Founded 1905

INVARIANT SEGMENTATION OF TEXTURE IN IMAGES OF NATURAL SCENE

BY

ZHANG NAN

BACHELOR OF SCIENCE

BEIJING COLLEGE OF ECONOMICS

1990

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF SCIENCE

(COMPUTER SCIENCE AND INFORMATION SYSTEMS)

NATIONAL UNIVERSITY OF SINGAPORE

1997

Acknowledgements

First, I would like to thank my thesis advisor, Dr. Leow Wee Kheng, for his guidance at the Intelligent Systems Laboratory in the past two years. Dr. Leow originated the research topic of invariant texture segmentation. He also actively organized a series of seminars which broadened my view in AI area. His patient and strong support will always be in my mind.

I would also like to thank Asso. Prof. Tan Chew Lim, Dr. Sung Kah Kay for their help on both teaching me and providing powerful workstations for my research.

My friends in the lab make me feel enjoyable working here. Discussion with Lai Seow Yeong and Wong Swee Seong, my research pals, is always a stimulating and comfortable thing. I also enjoy the communication with Yao Jun, Manoranjan Dash, Lu Ying, Huang Lilin, Ma Xiang, Shu Jian, Teo Ting Hui. My friends in Eusoff College always brings me endless happiness. I give my thanks to Ngan Wei Yeong, Lu Yuan, Wang Xiang Yu, Zhu Qi, Wang Rong, Lee Yen Hoe, Lu Fang, Gu Xin and a lot of warmhearted school mates.

Most of all, I would like to thank my loving parents. Although I'm thousands of miles away from you, I love you forever.

Contents

Summary

Texture segmentation is an important task which has been studied by many researchers. It is a very difficult task for images containing natural scenes. In these images, texture in a perceptually uniform region can change gradually in scale and orientation due to perspective distortion. On the other hand, neighboring regions with the same texture but sharp difference in scales or orientations are perceived as distinct regions. Most existing works on texture segmentation deal with only images containing *uniform texture*, i.e., texture that does not vary in scale and orientation. Typically, these images contain juxtaposed plane texture taken from the Brodatz album. The methods described in these works cannot correctly segment images containing non-uniform texture. There are also a number of recent methods on invariant classification of texture. However, these methods do not consider the rate of change of texture scale and orientation. Therefore, they are unable to segment images into regions consistent with human perception.

This thesis presents a novel method that performs perceptually consistent segmentation of images containing natural texture. The method can segment an image containing non-uniform texture into regions that are consistent with human perception. Such texture segmentation is indispensable in applications such as computer understanding of natural images, texture-based image retrieval, etc.

List of Figures

Chapter 1

Introduction

1.1 Motivation

Texture is characterized by a repeated pattern of light intensity reflected from object surfaces. It appears in many natural scenes such as grassland, buildings, desert and forest, and on many objects such as cloth, wire mesh, brickwalls and pebbles (Figure 1.1). Texture contains important information about the properties and structures of object surfaces. Analysis of texture can, therefore, aid in machine understanding of natural images and recognition of objects.

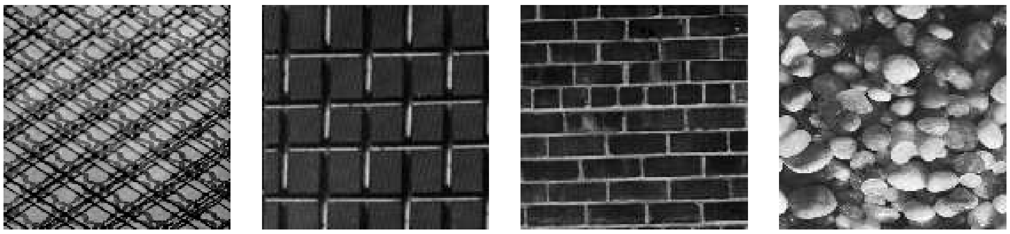


Figure 1.1: Some texture samples in the Brodatz album [4]. These images contain plane texture with uniform scale and orientation.

Texture segmentation aims at partitioning an image into regions each containing a uniform texture. It determines the boundaries between distinct textured regions. By performing texture segmentation, an image can be divided into uniformly textured regions such that each region corresponds to a distinct object.

Texture segmentation plays an important role in computer analysis and understanding of natural images. It is the first task of most image analysis or understanding processes. All subsequent tasks, such as shape determination, object recognition, and scene interpretation rely heavily on the quality of the segmentation results. Useful information can be obtained from texture segmentation:

1. Texture boundary. In recognizing objects in natural images, the objects should be first isolated from the background by identifying their boundaries and positions. In these images, different objects tend to have different texture surfaces and texture boundaries correspond closely to object boundaries. The application in [19] illustrates the use of texture segmentation technique to isolate the forest, urban areas and water areas in the LANSAT images. These areas are first segmented by their texture properties before they are classified into various categories. Thus, texture segmentation is an important part of object segmentation.

2. Texture feature. A set of texture features can be extracted in the segmentation process to represent each textured region. These features describe the distinct properties of different texture patterns. They can be used for classifying the texture patterns and for recognizing the objects that contain the texture. Thus, correct extraction of texture features will benefit object classification and recognition. Another typical application of texture feature is image retrieval [26]. Texture features are stored in the database together with the images. Given a query texture, images in the database

are compared with the query texture based on their features. Images with similar texture are then retrieved from the database.

Texture segmentation turns out to be a very difficult task for images of natural scenes. In these images, texture in a perceptually uniform region can change gradually in scale and orientation due to perspective distortion. Take Fig. 1.2 for example. The image contains several perceptually uniform regions each covered by one type of material: bricks or pebbles. Texture in a region can decrease gradually in scale with increasing viewing distance. The texture nearer to the viewer (bottom of image) appears larger than that further away (top of image). Orientation of the brick texture also changes gradually from the left of the image to the right. Texture boundaries are perceived at locations where the type of texture differs (e.g., between bricks and pebbles) or when the texture scale or orientation changes abruptly (e.g., between the top and the bottom brick patterns). A naive segmentation method may erroneously group image patches with the same texture but slowly varying scales and orientations into distinct regions, or neighboring patches with sharp changes in scales or orientations into the same region. To correctly segment the image, a texture segmentation algorithm has to take into account the rate of change of texture scale and orientation. So far, existing works on image segmentation have not considered this issue and are thus inappropriate for segmenting the image. This thesis presents a novel method for segmenting images of natural scene into perceptually uniform textured regions.

1.2 Background

Texture segmentation belongs to a class of texture analysis problems which includes three other subproblems: Texture classification (recognition), texture matching and shape from texture.

Figure 1.2: Image of floor covered with bricks and pebbles. Texture in a perceptually uniform region can vary gradually in scale and orientation. Texture boundaries are perceived at locations where the type of texture differs (e.g., between bricks and pebbles) or when the texture scale or orientation changes sharply (e.g., between the top and the bottom brick patterns).

Texture classification categorizes a given texture sample into a known texture class. For example, a particular region in an image may represent grass, brick, river or forest region. Each of these regions possesses a unique texture pattern. The texture classification algorithms extract the distinguishing features from the regions and use pattern recognition techniques to categorize the texture features. Implicit in the feature extraction process is the assumption that the boundaries between different regions have already been determined by a texture segmentation process.

Texture matching aims at searching for images containing a prescribed texture by comparing the texture with the textured regions in the images. A similarity measure close to human visual judgement is used in the comparison. To perform texture matching, the images must also be segmented so that each region can be represented by a unique set of texture features.

Shape from texture [1, 21] seeks to recover 3-D information from 2-D textured images. Variations in the size, shape and orientation of texture provide clues for estimating the surface shape and orientation. For example, texture gradient, defined as the magnitude and direction of maximum change of texture elements, determines the orientation of the surface. Shape from texture is also assumed to operate on homogeneous regions which are already segmented.

As discussed, texture segmentation is indispensable in texture analysis. Accurate segmentation results will greatly help to solve the other problems.

1.3 Problem Definition

The previous sections have presented an intuitive idea about texture segmentation. This section attempts to give a more precise definition to the problem. Let I denote an image and $T(R)$ be the texture type of a region R. Then, texture segmentation partitions the image I into disjoint non-empty subsets $R_1, R_2, ..., R_N$ such that:

(i) $\bigcup_{i=1}^{N} R_i = I$,

(ii) each R_i is composed of connected pixels,

(iii) texture in each region R_i is homogeneous, and

(iv) for any adjacent regions R_i and R_j, $T(R_i) \neq T(R_j)$.

The first condition specifies the completeness of segmentation: every pixel should be assigned to a region. The second condition requires that the regions be composed of connected pixels. The third one requires that each region contains a single type of texture. The forth condition says that neighboring regions should contain different types of texture; otherwise, they should be merged into a single region. Texture segmentation is well known as an ill-defined problem. The

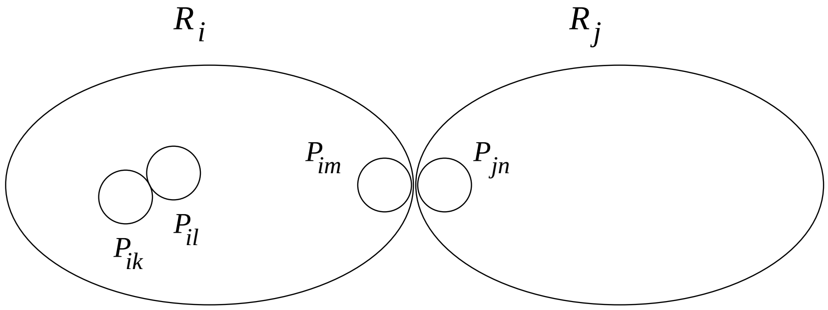


Figure 1.3: Definition of invariant texture segmentation

conditions listed above do not define a specific algorithm but only outline the important properties of a segmentation algorithm.

This thesis focuses on a specific texture segmentation which considers the gradual change of texture scale and orientation as explained in Section 1.1. An additional constraint must be added to item (iii) to elaborate the concept of homogeneity as follows. Referring to Fig. 1.3, let R_i and R_j be two arbitrary neighboring texture regions, P_{ik} and P_{il} be any neighboring texture patches inside region R_i, and P_{im} and P_{jn} be any adjacent texture patches within R_i and R_j respectively. Let S denote the similarity measure between two texture patches. Then, the texture segmentation algorithm should satisfy the condition:

$$S(P_{ik}, P_{il}) > S(P_{im}, P_{jn}) \tag{1.1}$$

This constraint requires that adjacent texture patches in the same region be more similar than those in different regions. Therefore, neighboring texture patches with slight difference will be regarded as belonging to the same region, whereas those with sharp difference will be classified into different regions. This constraint makes the segmentation more consistent with human's perception of the image.

1.4 Technical Considerations

Having discussed the problem definition, this section will focus on the technical considerations which are important in solving the problem. Three main points are discussed: (a) texture features, (b) localization of texture boundary, and (c) similarity measure.

1.4.1 Texture Features

Texture is characterized by the intensity distributions in a region instead of the grey level of a single pixel. Different distributions form different patterns which determine the texture type. To differentiate between different texture types, it is necessary to extract properties or features that describe the patterns. Therefore, texture segmentation is performed based on texture features rather than raw image pixels.

Texture features provide a unique quantitative representation of texture properties such as fineness, coarseness, smoothness, granularity and randomness. These properties are characterized by the texture pattern's spatial distribution and light intensity. In order to correctly describe the texture pattern and its type, texture features should include information about the pattern's intensity distributions.

As discussed in Section 1.1, the invariant texture segmentation should tolerate smooth changes in texture scale and orientation. Scale information is reflected in the spatial frequency of texture. Therefore, texture features should contain frequency and orientation information. In summary, a good set of features for invariant texture segmentation must well represent the type of the texture, its frequency and orientation.

1.4.2 Localization of Texture Boundary

Another crucial factor in texture segmentation is the localization problem. The purpose of segmentation is to identify the boundary of the texture regions. The invariant segmentation algorithm has to consider the gradual variation of texture scale and orientation. Fourier transform can be used to extract frequency and directionality information which correspond to scale and orientation. But the transform operates on the whole image instead of local regions. It provides only the global scale and orientation information of the whole image. A better way to extract the local scale and orientation of a region is to use windowed Fourier transform of which Gabor transform is a special case. Gabor transform has been proved to be optimally localized in both spatial and spatial frequency domains [7, 12]. By using such a transform, scale and orientation information in small regions can be extracted and texture boundaries can be localized more accurately.

