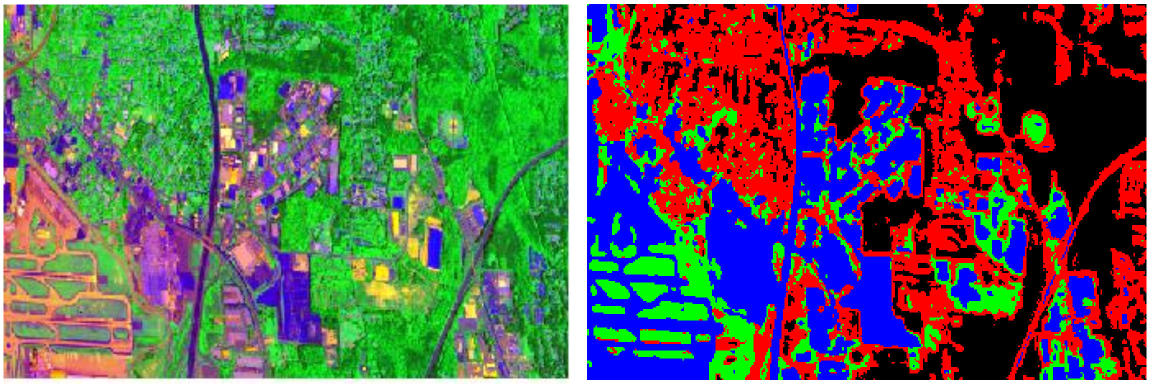


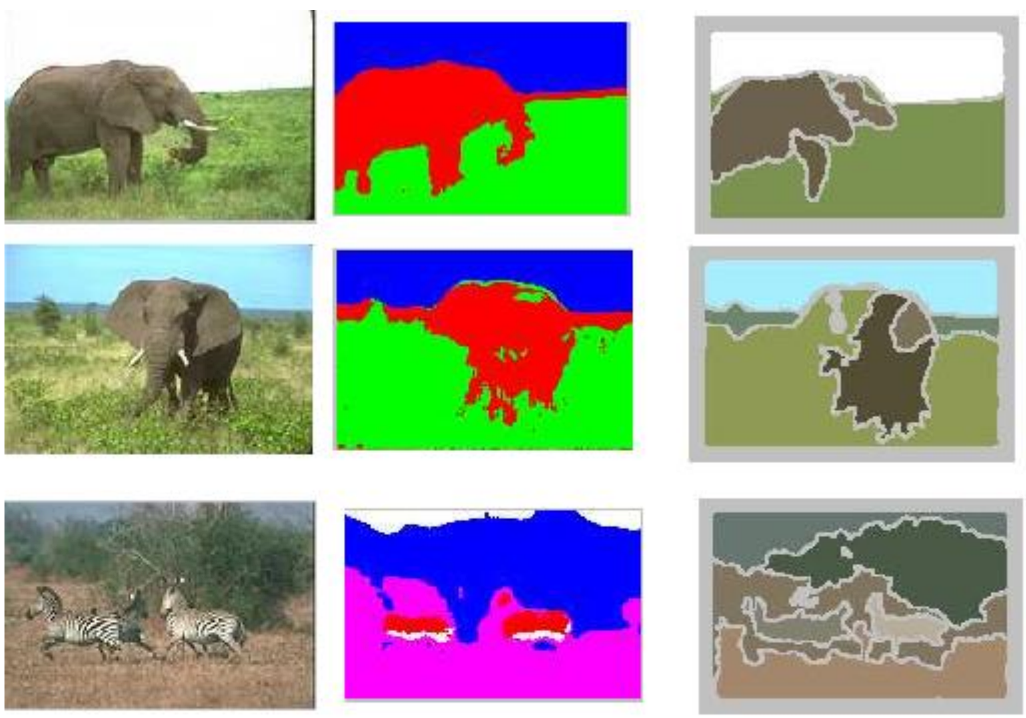
Figure 5.6
Five sets with a varying number of characteristic views (K-views) for mountain texture class; the K, from the first row, was set to 1, 2, 5, 10, and 20, until the last row.

Figure 5.7 shows the classification results on another remotely sensed image. The original image in Figure 5.7(a) is a region of Atlanta city in Georgia. The image is classified into the residential area (red), the lawns (green), the commercial area (blue), and the undeveloped area (black) in Figure 5.7(b). Figure 5.8 illustrates some classified results of animal images taken from the Berkeley website (http://sunsite.berkeley.edu/ImageFinder/) and compared with the results of the color region method. Figure 5.9 shows an image consisting of four different texture classes taken from Brodatz texture album and their classification results using the K-views-T algorithm. When the number of K-views increases, the classification accuracy is improved. Each texture class has a size of 100 ×100.



(a) (b)

Figure 5.7
a) a sub-image of Atlanta city in Georgia, USA and b) the classified image using the K-views template algorithm: residential area (red), lawns (green), commercial areas (blue), and undeveloped area (black).



(a) (b) (c)

Figure 5.8
a) three images of animals, b) classified results using the K-views template algorithm, and c) classified results using the color region method. (images in the column of (a) and segmented results in the column of (c) are taken from the Berkley website (http://sunsite.berkeley.edu/ImageFinder/)

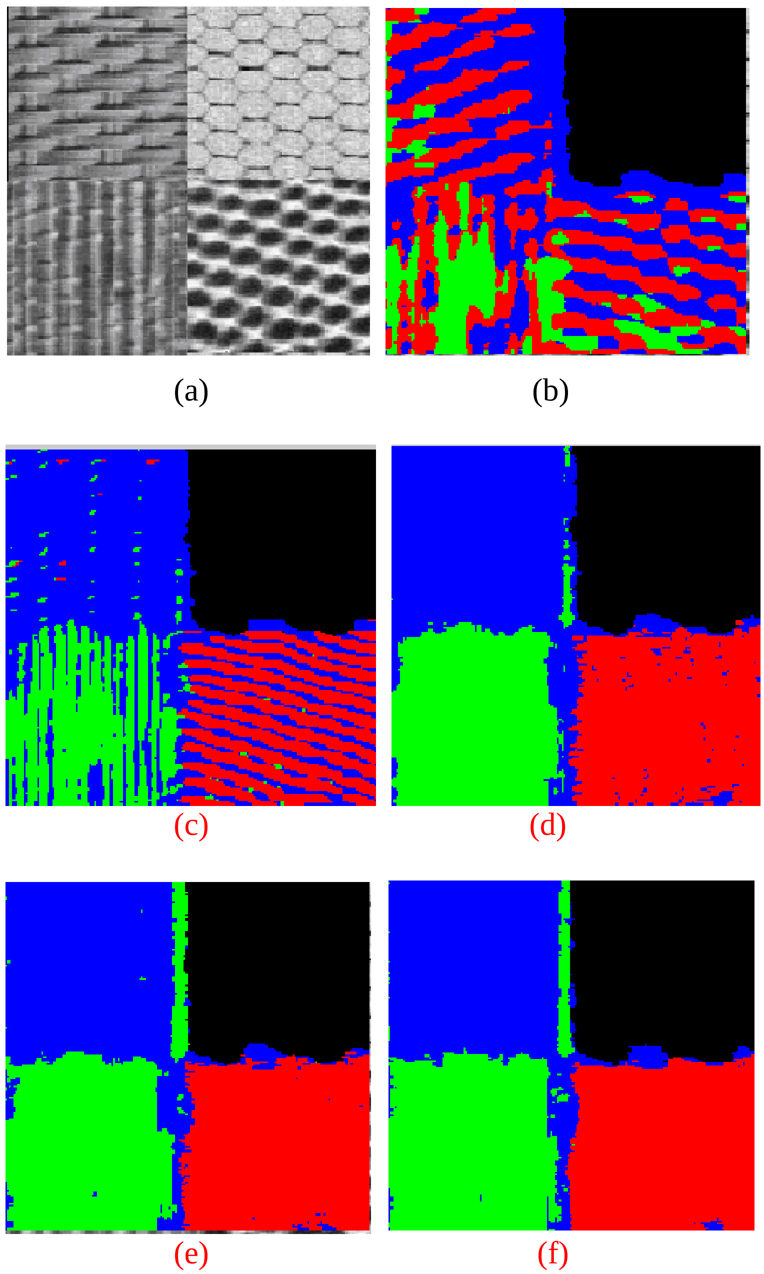


Figure 5.9
Classified results using the K-views template algorithm on a gray-scale texture image with four different textures. (a) an original image with the size of 200 x 200, (b), (c), (d), (e), and (f) are classified results obtained with the number of K-views from 1, 2, 8, 15, and 25, respectively.