1.4.3 Similarity Measure

Similarity measure is used to judge whether two texture features describe the same pattern. Texture pattern is not uniform in natural images. The same texture may have slightly different appearance due to scale and orientation changes or different lighting conditions. So, the similarity measure should take these variations into consideration. A normalized inner product (Cosine similarity) can be formulated to handle these problems (see Section 3.3 for details).

The above technical considerations outline the characteristics of a good texture segmentation algorithm which is more consistent with human perception. Details of the algorithm will be described in the following chapters.

1.5 Organization of the Thesis

The remainder of thesis is organized as follows: Chapter 2 discusses the existing works on texture segmentation, and compares and contrasts them. Chapter 3 presents the novel invariant texture segmentation method. The texture features and similarity measure used for invariant segmentation will also be discussed. Chapter 4 shows some test results on images of natural scenes. Chapter 5 discusses possible future enhancements of the algorithm. Finally, Chapter 6 concludes this thesis.

Chapter 2

Related Works

Existing methods of texture segmentation, and texture analysis in general, can be categorized based on the texture models that they used. These texture models include local statistics, texton, Markov Random Field (MRF), Fourier spectrum, and Gabor spectrum and wavelets.

2.1 Local Statistics

Early research by Julezs [17] suggested that first and second order statistics such as mean and moments may be sufficient for human visual system to discriminate between different types of textures. However, counter-examples have been found to disprove Julezs' suggestion [18]. More complex second-order statistics such as grey level cooccurrence matrices and higher order statistics such as covariance matrices are required.

A covariance matrix [24, 25] contains the covariances of every pixel in the image. Three eigenvalues are derived from the covariance matrix as local descriptors. They are interpreted as roughness measure, neighborhood structure and sampling density. The first two descriptors are invariant to rotation and translation. The means, variances, and third-order moments of the eigenvalues are computed and used as texture features. Although such texture features are orientation-invariant, they do not contain information about texture orientation and scale. Therefore, they lack the information for determining whether the change of scale and orientation is gradual or sharp.

Grey level cooccurrence matrices [13, 30] are second-order statistics of the intensities in an image. Each entry of the matrix is a joint conditional probability density function. It is the probability of going from one grey level to another grey level given the distance and the direction between two pixels. A lot of statistical features can be extracted from the matrices [14]. Among them, energy, entropy, correlation, local homogeneity, and inertia are usually used. Sampling along different directions and combining the results can produce the orientation-independent cooccurrence matrices. Some reports [31] showed that in computing the matrices, distances between pixels should be small in order to yield the best result. That is, they are not appropriate for describing coarse-scaled patterns. So, grey level cooccurrence matrices can carry orientation information but not scale information.

2.2 Texton Model

Texton refers to local conspicuous features, such as elongated blobs, rectangles, ellipses and line segments, with specific properties including color, angular orientation, width and length. Line terminators and line crossings are also considered as textons. Texture patterns are differentiated by the type of textons, their density and the positional relationship between them.

Most of the methods that use this model deal with artificial texture in which the textons are very obvious and easy to extract. There is no known method for extracting textons from natural images. In these images, noise and perspective distortion greatly complicate the detection of textons. Furthermore, some natural texture cannot be easily defined in terms of texton. Therefore, it is difficult to apply the texton model to segment natural images.

2.3 Markov Random Field Model

The Markov random field (MRF) models texture by a parameter estimation method [6, 20, 22, 28]. It assumes that texture is formed by a stationary, stochastic process. A pixel's intensity is given by a weighted sum of the intensities of neighboring pixels and a noise term. The weights are computed by a least squared method over the entire image. In this model, texture features correspond to the MRF weight parameters and feature extraction is equivalent to parameter estimation. A particular MRF model tends to favor a particular class of texture by associating them with larger probabilities than the others.

A typical model is Gibbs distribution model [27]. Subsets of pixels within the neighborhood of a pixel are defined. Weight coefficients are assigned to each pattern to indicate their relative importance in contributing to the overall characteristic of the texture. Computation of the weight parameters requires a laborious estimation process. Many estimation methods, e.g. maximum likelihood, linear least-square with histogramming, are applied. But they still sometimes generate unreliable or inconsistent results because they are performing linear optimizations on possibly non-linear segmentation problem. More complex model is required for images of natural scenes because they contain more than one textured regions and they are not uniformly illuminated.

It is very difficult for MRF model to be scale- and orientation-invariant since it would require a different set of weights for every possible scales and orientations, although some reports show that they can be orientation- and grey level-invariant for texture description [5].

2.4 Fourier Spectrum

As defined in Chapter 1, texture is a 2-D repetitive pattern characterized by its frequency and orientation distributions. Frequency and orientation properties can be extracted by performing 2-D Fourier transform on the image. The magnitudes of the resultant Fourier spectrum represent the amounts of energy in various frequency bands. Fourier spectrum gives the global frequency description on the whole image. It also carries the directionality information of the image and they have been used as texture features [11, 22].

Unfortunately, Fourier spectrum is poorly localized in the spatial domain. As we can observe in Figure 2.1, Fourier spectrum of a single texture has a regular distribution which can be distinguished by low order statistics. But for an image containing several texture patterns, its Fourier spectrum is equal to the superposition of the Fourier spectra of the individual patterns. It is impossible to isolate the Fourier spectrum of individual texture from the summed spectrum. It is also impossible to determine the locations of the texture patterns based on the summed spectrum.

2.5 Gabor and Wavelet Models

Windowed Fourier transform gives better localization in the spatial domain. Gabor filters [12] is a special case which simultaneously optimizes the spatial and spatial frequency localizations [7]. Gabor filter magnitudes have been widely used as texture features [2, 3, 8, 10, 16]. They are obtained by filtering the input image

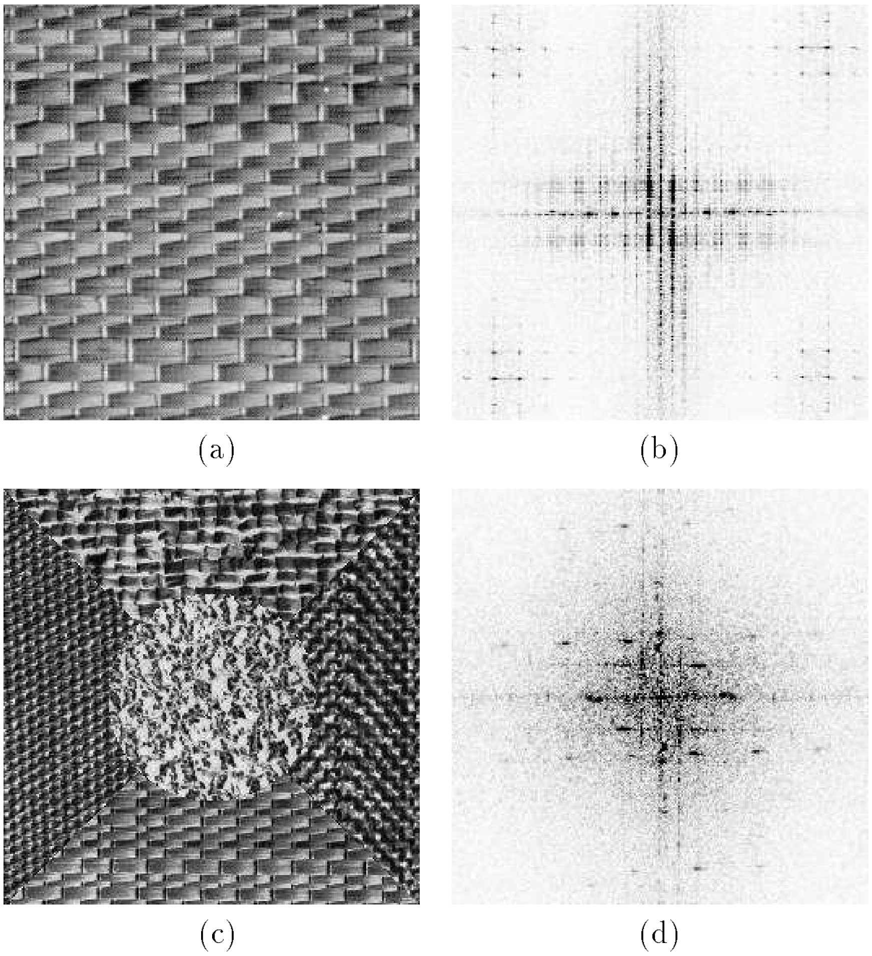


Figure 2.1: Comparison of Fourier spectrum of images with single texture pattern and multiple texture patterns. (a) This image contains a single texture (D55) taken from the Brodatz album. (b) The Fourier spectrum of (a). (c) This image contains the texture in (a) and four other texture patterns (D24, D77, D84, and D72 from Brodatz album). (d) The Fourier spectrum of (c) which is the superposition of the Fourier spectra of the individual texture patterns.

with a set of Gabor filters tuned to different spatial frequencies and orientations. The filters with matching frequencies and orientations will respond more strongly than other filter channels. Existing segmentation methods based on Gabor filters use only some of the filter channels, typically those with large magnitudes. The method described in this thesis, however, uses all the channels so as to recover slight variations of texture frequency and orientation (see Chapter 3 for details).

Wavelet functions are obtained by dilating and translating a basis function. The resultant functions are chosen to be orthorgnal and to decay rapidly towards zero. Like Gabor functions, they also decompose an image into various channels with different scales and orientations. The wavelet approach does not yield significantly better results compared to Gabor approach [23, 29]. It is not as optimal as Gabor functions in spatial and spatial frequency localization. Thus the position information carried by wavelets is not as accurate as that of Gabor functions.

2.6 Summary

A comprehensive list of references on texture segmentation can be found at the website *http://www.icv.ac.il/DataBases/biblio/bibliography/segment185.html*. Most existing works (e.g., [3, 6, 9, 10, 16, 22, 28, 33]) deal with only images containing *uniform texture*, i.e., texture that does not vary in scale and orientation within a perceptually uniform region. Typically, these images contain juxtaposed plane texture taken from the Brodatz album [4]. Segmentation of such an image is typically carried out by clustering the texture in the image so as to minimize intra-cluster distances and maximize inter-cluster distances. There are also a small number of methods dealing with natural scene images [22, 32]. However, all the above methods do not take into consideration gradual variation of texture scale and orientation, they cannot correctly segment images containing non-uniform texture. For example, in Fig. 11 of [22], the rock region is segmented into may small fragments instead of being grouped into a single region. Although multiresolution model is applied [29] to adapt to the scale change, the current methods still do not work if the scale change is non-isotropic.

There are also a number of recent methods on invariant classification of texture [6, 20, 24, 25, 33]. However, these methods also deal with images containing uniform texture, although the texture can have different scales and orientations in *different images*. The so-called invariant classification refers to correctly classifying texture that may differ in scales and orientations in different images. Since the images contain uniform texture, segmentation of these images is comparatively easier than for those containing non-uniform texture.

So far, we have not come across any other work that can truly perform scale- and orientation-invariant segmentation of images containing natural, non-uniform texture. By performing invariant texture segmentation, the method described in this paper can segment an image into regions that are more consistent with human perception.

Chapter 3

Invariant Texture Segmentation

This chapter describes how the Gabor filters are used to achieve the goal of invariant texture segmentation. Gabor functions will be presented first, followed by detailed discussion of the segmentation algorithm.

3.1 Gabor Function

The Gabor function $h(x, y)$ at image position (x, y) is a complex sinusoidal grating modulated by an oriented Gaussian function $g(x', y')$ [3]:

$$h(x, y) = g(x', y') \exp[2\pi j(Ux + Vy)] \tag{3.1}$$

$$g(x', y') = \frac{1}{2\pi\lambda\sigma^2} \exp\left[-\frac{(x'/\lambda)^2 + y'^2}{2\sigma^2}\right] \tag{3.2}$$

where $(x', y') = (x\cos\phi + y\sin\phi, -x\sin\phi + y\cos\phi)$ are rotated coordinates oriented at angle ϕ from the x-axis, λ is the aspect ratio, and σ is the scale parameter. The complex sinusoid has radial frequency f and orientation θ:

$$\begin{aligned} f &= \sqrt{U^2 + V^2} \\ \theta &= \tan^{-1}\left(\frac{V}{U}\right) \end{aligned} \quad . \tag{3.3}$$

where U and V are the spatial frequencies along the x- and y-directions. The sinusoid's orientation θ is typically chosen to be the same as the Gaussian's orientation ϕ. In this case, Eq. 3.1 becomes:

$$h(x,y) = g(x',y') \exp(2\pi j f x') \ . \tag{3.4}$$

The properties of Gabor filters can be more easily understood in the spatial frequency domain. Gabor function (Eq. 3.4) has spatial frequency response

$$H(u,v) = \exp\{-2\pi^2\sigma^2[(u'-f)^2\lambda^2 + v'^2]\} \tag{3.5}$$

where $(u',v') = (u\cos\theta + v\sin\theta, -u\sin\theta + v\cos\theta)$. $H(u,v)$ is a Gaussian function rotated by an angle θ from the u-axis and shifted by an amount f along the rotated u'-axis. Therefore, Gabor function is a bandpass filter with radial central frequency f and orientation θ. Its frequency (octave) bandwidth B and orientation (radian) bandwidth Ω at half-peak are [3]:

$$\begin{aligned} B &= \log_2\left[\frac{\pi f\lambda\sigma + \alpha}{\pi f\lambda\sigma - \alpha}\right] \\ \Omega &= 2\tan^{-1}\left(\frac{\alpha}{\pi f\sigma}\right) \end{aligned} \tag{3.6}$$

where $\alpha = \sqrt{(\ln 2)/2}$. The range of spatial frequencies within the frequency and orientation bandwidths is called the *half-peak support* (Fig. 3.1).

3.2 Gabor Amplitude as Texture Features

A multiple channel filter consists of a set of Gabor filters with appropriate frequencies f and orientations θ. Each channel has a real (cosine) and an imaginary (sine) component (Fig 3.2):

$$\begin{aligned} h_{c,f\theta}(x,y) &= g(x',y')\cos(2\pi f x') \\ h_{s,f\theta}(x,y) &= g(x',y')\sin(2\pi f x') \end{aligned} \ . \tag{3.7}$$

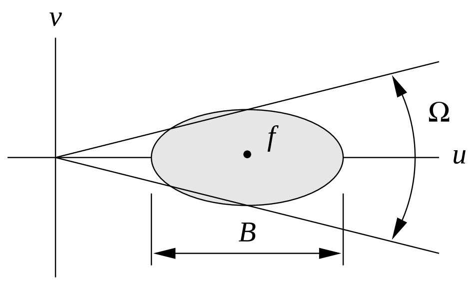


Figure 3.1: Half-peak support of Gabor function (shaded area) is the range of spatial frequencies within the frequency bandwidth B and orientation bandwidth Ω.

The corresponding spatial frequency response is

$$\begin{aligned} H_{c,f\theta}(u,v) &= \frac{1}{2}\exp\{-2\pi^2\sigma^2[(u'-f)^2\lambda^2+v']\}+\frac{1}{2}\exp\{-2\pi^2\sigma^2[(u'+f)^2\lambda^2+v']\} \\ H_{s,f\theta}(u,v) &= -\frac{j}{2}\exp\{-2\pi^2\sigma^2[(u'-f)^2\lambda^2+v']\}+\frac{j}{2}\exp\{-2\pi^2\sigma^2[(u'+f)^2\lambda^2+v']\} \end{aligned} \quad . \tag{3.8}$$

Typically, the filters are chosen such that they tile the spatial frequency plane as closely as possible with minimum overlap of their half-peak supports. That is, the difference in central frequencies Δf between neighboring filters equals spatial frequency bandwidth B, and the difference in orientations $\Delta\theta$ equals orientation bandwidth Ω (Fig. 3.3). This way, the spatial frequency plane can be covered by a small number of Gabor filters with varying spatial frequencies and orientations.

Given an input image $I(x,y)$, texture features in the image are extracted by first convolving the image with the set of Gabor filters:

$$\begin{aligned} k_{c,f\theta}(x,y) &= h_{c,f\theta}(x,y) * I(x,y) \\ k_{s,f\theta}(x,y) &= h_{s,f\theta}(x,y) * I(x,y) \end{aligned} \quad . \tag{3.9}$$

Figure 3.4 shows the Gabor filters' response to image D21 of the Brodatz album. The filter channels tuned to the direction and scale of the texture have strong response while the others have weak response.

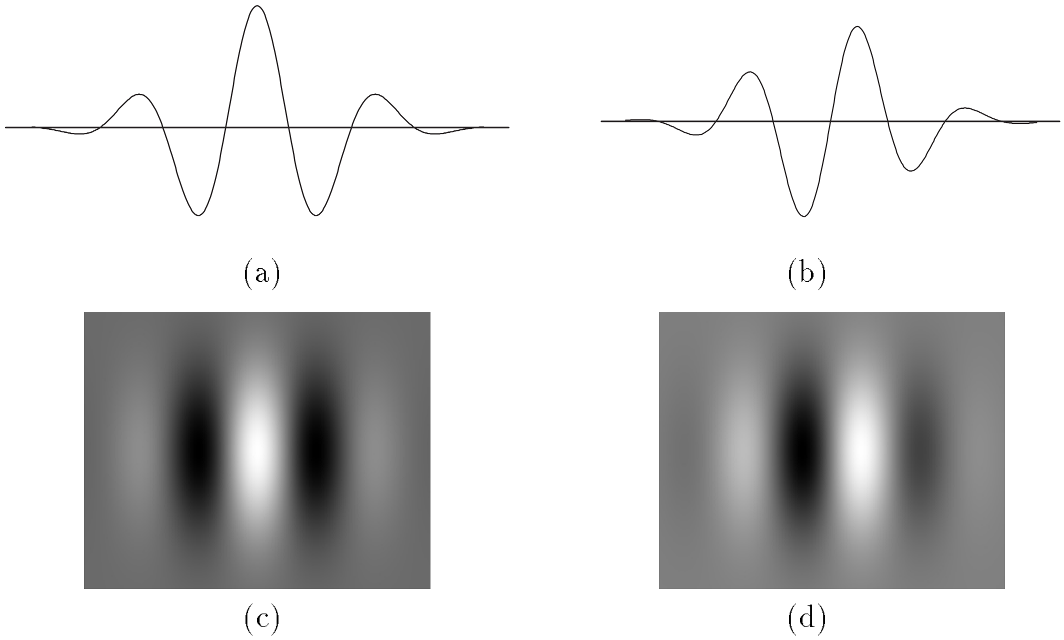


Figure 3.2: The Gabor function. (a) The sideview and (c) the topview of the real (cosine) part. (b) The sideview and (d) the topview of the imaginary (sine) part.

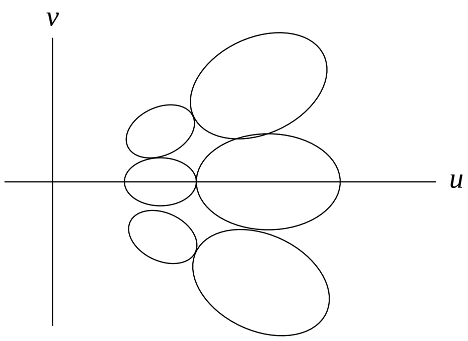


Figure 3.3: Gabor filter placement with minimum overlaps. Filters are chosen such that their half-peak supports have minimum overlaps.

The amplitudes of the filters' outputs

$$k_{f\theta}(x,y) = \sqrt{k^2_{c,f\theta}(x,y) + k^2_{s,f\theta}(x,y)} \tag{3.10}$$

form the *texture feature vector* $\mathbf{k}(x,y)$ in the f-θ space (Fig. 3.5). In typical applications, $\mathbf{k}(x,y)$ is low-passed filtered to remove variations introduced by the sinusoidal terms of the Gabor functions.

Texture features obtained in this way can be used for invariant texture matching and segmentation. To see how invariant matching can be accomplished, let us consider, for simplicity, a simple texture $t(x,y)$ given by the sum of sinusoids:

$$\begin{aligned} t(x,y) &= 2\sum_i a_i \cos[2\pi(U_i x + V_i y)] \\ &= 2\sum_i a_i \cos(2\pi f_i x') \end{aligned} \tag{3.11}$$

where the a_i's are constant amplitudes and $f_i = \sqrt{U_i^2 + V_i^2}$. This texture has the following Fourier transform:

$$T(u,v) = \sum_i a_i[\delta(u' - f_i, v') + \delta(u' + f_i, v')] \tag{3.12}$$

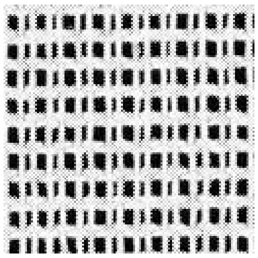
where $\delta(a,b) = 1$ if $a = b = 0$, and $\delta(a,b) = 0$ otherwise.

In response to this texture, the filters' outputs in the spatial frequency domain are given by

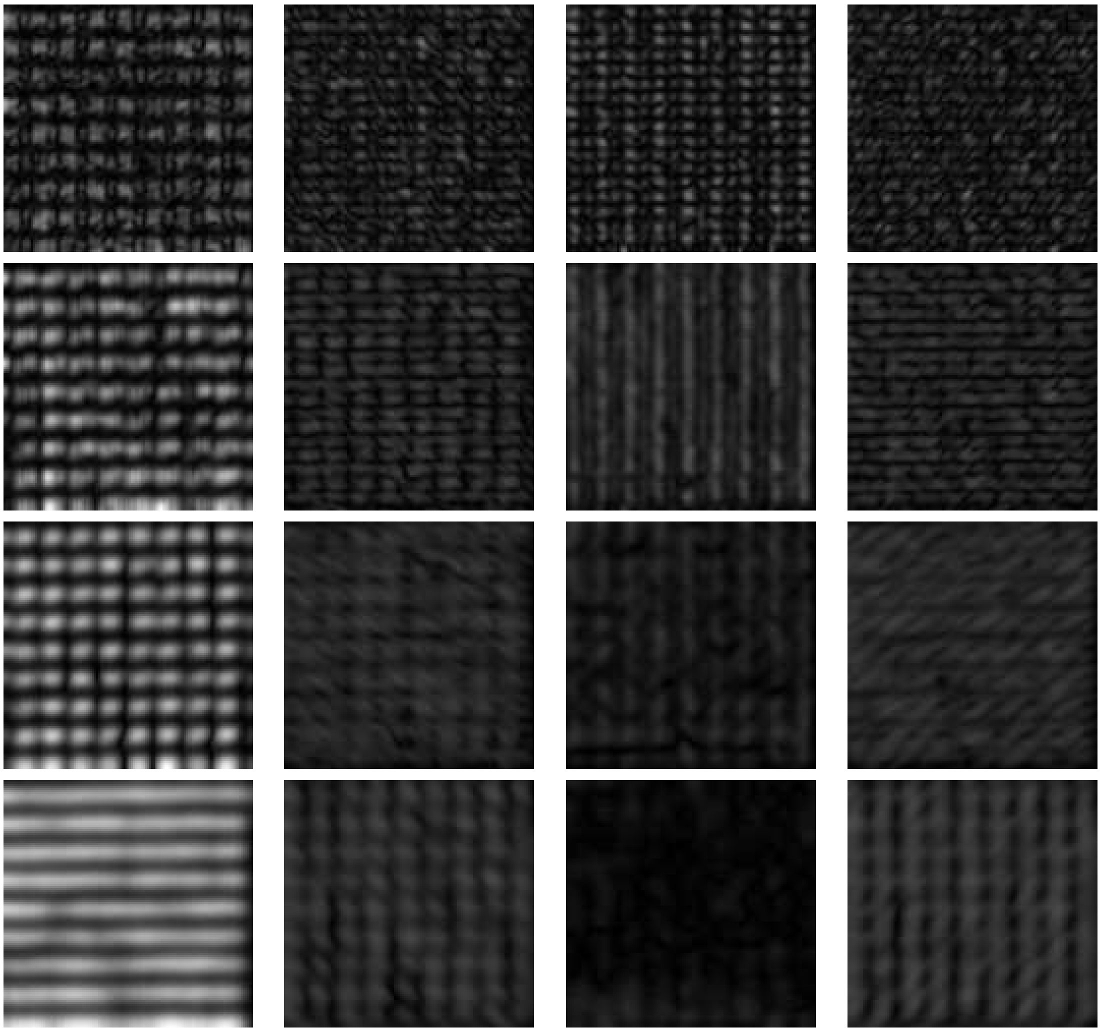
$$\begin{aligned} K_{c,f\theta}(u,v) &= H_{c,f\theta}(u,v)\ T(u,v) \\ K_{s,f\theta}(u,v) &= H_{s,f\theta}(u,v)\ T(u,v) \end{aligned} . \tag{3.13}$$