5.4 Empirical Comparison with GLCM and Gaussian MRF (GMRF)

Spatial models are frequently used for image texture classification [1, 2, 3, 4, 5, 6, 11]. These models exploit the contextual information by utilizing spatial features for the classification. These spatial features capture the spatial relationships encoded in the image. As described in chapter 2, the Gray-Level Co-ocurrence Matrix (GLCM) is a statistical method which calculates properties of the relationship of pair-wise of pixels [5-6]. The spatial relationships between a pixel and its neighbors are recorded into Gray-level Co-occurrence Matrices (GLCM) and then used for calculating the statistics for features [5]. The statistics that produce independent features are preferred such as dissimilarity (D), entropy (E), and correlation (C) [5-6]. These features will be mapped into corresponding feature vectors, and the K-means and other clustering algorithms can be used to cluster these vectors.

Geostatistics has been used to measure the spatial properties. Car and Miranda [2] proposed a method which is based on the geostatistics (called the variogram), which is also a second-order statistics, to extract the texture features from the image. Unlike the GLCM, the variogram is to capture average gray level spatial dependence. The variogram can be computed with particular spatial directions. Four directions, E-W, N-S, NE-SW, NW-SE, are usually used for the spatial directions and statistics calculation [1]. Markov Random Field (MRF) models are stochastic processes which define the local spatial characteristics of an image. They are used to model the textural content of the observed image. The models characterize the statistical relationships between a pixel and its neighbors [3, 10].

Experimental results were performed on some of the Brodatz texture images to show the effectiveness of the models [11]. There are four different patterns of textures in images as shown in Figures 5.10 and 5.12 and five different patterns of texture image in Figure 5.11. In the experiments for the K-views-T model, the size of the K-views (i.e. characteristic view), the number of characteristic views and the statistics size (kernel size) are randomly selected for testing three different texture images. Although a variety of GLCM techniques are used in the literature, a simple GLCM is developed for the comparison. The classified results using the GLCM depend on the parameters such as distance, angle, and the number of gray-levels. In this experiment, the distance $\delta = 1$ and the angle $\alpha = 0, 45,$ 90, 135, 180 and 225 degrees were used. Since the gray-level of 256 causes some overhead, the image is quantized to 16 gray-levels. The experimental results are shown in Figures 5.10, 5.11 and 5.12.

Experimental results demonstrate that the K-views-T model is effective in the classification of an image texture. By increasing the size of K-view template and the number of K-views, it will improve the classification result in the K-views-T model. The K-views-T model has shown the significant improvement in the texture classification.