Assuming that the Gabor filters have small bandwidths and they are more closely packed than the spacing between the f_i's of the texture, then Eq. 3.13 becomes

$$\begin{aligned} K_{c,f\theta}(u,v) &= \tfrac{1}{2}\sum_i a_i[\delta(u' - f_i, v') + \delta(u' + f_i, v')]\,\delta(f - f_i, \theta - \theta_i) \\ K_{s,f\theta}(u,v) &= \frac{j}{2}\sum_i a_i[-\delta(u' - f_i, v') + \delta(u' + f_i, v')]\,\delta(f - f_i, \theta - \theta_i) \end{aligned} . \tag{3.14}$$



(a)



(b)

Figure 3.4: Gabor filters' responses to texture. (a) Texture D21 of the Brodatz album. (b) Each sub-image corresponds to the response of a particular Gabor channel. The channel's orientations are 0°, 45°, 90° and 135° from left to right, and the frequencies range from 0.3 circles/pixel and down sampling to 0.08 circles/pixel from top to bottom.

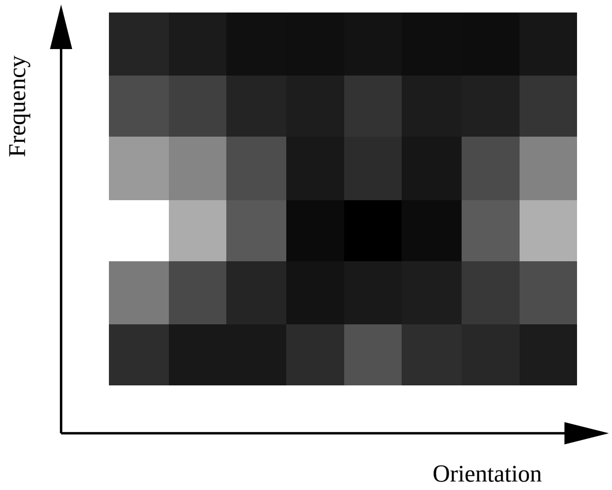


Figure 3.5: The 2-D representation of the Gabor channels' outputs at a pixel location. 48 filters are used with 8 orientations and 6 frequencies. Bright squares represent high intensity. The horizontal direction represents orientations from 0° to 157.5° and vertical direction represents frequencies from 0.08 to 0.3 cycle/pixel (see details in Section 3.3).

That is, the filters whose frequencies f and orientations θ match any of the texture's component frequencies f_i and orientations θ_i will have non-zero outputs; the others will have zero outputs. Thus,

$$\begin{aligned} k_{c,f\theta}(x,y) &= \sum_i a_i \cos(2\pi f_i x')\, \delta(f - f_i, \theta - \theta_i) \\ k_{s,f\theta}(x,y) &= \sum_i a_i \sin(2\pi f_i x')\, \delta(f - f_i, \theta - \theta_i) \end{aligned} \tag{3.15}$$

and the texture features as given by the filters' output amplitudes are

$$k_{f\theta}(x,y) = \sum_i a_i\, \delta(f - f_i, \theta - \theta_i) \tag{3.16}$$

Consider now an isotropically scaled version $t'(x,y)$ of the texture $t(x,y)$:

$$t'(x,y) = 2\sum_i a_i \cos[2\pi(U_i' x + V_i' y)] \tag{3.17}$$

where $U_i' = SU_i$ and $V_i' = SV_i$ for some constant scaling factor S. Take note that the orientation of the texture remains unchanged, i.e., $\theta_i' = \tan^{-1}(V'/U') = \theta_i$. Gabor filters' response to this texture is

$$k_{f\theta}(x, y) = \sum_i a_i\, \delta(f - f_i', \theta - \theta_i) \tag{3.18}$$

where $f_i' = \sqrt{U_i'^2 + V_i'^2}$. That is, the pattern of non-zero outputs remains unchanged except for a shift of $f_i' - f_i$ along the f-axis in the f-θ space.

Similar invariant property is preserved by a change in orientation. Consider a rotated version $t'(x, y)$ of the texture $t(x, y)$:

$$t'(x, y) = 2 \sum_i a_i \cos[2\pi(U_i'x + V_i'y)] \tag{3.19}$$

where $(U', V') = (U \cos\theta + V \sin\theta, -U \sin\theta + V \cos\theta)$. Notice that, in this case, the frequency of the texture remains unchanged, i.e., $f_i' = \sqrt{U_i'^2 + V_i'^2} = f_i$. Gabor filters' response to this texture is

$$k_{f\theta}(x, y) = \sum_i a_i\, \delta(f - f_i, \theta - \theta_i') \tag{3.20}$$

where $\theta_i' = \tan^{-1}(V'/U')$. In other words, the pattern of non-zero outputs remains unchanged except for a shift of $\theta_i' - \theta_i$ along the θ-axis in the f-θ space.

Figure 3.6 illustrates how the feature vector changes with texture scale and orientation. The first row in the figure displays the original image and those rotated by 45°, 90°, 135° respectively. The second row shows that the feature patterns shifts horizontally along the orientation axis. The third row shows the patterns at 60%, 80%, 100%, and 120% of the original texture. The forth row indicates that the corresponding feature vectors shift vertically along the frequency axis.

In images of natural scenes, texture is more complex than that given by Eq. 3.11. Moreover, practical applications typically use Gabor filters with considerably larger bandwidths. Nevertheless, empirical results show that, for small

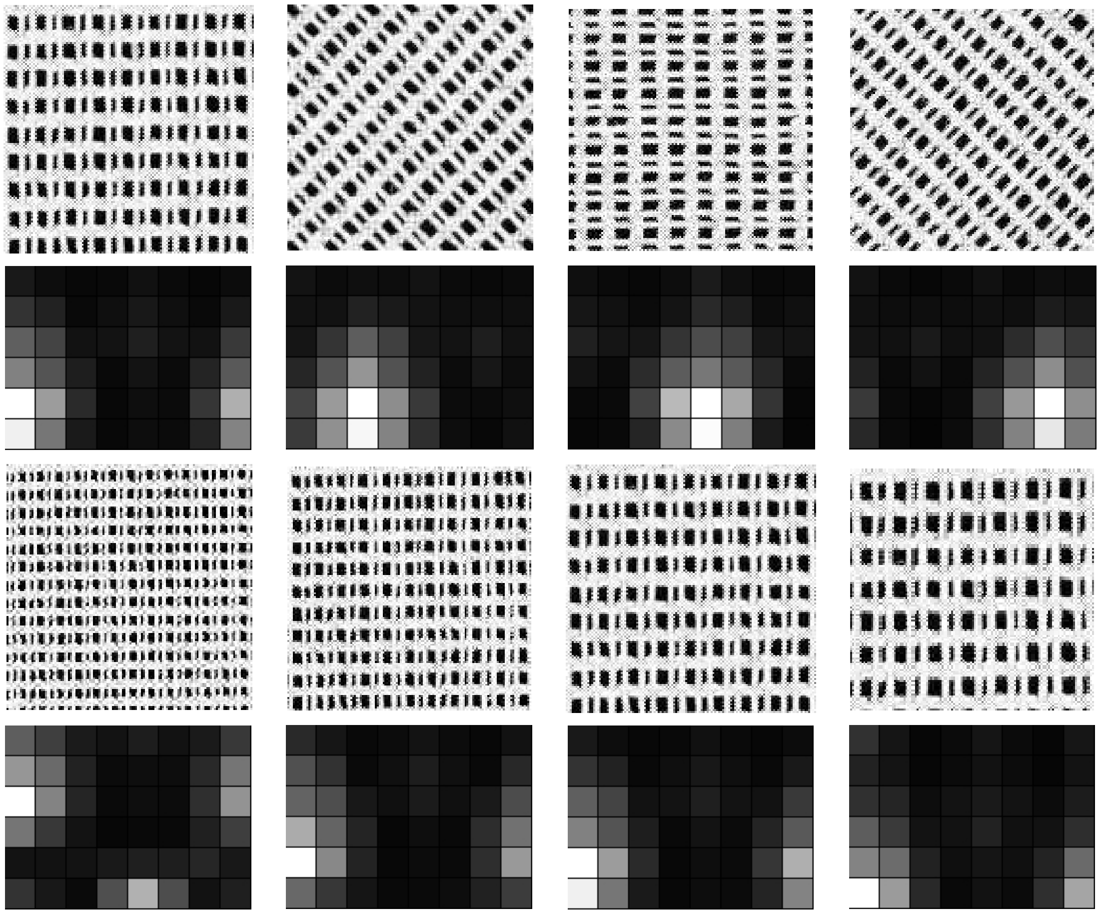


Figure 3.6: Feature pattern shifts with orientation and scale change. The first row displays the original image and those rotated by 45°, 90°, 135°. The second row shows the feature pattern shifts horizontally along the orientation axis. The third and the forth rows illustrate the images at 60%, 80%, 100%, and 120% scaling of the original texture and the vertical shifts of the feature pattern along the frequency axis.

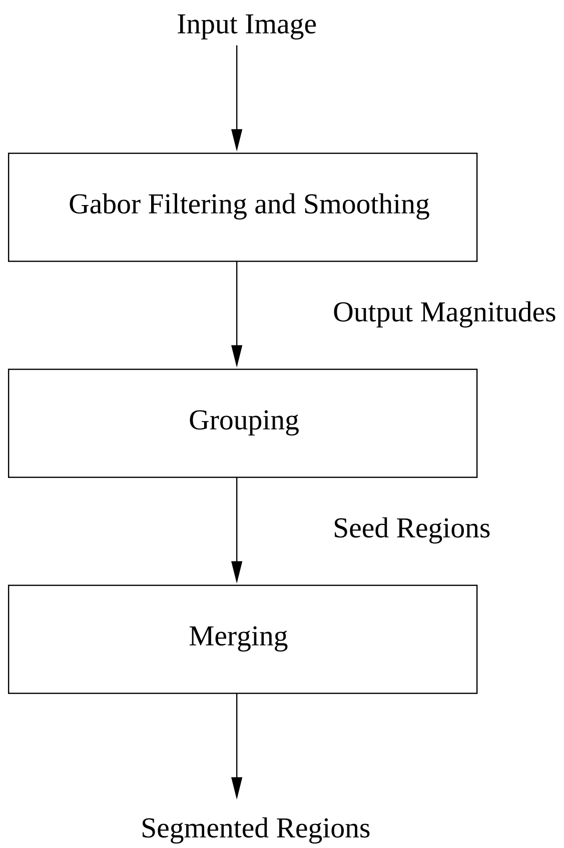


Figure 3.7: The invariant texture segmentation consists of 3 main stages: Gabor filtering and smoothing, grouping, and merging.

changes in scale and orientation, the above invariance properties are largely preserved (see details in Section 3.3). The next section describes how Gabor filters are used in the invariant segmentation algorithm.

3.3 Invariant Segmentation Method

The texture segmentation algorithm consists of 3 main stages (Fig. 3.7): Gabor filtering and smoothing, grouping, and merging. The first stage computes the smoothed amplitudes of Gabor filters' outputs, which are used by the next two

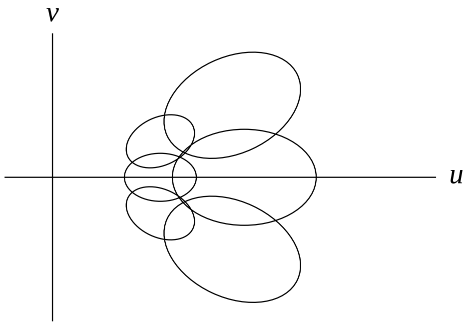


Figure 3.8: Gabor filter placement with overlaping half-peak supports. Filters are chosen such that their half-peak supports have some overlaps.

stages to group pixels with similar texture into the same region.

3.3.1 Gabor Filtering and Gaussian Smoothing

In the Gabor filtering stage, the input image is filtered by a set of Gabor filters (Eq. 3.7) with different frequency and orientation selections. Unlike existing methods, the filters are chosen such that there are some overlaps in the filters' half-peak supports (for reasons given below, Fig. 3.8):

$$\begin{aligned} f_i &= \frac{f_m}{2^{i\beta B}} \quad . \\ \theta_i &= i\beta\Omega \end{aligned} \tag{3.21}$$

The parameter f_m is the maximum spatial frequency. It is set at 0.3 cycle/pixel since higher frequencies are beyond the resolution of the discrete Gabor filters. Extremely fine texture region with very high frequency is almost equivalent to a plane surface. The parameter β determines the amount of overlap in the filters' supports. With $\beta = 1$, the filters will have minimum overlap. With $\beta = 0$, all filters overlap completely. It is determined experimentally that $\beta = 0.5$ provides a suitable amount of overlap. The parameters B and Ω are the spatial frequency and orientation half-peak bandwidths. With $B = 0.75$ octave, 6 frequency bands can cover frequency range of 3 octaves, which are enough for texture in most natural images. With $\Omega = 45°$, 8 orientation bands are enough to cover range from $0°$ to $180°$. It is possible to have smaller bandwidthes but it would increase the number of filters required to cover the same amount of Fourier space. Therefore, $B = 0.75$ octave and $\Omega = 45°$ provide large enough bandwidth to cover substantial amounts of Fourier space without sacrificing spatial frquency and orientation resolutions. The above parameter setting results in a total of 48 filters, with 6 spatial frequencies and 8 orientations.

In contrast to existing methods which use only some of the Gabor channels' outputs, the invariant segmentation method described here uses the *pattern* of outputs from all the channels as the texture feature. This method of representing texture feature is known as *distributed representation* in the Neural Network literatures [15]. Figure 3.9 illustrates examples of how texture can be represented by the pattern of Gabor outputs. The feature in Fig. 3.9(a) has 2 specific orientations of $0°$ and $90°$, each with a different scale or spatial frequency. Its feature vector shows 2 localized high response regions at corresponding spatial frequencies and orientations. The one in Fig. 3.9(b) has a predominantly horizontal orientation. The thick stripes contribute to low frequency components while the fine details in each stripe contribute to high frequency components. So, the texture's feature vector shows high response in all the channels at the $0°$ orientation. The pebble texture in Fig. 3.9(c) has a specific scale but no specific orientations. As a result, its high response channels spread over many orientation bands but restricted to specific frequency bands.

The overlapping of filters' supports ensures that the output pattern will not be severely altered by slight changes in texture's spatial frequency and orientation. Figures 3.10 and 3.11 illustrate the intuitive idea in the spatial frequency domain. In Fig. 3.10, top row shows the spatial frequency of a hypothetical input

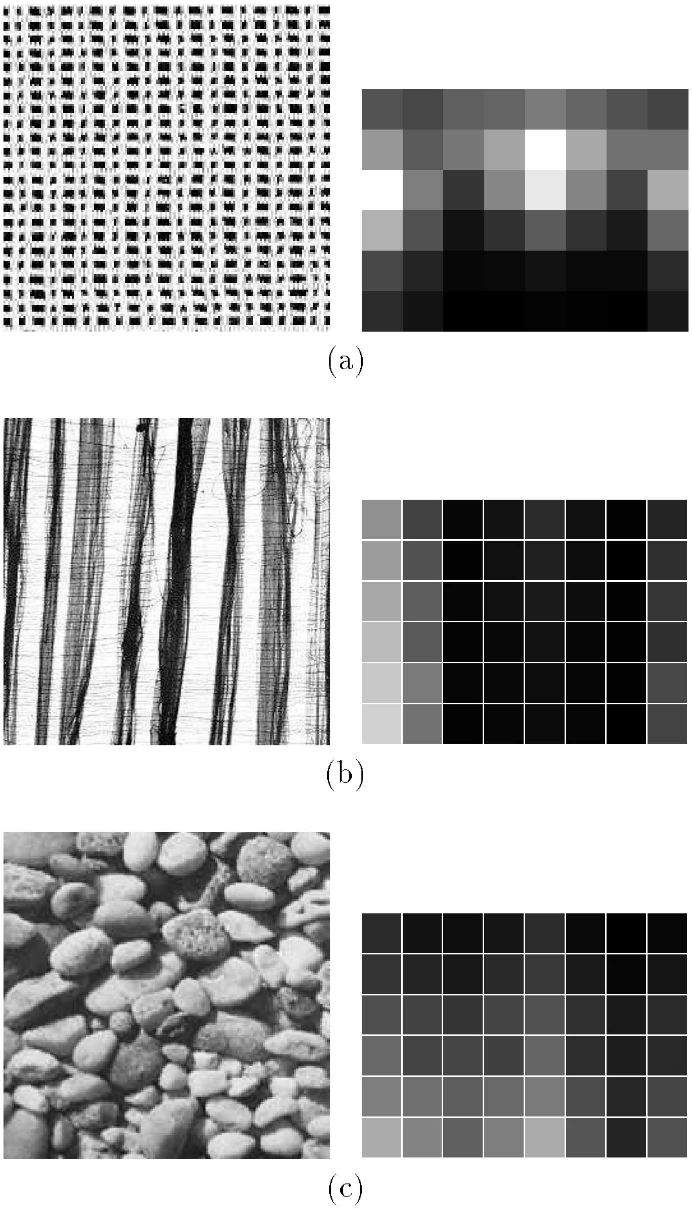


Figure 3.9: Distributed representation of texture feature. (a) Texture with specific scale and orientation. (b) Texture with specific orientation and many scales. (c) Texture with specific scale and many orientations

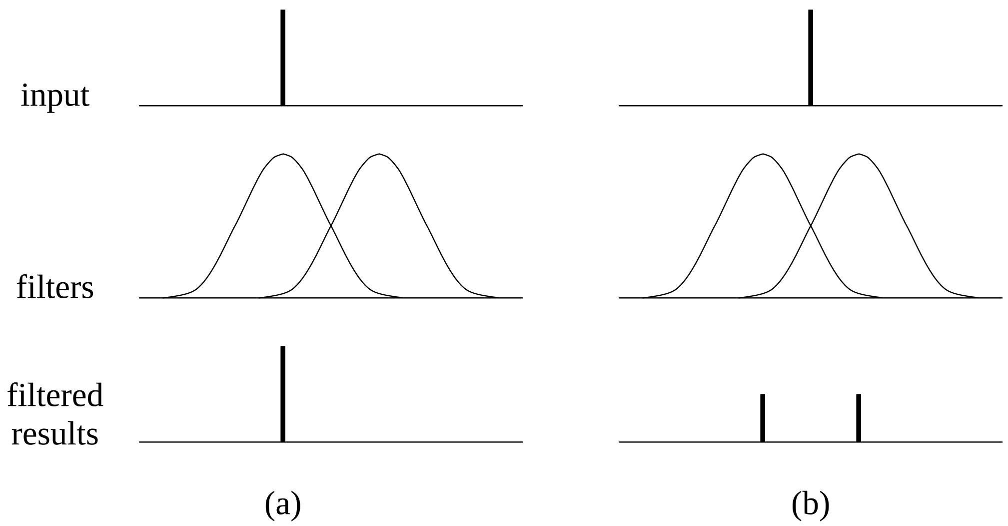


Figure 3.10: Response of filters to variation of spatial frequency. Top row shows the spatial frequency of a hypothetical input texture. Middle row shows the spatial frequency response of two filters whose half-peak supports do not overlap. Bottom row shows that, as the input texture's frequency changes from (a) to (b), the filters' output pattern changes drastically.

texture. Middle row shows the spatial frequency response of two filters whose half-peak supports do not overlap. Bottom row shows the results of filtering the input texture, which is equivalent to multiplying the input and the filters' frequency response. As the input frequency changes from (a) to (b), the filters' response pattern changes drastically. Figure 3.11 illustrates the case where the filters supports overlap slightly. Compared to Fig. 3.10, the filters' output pattern changes less drastically as the input frequency varies. Closer similarity between the output patterns facilitates the grouping of these texture inputs into the same region. In addition, more texture types can be encoded by distributed representation than single channel representation encoding. With single channel encoding, each channel represents a specific texture type. On the other hand, with distributed representation, a texture type is represented by a combination

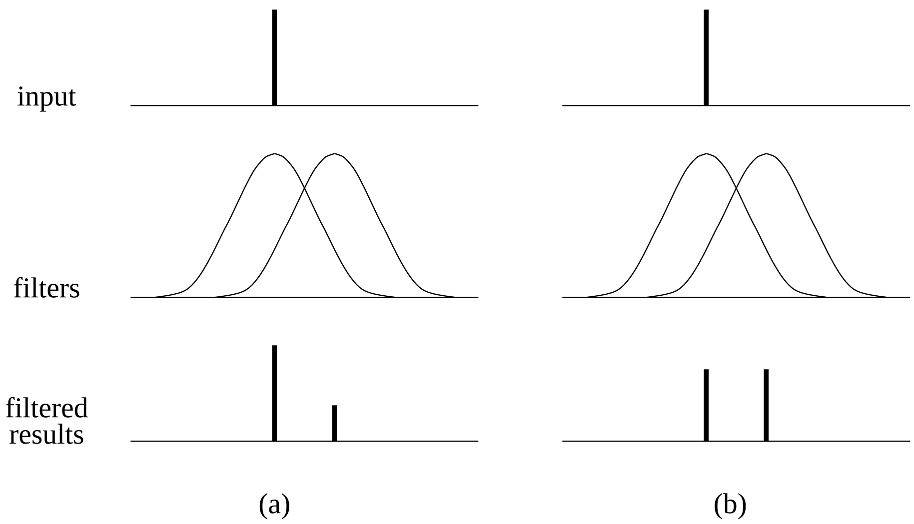


Figure 3.11: Response of filters to variation of spatial frequency with overlapped support. Top row shows the spatial frequency of a hypothetical input texture. Middle row shows the spatial frequency response of two filters whose half-peak supports overlap slightly. Bottom row shows the filtered results. Compared to Fig. 3.10, the filters' response pattern changes less drastically.

of channels.

A nonlinear transform $\phi(k)$, suggested by Jain [16], is performed on all the channels' outputs to enhance the channel response and to reduce noise:

$$\phi(k) = \tan(\alpha k) = \frac{1 - \exp^{-2\alpha k}}{1 + \exp^{-2\alpha k}} \tag{3.22}$$

where α is a constant and k is the Gabor amplitude. In the current implementation, $\alpha = 0.25$ is used which results in a rapidly saturating, threshold-like transformation.

In the smoothing stage, the filtered results are smoothed by Gaussian filters to remove local variations introduced by the sinusoidal terms in the Gabor functions. The Gaussian functions used have the same aspect ratios and orientations as the Gabor functions but larger scale parameters σ. Gaussian weighted smoothing templates are likely to result in more accurate localization of texture boundaries than other kinds of smoothing functions. Currently, the Gaussian scale parameters are set at four times those of the corresponding Gabors. After smoothing, the set of 48 Gabor output amplitudes at each pixel location forms the texture feature vector at that position (Fig. 3.5).