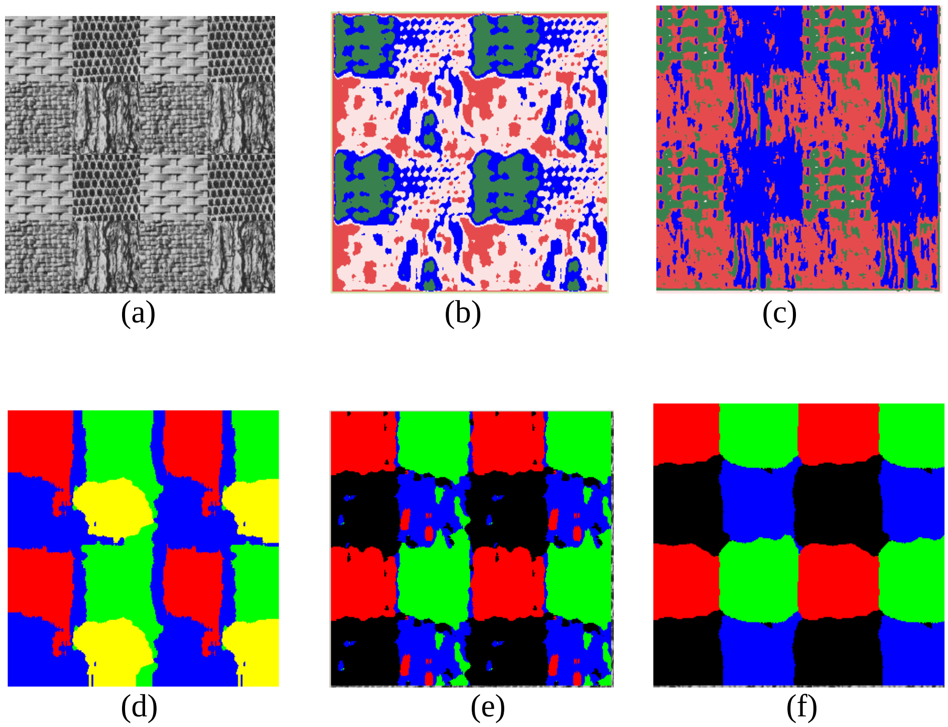
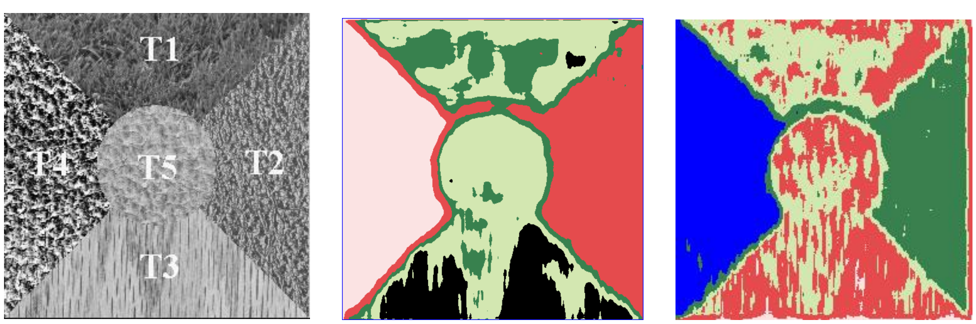
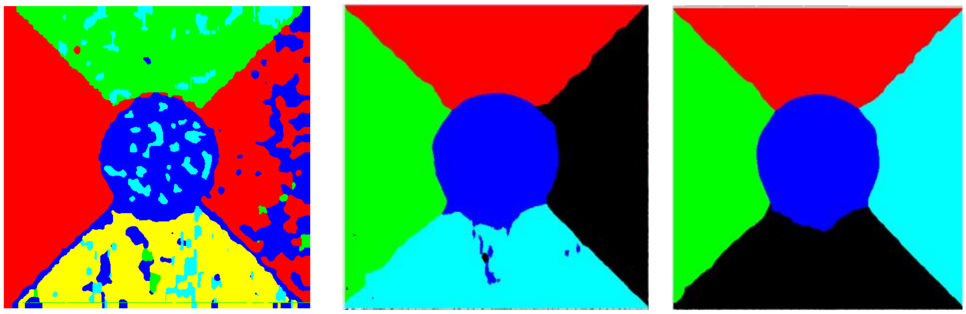


Figure 5.10

Classified results of an image texture using different spatial models. (a) an original image, b) the GLCM with the gray-level of 16, window size of 13, distance 1 and average of all directions, (c) the Variogram with the window size 9 and average of all directions, (d) the GMRF with the window size of 16 and the fourth-order neighborhood structure, (e) K-views-T results (the size of K-views is 10, the number of characteristic views K is 20, and the sample sub-image size M is 20), and (f) K-views-T results (the size of K-views is 10, the number of characteristic views K is 30, and the sample sub-image size M is 30) [11].