3.3.2 Grouping

In the grouping stage, similar feature vectors at neighboring locations are grouped into *seed regions*. This operation reduces the number of feature vectors that will be involved in the next stage. Each seed region R_i is characterized by a *peak* vector $\mathbf{P}_i$ and a *mean* vector $\mathbf{M}_i$. Initially, one seed region R_1 is created which contains the feature vector $\mathbf{k}(0,0)$ at location $(0,0)$, and $\mathbf{P}_1$ and $\mathbf{M}_1$ are set to $\mathbf{k}(0,0)$. Subsequently, a feature vector $\mathbf{k}(x,y)$ is grouped into region R_i if

- there is a vector $\mathbf{k}'(x',y')$ in R_i that is a 4-neighbor of $\mathbf{k}(x,y)$ in the x-y space,
- similarity $C(\mathbf{k},\mathbf{P}_i) > \Gamma_P$, and
- similarity $C(\mathbf{k},\mathbf{M}_i) > \Gamma_M$,

where $\Gamma_P = \cos 5°$ and $\Gamma_M = \cos 3°$ are constant thresholds, and C is the cosine similarity:

$$C(\mathbf{k}_1,\mathbf{k}_2) = \frac{\mathbf{k}_1 \cdot \mathbf{k}_2}{\|\mathbf{k}_1\| \; \|\mathbf{k}_2\|} \ . \tag{3.23}$$

If a vector can be grouped into more than one region, then one of the regions is arbitrarily chosen to contain the vector. After grouping $\mathbf{k}$ into region R_i, peak vector $\mathbf{P}_i$ and mean vector $\mathbf{M}_i$ are updated as follows:

$$\begin{aligned} \mathbf{P}_i &= \max_{\mathbf{k}_j \in R_i} \mathbf{k}_j \\ \mathbf{M}_i &= \frac{1}{n} \sum_{\mathbf{k}_j \in R_i} \mathbf{k}_j \end{aligned} \ . \tag{3.24}$$

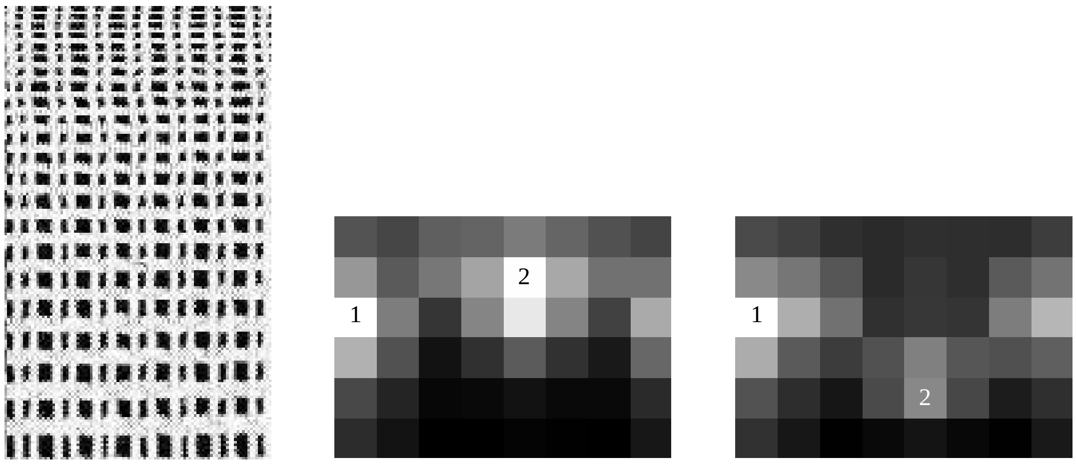
where n is the number of vectors in R_i.

Pixels with very small Gabor amplitudes do not contain significant information and are regarded as plain region without texture. Currently, pixels with Gabor amplitudes smaller than 8% of the image's highest magnitude are ignored.

3.3.3 Merging

After grouping, seed regions are merged in the merging stage into larger regions that are more consistent with human's perception of the input image. Two regions that share a common boundary are merged if their feature vectors are similar enough. The similarity measure used in this stage is slightly more complex because it has to take into account gradual variations of texture scale and orientation.

Section 3.2 shows that a rotation or an isotropic scaling of a texture pattern results in only a shift of the corresponding Gabor feature pattern in the f-θ space. In an image of natural scene, however, changes in scale may not be isotropic. As a result, the shifting of Gabor output pattern is more complex. Figure 3.12 illustrates a simple example in which a grid texture is vertical frequency is gradually increased but its horizontal frequency remains unchanged. Fig. 3.12(b) and (c) show the magnitudes of the feature vectors at the top-middle and bottom-middle regions of the image. There are two local peaks in each texture pattern. The right peak (at $90°$ orientation) is shifted vertically because vertical frequency is changed, but the left peak's position remains unchanged. In the case of perspective distortion, the shifting can be more complex. Therefore, to compare the similarity between two feature patterns, it is necessary to consider individual shifting of local peaks. Existing methods of scale-invariant classifications [20, 6, 24, 33, 25] consider only isotropic scaling of texture and are, therefore, inappropriate for processing texture with perspective distortion.



(a) (b) (c)

Figure 3.12: Vector change according to the scale change. (a) Input image whose vertical frequency is gradually increased. (b) Feature vector at the top-middle region, with spatial frequency increasing from the bottom up and orientation increasing from the left to the right. (c) Feature vector at the bottom-middle region.

The similarity measure between feature vectors are defined as follows. Suppose that feature vector $\mathbf{k}$ has a peak at (f, θ). Form a *local peak vector* by taking the feature vector components $k_{f'\theta'}$ in the 3×3 neighborhood of (f, θ). Now, consider two feature vectors $\mathbf{k}_1$ and $\mathbf{k}_2$ each having a set of local peak vectors $\{\mathbf{p}_i\}$ and $\{\mathbf{q}_j\}$. The *invariant similarity* $S(\mathbf{k}_1, \mathbf{k}_2)$ between $\mathbf{k}_1$ and $\mathbf{k}_2$ is defined as

$$S(\mathbf{k}_1, \mathbf{k}_2) = \tfrac{1}{2}[\zeta(\mathbf{k}_1, \mathbf{k}_2) + \zeta(\mathbf{k}_2, \mathbf{k}_1)] \tag{3.25}$$

where $\zeta(\mathbf{k}_1, \mathbf{k}_2)$ (and, similarly, $\zeta(\mathbf{k}_2, \mathbf{k}_1)$) is given as follows: For each $\mathbf{p}_i$ of $\mathbf{k}_1$, find the matching $\hat{\mathbf{q}}_i$ such that

$$C_d(\mathbf{p}_i, \hat{\mathbf{q}}_i) = \max_l C_d(\mathbf{p}_i, \mathbf{q}_l) \; . \tag{3.26}$$

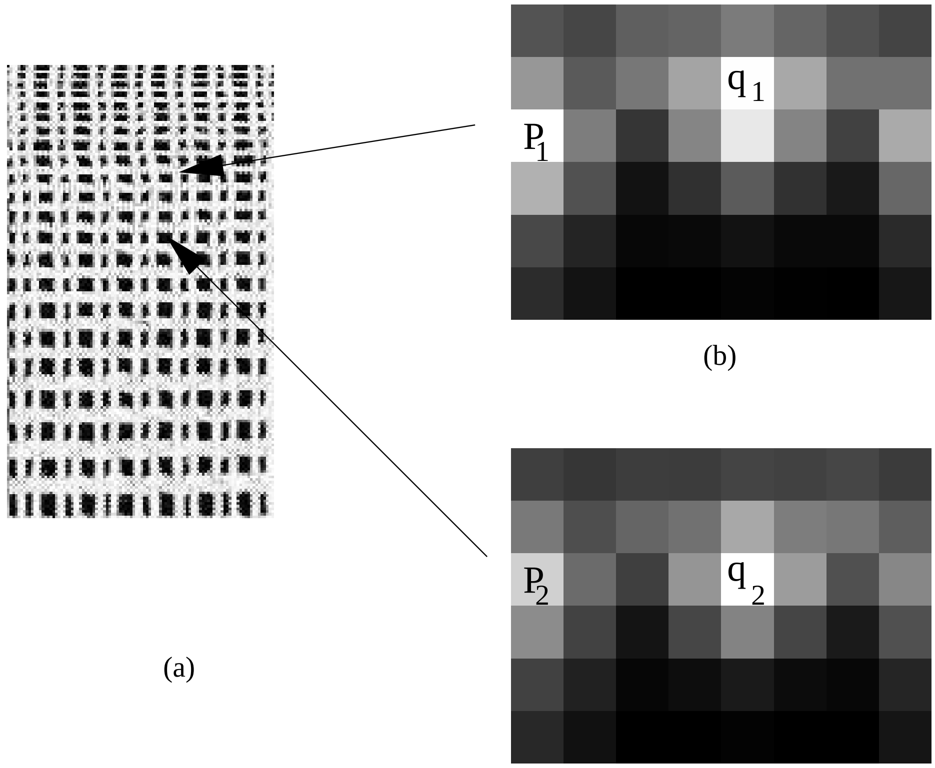
The similarity measure C_d is a distance-weighted vector dot-product:

$$C_d(\mathbf{p}, \mathbf{q}) = \frac{w(\mathbf{p}, \mathbf{q})\, \mathbf{p} \cdot \mathbf{q}}{\|\mathbf{p}\| \; \|\mathbf{q}\|} \tag{3.27}$$

where $w(\mathbf{p}, \mathbf{q})$ is a constant weighting factor that decreases with increasing Euclidean distance between $\mathbf{p}$ and $\mathbf{q}$ in the f-θ space. In other words, Eq. 3.26 finds the $\hat{\mathbf{q}}_i$ that is most similar and nearest to $\mathbf{p}_i$ in f-θ space (e.g., Fig. 3.13). Then, $\zeta(\mathbf{k}_1, \mathbf{k}_2)$ is computed as a normalized, weighted vector dot-product of all the local peak vectors $\mathbf{p}_i$ of $\mathbf{k}_1$ and their matching $\hat{\mathbf{q}}_i$ of $\mathbf{k}_2$:

$$\zeta(\mathbf{k}_1, \mathbf{k}_2) = \frac{\sum_i w(\mathbf{p}_i, \hat{\mathbf{q}}_i)\, \mathbf{p}_i \cdot \hat{\mathbf{q}}_i}{\sqrt{\left(\sum_i \|\mathbf{p}_i\|^2\right) \left(\sum_i \|\hat{\mathbf{q}}_i\|^2\right)}} \; . \tag{3.28}$$

In the beginning of the merging process, each (seed) region R_i is characterized by its mean feature vector $\mathbf{M}_i$. As two regions are merged into one, both their mean vectors are collected in the merged region. It is necessary to collect the mean vectors into the merged region instead of averaging them because, as a region gets larger in size, the mean vectors at its extreme ends may differ significantly even



(a)

(b)

(c)

Figure 3.13: Invariant matching of neighboring feature vectors. (b) and (c) are neighboring feature vectors in feature image (a). In comparison, the local peak $\mathbf{q}_2$ of vector (c) is shifted vertically relative to $\mathbf{q}_1$ of (b); whereas $\mathbf{p}_2$ of (c) remains at the same position as $\mathbf{p}_1$ of (b). To obtain a good match, local region at $\mathbf{q}_2$ must be shifted to the position of $\mathbf{q}_1$.

though the mean vectors in the region vary gradually over the entire region (e.g., Fig. 3.12). Two neighboring regions can be merged if

- they share part of their boundaries, and
- they are similar enough: $S(\mathbf{M}_1, \mathbf{M}_2) > \Gamma_S$

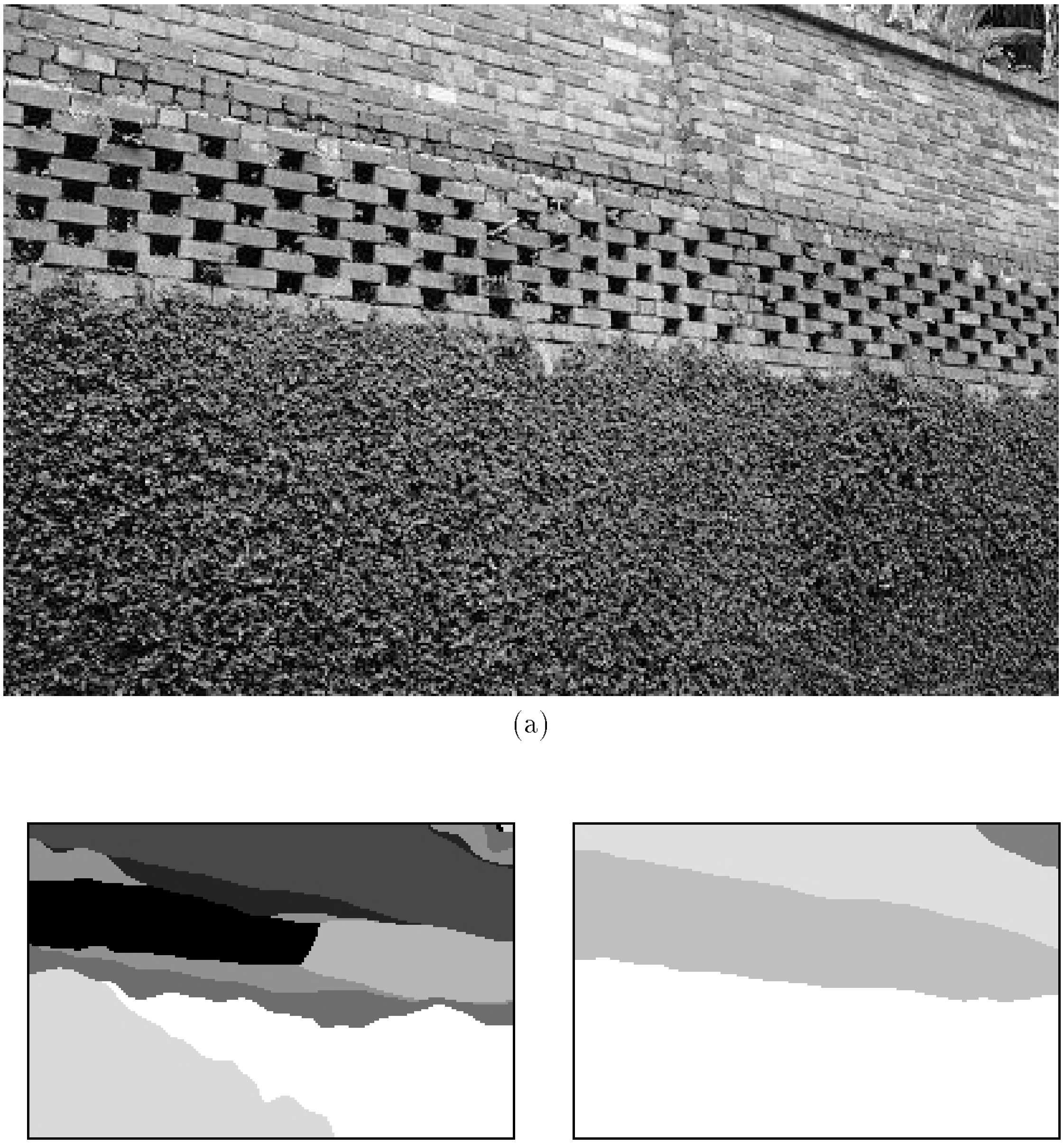
where $\mathbf{M}_1(x, y)$ and $\mathbf{M}_2(x, y)$ are the regions' mean vectors that are nearest to each other in space, and Γ_S is a constant threshold currently set to $\cos 23°$. At the end of the merging stage, the result is cleaned-up by merging very small regions into neighboring larger regions that are most similar in texture. These small regions are those that cover less than 1.5 percent of the whole image area.

Chapter 4

Test Results

This section illustrates some results of applying the invariant texture segmentation method on images of natural scenes. Figure 4.1 compares the performance of the method, which is scale- and orientation-invariant, with a typical non-invariant method. The non-invariant method used here is derived from the invariant method by removing the key feature that makes the algorithm scale- and orientation-invariant, i.e., by replacing the invariant similarity measure S by a simpler cosine similarity C. The non-invariant method tends to split a perceptually uniform region into several fragments when there are significant differences between the texture scales or orientations (Fig. 4.1b). In contrast, the invariant method produces results that are more consistent with human perception (Fig. 4.1c).

Figure 4.2 illstrates another comparison between invariant and non-invariant methods. Due to perspective distortion, the texture on the metal cover in Fig. 4.2(a) changes gradually from very coarse scales to very fine scales. The non-invariant method segments the metal cover into several fragments each containing a texture of approximately the same scale (Fig. 4.2b). In contrast, the invariant method segments the cover into a single region (Fig. 4.2c). It also



(a) (b) (c)

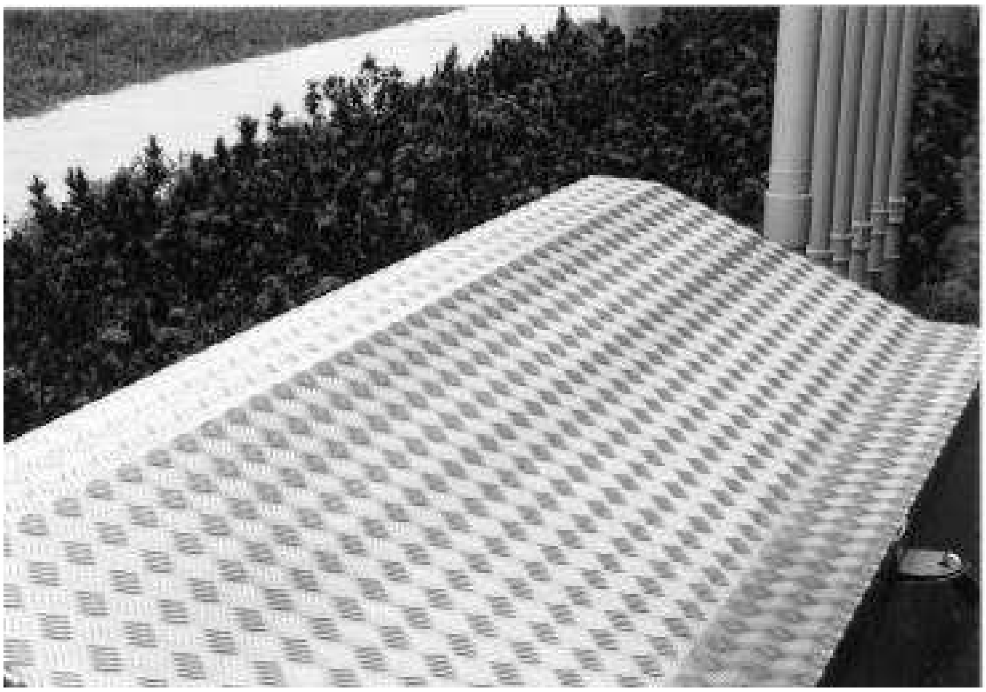
Figure 4.1: (a) Image of a wall covered with leaves and two types of brick texture. (b) Non-invariant segmentation method splits a perceptually uniform region into several fragments (painted in different shades of gray). (c) Invariant segmentation method produces regions that are more consistent with human perception. (The input image is shown at twice the size of the results so that fine texture can be clearly perceived by the readers.)

segments other regions in the image well.

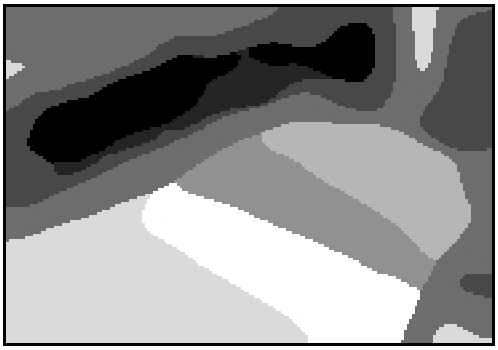
The next example illustrates that the invariant segmentation method takes into account the rate of change of texture scale and orientation. Figure 4.3(a) shows an image of ground covered with bricks and pebbles. The two brick patterns at the top and the bottom are of the same type except for a rotation of 90°. The invariant method correctly identifies the texture boundary indicated by the sharp change in orientation, and yet successfully groups patches with the same texture but gradually varying scales and orientations into the same region.

Figures 4.4 and 4.5 show two segmentation results of images containing rectangular bricks, square tiles, and metal mesh. In both examples, the segmentation results are consistent with human perception. The vertical mesh in Fig. 4.5(a) is split into two regions (Fig. 4.5b) because the texture pattern shows up as vertical strips in the top part and horizontal holes in the bottom part.

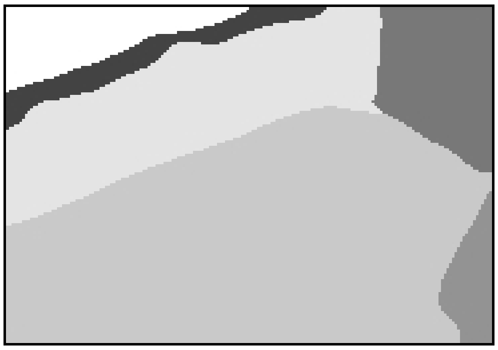
The next example illustrates the segmentation of boundaries formed by irregular texture, such as grass and rocks, which has rather ill-defined orientation. Figure 4.6 shows that the invariant segmentation method can identify the curvilinear boundary between the grass lawn and the rocks. In the final example, Fig. 4.7(a) illustrates an image of a complex scene containing a building sitting behind two clumps of plants. The building has three ventilation windows and its wall is covered with bricks. Figure 4.7(b) shows that the walls, windows and plants are segmented into different regions. The segmentation result is quite consistent with human perception of the scene.



(a)



(b)



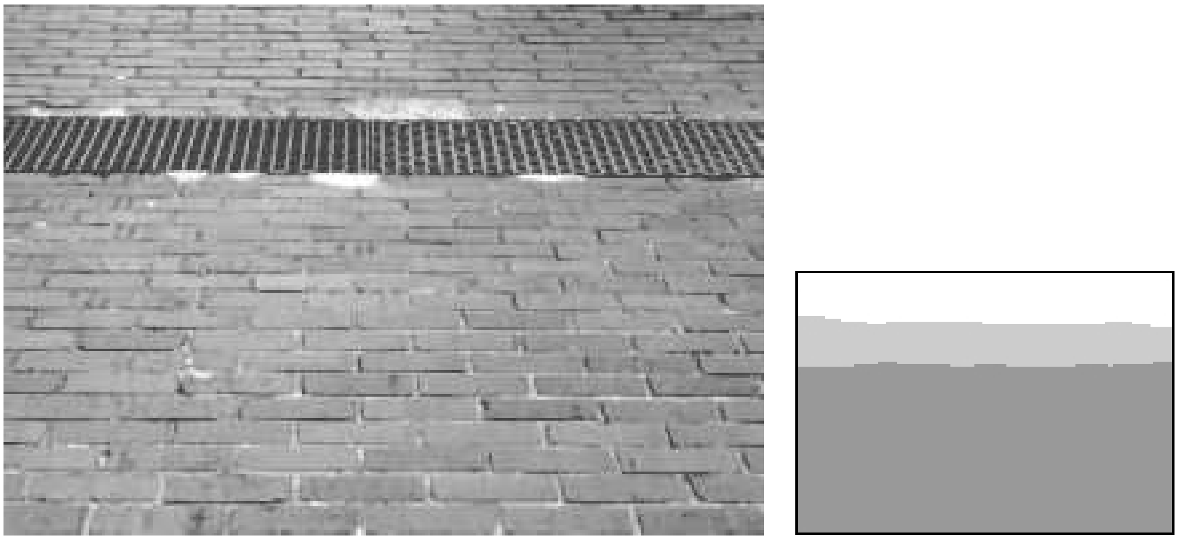
(c)

Figure 4.2: (a) An image of a complex scene. (b) Non-invariant segmentation method splits the metal cover into several fragments according to the scale of the texture. (c) Invariant segmentation method identifies the patterns over the entire cover as the same texture despite gradual change in scale.