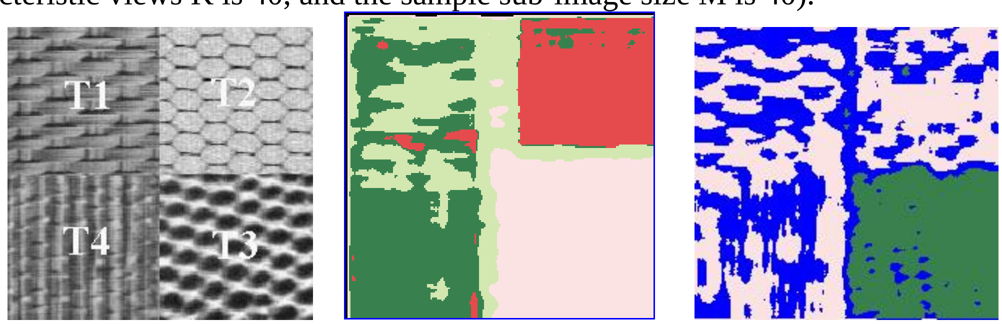
(a) (b) (c)



(d) (e) (f)

Figure 5.11

Classified results of an image texture using different spatial models. (a) an original image, b) the GLCM result with the gray-level of 16, window size of 32, distance 1 and average of all directions, (c) the Variogram with the window size 11 and average of all directions, (d) the GMRF with window size of 8 and fourth-order neighborhood structure, (e) K-views-T results (the size of K-views is 6, the number of characteristic views K is 20, and the sample sub-image size M is 30), and (f) K-views-T results (the size of K-views is 6, the number of characteristic views K is 40, and the sample sub-image size M is 40).



(a) (b) (c)

(d) (e) (f)

Figure 5.12

Classified results of an image texture using different spatial models. (a) an original image b) GLCM result with the gray-level of 16, window size of 32, distance 1 and average of all directions, (c) the Variogram with the window size 11 and average of all directions, (d) GMRF with the window size of 8 and four order neighborhood structure, (e) K-views-T results (with characteristic views are 4, the number of characteristic views k is 20, and the sample sub-image size M is 20), and (f) K-views results (with characteristic views are 4, the number of characteristic views k is 25, and the sample sub-image size M is 25).

5.5 Simplification of the K-views

The concept of K-views has illustrated the capability to distinguish different texture classes. The view set consisting of characteristic views is used to describe this relationship. The K-views template algorithm was developed for texture image classification based on the set of characteristic views.

If we simplify and reduce the size of K-views, some interesting results can be obtained. An K-views template with a size of 1 × 1 is equivalent to a pixel. Hence, a view with any size can be simplified as a line, a surface or represented by using the normal. If the view size is 1 × 2, and assuming that the set of characteristic views has a size of 256 * 256 for an 8-bit pixel in an image, a view distribution in the datagram is equivalent to a horizontal gray-level co-occurrence matrix (GLCM) with the distance of 1. Similarly, If the view size is 2 × 1, a view distribution in the datagram is equivalent to vertical co-occurrence matrix with the distance of 1. However, the K-views are derived based on a sample sub-image while the GLCM is derived based on a matrix with the arrangement of gray-levels.

If we are only interested in the two-end pixels of views with a size of 1 × n, the view can be denoted as a vector, $(x_1, x_2, \dots, x_n)^T$. The distribution in a datagram would be equivalent to the horizontal co-occurrence matrix with the distance of n-1. This similarity can also be found between the distribution in a datagram and other different co-occurrence matrices. In general, the view distribution in a datagram contains more messages because the information of all pixels are preserved. In the image texture classification literatures, the features of image textures can be described by surface, normal, and basic textural unit. Some of these are discussed in Chapter 2. Compared with those features, characteristic views keep the original messages and information for the classification.

5.6 Summary

The set of characteristic views is a powerful feature extraction method to describe an image texture. Different textures show different patterns. The patterns of a texture also show different views. If the size of K-views and the number of K-views in a set of characteristic views are properly defined, it is feasible to use a set of characteristic views for texture classification. The K-views template method (K-views-T) is an algorithm that uses a set of characteristic views for image texture classification. The K-views-T algorithm is suitable for classifying image textures that have basic local patterns repeated in a periodic manner.

The performance of K-views algorithm is related to the size of K-views template and the number of characteristic views (i.e. K-views) in each set. Increasing the view size and the number of characteristic views will generally improve the classification result at the expense of processing time. However, an issue remains to be explored in determining the relationship between the view size and the number of characteristic views in a set. This means that it is necessary to develop an algorithm which can automatically determine the view size and the number of characteristic views so that the K-views-T algorithm will have an optimal performance in terms of computation time and classification accuracy. The proposed classification method needs to interactively select the sample sub-images for each class. It would be useful to develop an unsupervised learning approach without human interaction by using view related features.

5.7 Exercises

For a numerical image shown below: assume that there are two different textures; one texture in the first four columns and the other in the remaining of the image.

0	1	2	3	4	5	6	3
1	2	3	0	5	6	7	6
2	3	0	1	5	4	7	7
3	0	1	2	4	6	5	6
3	2	1	0	4	5	6	3
2	3	2	3	6	5	5	4
1	2	3	0	4	5	6	7
3	0	2	1	7	6	4	5

1. Develop a set of views with a template size of 2 x 2 and 3 x 3.
2. Develop a set of characteristic K-views from Exercise #1 using the K-views-T algorithm.
3. Compare the performance of the K-views-T algorithm with different K values.
4. Implement the K-views-T algorithm using a high-level programming language and apply the algorithm to an image with different textures.

References

[1] Carr, J. R., Classification of Digital Image Texture Using Variograms, in Advances in Remote Sensing and GIS Analysis Edited by P. M. Atkinson and N. J. Tate, John Wiley & Sons, 1999, pp. 135–146.
[2] Carr, J. R. and F. P. Miranda, The semivariogram in comparison to the co-occurrence matrix for classification for image texture, IEEE Transactions on Geosciences and Remote Sensing 36 (1998) (6), pp. 1945–1952.

[3] Chellappa, R. and S. Chatterjee, "Classification of Textures Using Gaussian Markov Random Fields," IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-33, No. 4, pp. 959-963, Aug. 1985.
[4] Gurney, C. M. and J. R. G., Townshend (1983),"The Use of Contextual Information in the Classification of Remotely Sensed Data", Photogrammetric Engineering and Remote Sensing, Vol. 49, No. 1, 55-64, 1983.
[5] Haralick, R. M., K. Shanmugam, and I. Dinstein,"Textural Features for Image Classification", IEEE Transactions on systems, man, and cybernetics, vol. SMC-3, No. 6, 610-621, 1973.
[6] Hung, C. C., D. Karabudak, M. Pham and T. Coleman, "Experiments on Image Texture Classification with K-views Classifier, Markov Random Fields and Cooccurrence Probabilities," proceedings of the 2004 IEEE International Geoscience & Remote Sensing Symposium (IGARSS), Anchorage, Alaska, September 20-24, 2004.
[7] Hung, C.-C., S. Yang and C. Laymon, "Use of characteristic views in image classification," in proceedings of 16th International Conference on Pattern Recognition, pp.949-952, 2002.
[8] Lee, J. H. and W. D. Phipot, "Spectral texture pattern matching: a classifier for digital imagery," IEEE Transactions on Geoscience and Remote Sensing, Vol. 29, Issue 4, Jul 1991, PP. 545 – 554.
[9] Tou, J. T. and R. C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, 1974.
[10] Woods, J. W., "Two-dimensional discrete Markovian fields," IEEE Trans. Inform. Theory, Vol. IT-18, pp. 232-240, Mar. 1972.
[11] Xiang, M., C. C. Hung, M. Pham and T. Coleman, "Spatial Models for Image Texture Classification: An Experiment," proceedings of The 4th International Conference on Information and Management Sciences, July 1-10, 2005, Kunming, China, pp. 341-343, ISSN 1539-2023.
[12] Yang, S. and C.-C. Hung, "Image Texture Classification Using Datagrams and Characteristic Views," proceedings of the 18th ACM Symposium on Applied Computing (SAC), Melbourne, FL, March 9 - 12, 2003, pp. 22 – 26. doi>10.1145/952532.952538