(a)

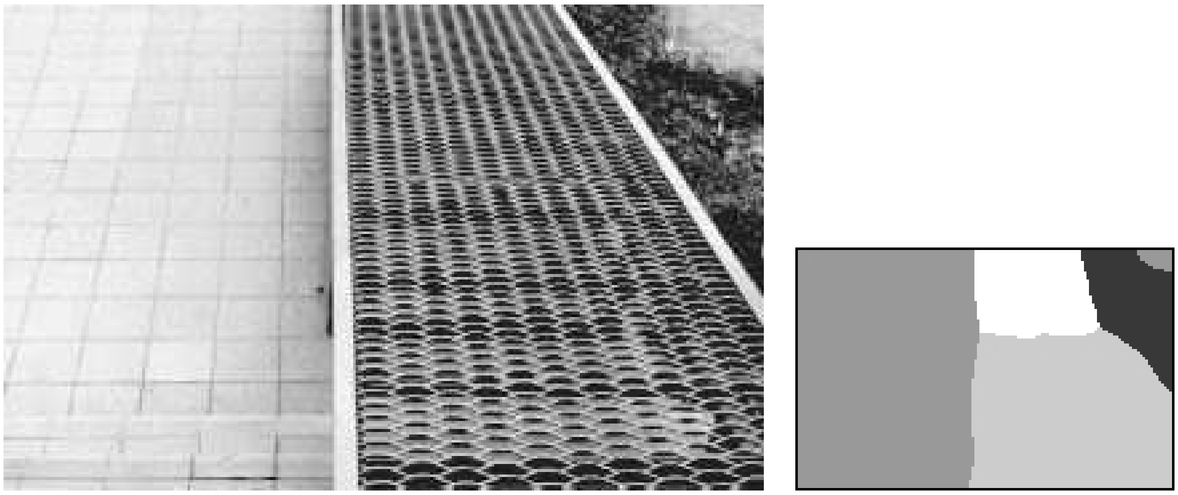
(b)

Figure 4.3: (a) Image of a floor covered with bricks and pebbles. (b) Invariant segmentation result.



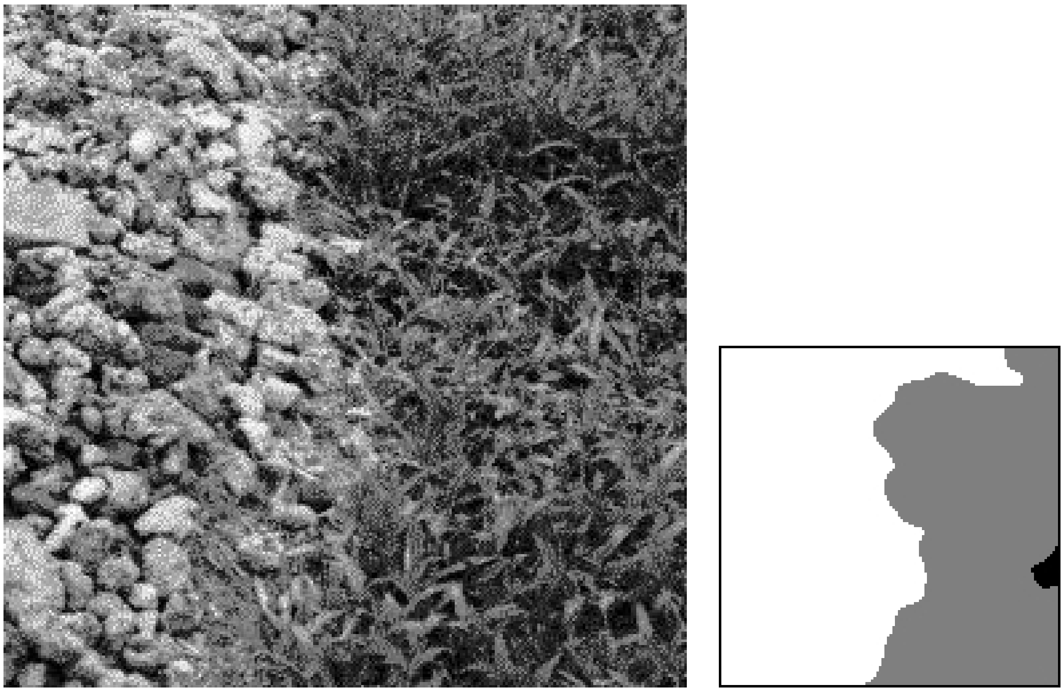
(a) (b)

Figure 4.4: (a) Image of floor covered with bricks and metal mesh. (b) Invariant segmentation result.



(a) (b)

Figure 4.5: (a) Image of floor covered with square tiles and metal mesh. (b) Invariant segmentation result.



(a) (b)

Figure 4.6: (a) Image of ground covered with grass and rocks. (b) Invariant segmentation method can identify the curvilinear boundary between the grass lawn and the rocks.

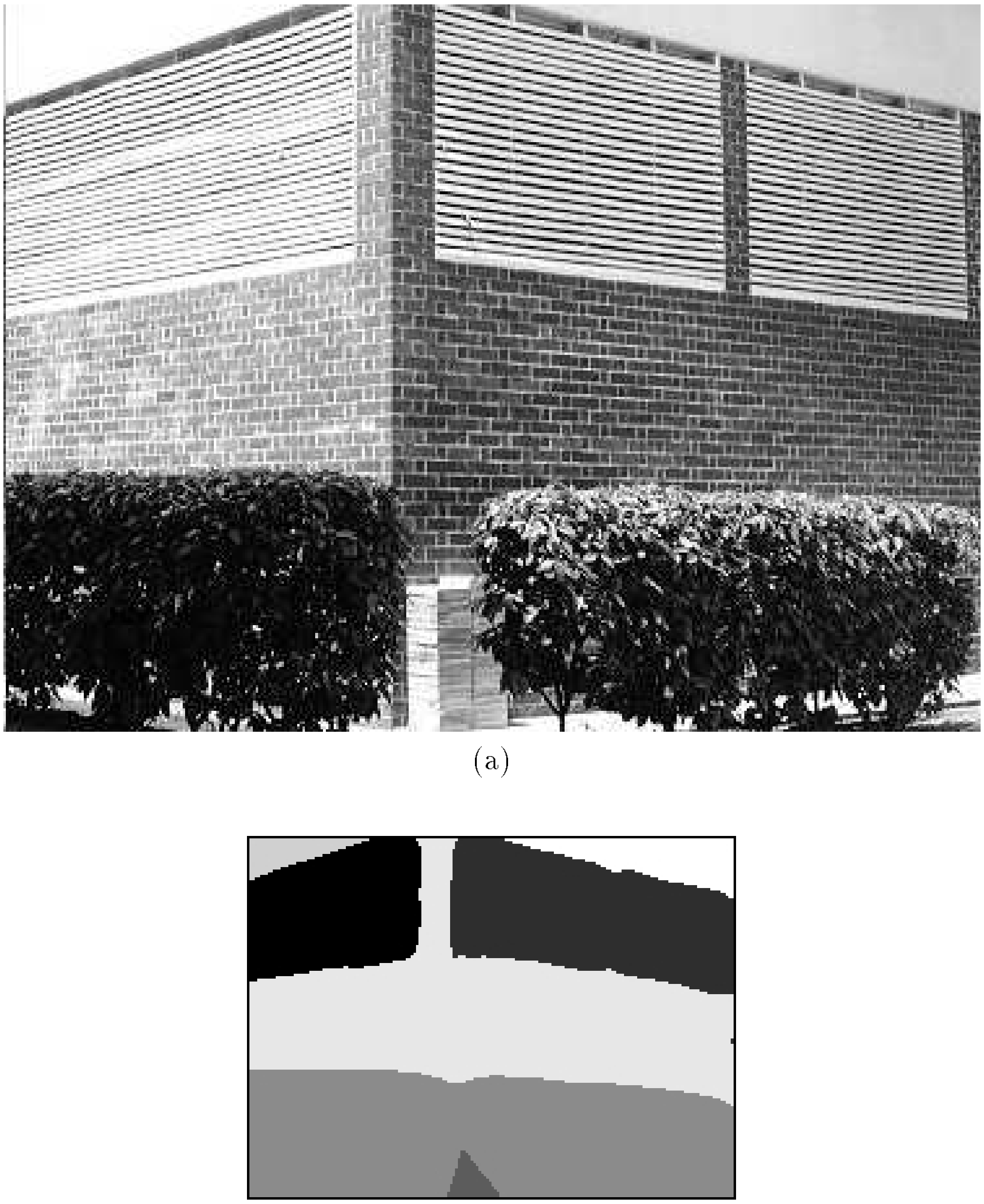


Figure 4.7: (a) Image of a complex scene containing a building with walls and windows, sitting behind two clumps of plants. (b) Invariant segmentation result.

Chapter 5

Future Work

Texture segmentation of natural scene images is a very complicated problem. The invariant segmentation algorithm described in this thesis works well for textured images with perspective distortions. However, it still has some shortcomings that need to be overcome.

5.1 Placement of Gabor Channels

In the method used in this thesis, Gabor filters are tiled at uniform intervals in the spatial frequency space. A denser tiling of the Gabor filters will help to better represent texture scale and orientation. However, it will also increase the computational effort of Gabor filtering. A better filter placement method is as follows: Observed that the energy of an image is not uniformly distributed in the Fourier domain. It is possible to cover areas of dense energy distributions with more Gabor filters and other areas with fewer filters. Areas with low magnitudes may be regarded as noise and need not be covered by any Gabor filter. Thus, Gabor transform can be operated on a small number of frequency bands and yet has good frequency and orientation resolutions.

Figure 5.1: Less well structured texture has no specific scale or orientation. Gabor features are not appropriate for describing this kind the texture

5.2 More Texture Features

The current invariant segmentation method works well on structured texture. However, Gabor features may not be useful in representing less structured texture such as the tree knot in Fig. 5.1. More texture features are needed to describe such unstructured patterns. Higher order statistics or features from MRF model, which is good at representing the randomness property of texture, may be used in addition to Gabor features to represent the texture patterns.

The effectiveness of the current method also depends on the homogeneity of the texture region. However, some perceptually uniform texture are homogeneous globally but not locally (for example, Fig. 5.2), and humans regard the whole image as a single texture. Together with the "knot" problem just mentioned, this

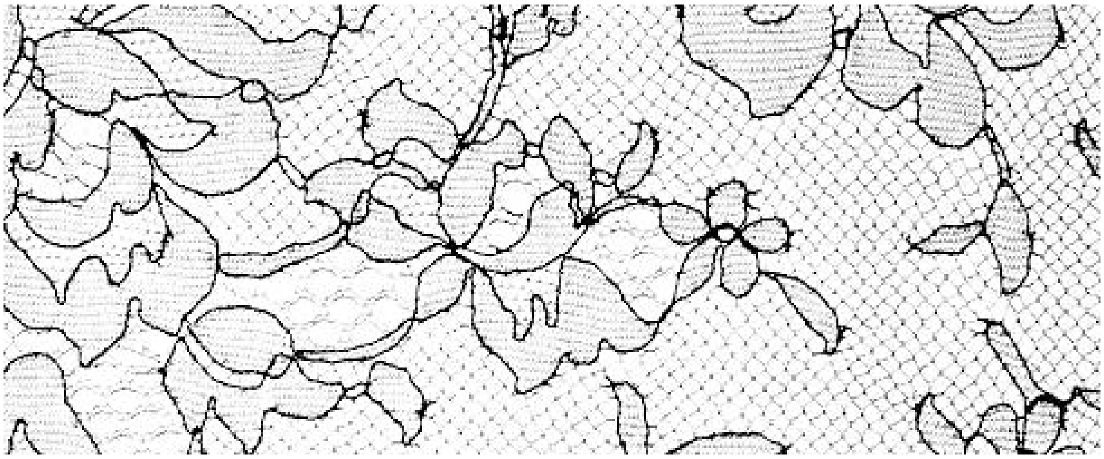


Figure 5.2: Inhomogeneous texture. But human regard it as a single uniform texture.

example indicates that a pre-classification of texture into structured or unstructured categories and a subsequent detailed classification specific to each category may benefit the texture segmentation process.

5.3 Similarity Measure

The result of texture segmentation relies heavily on the quality of similarity measure. A good similarity measure should match human's perception closely. The method presented in this thesis can take care of gradual change of texture scale and orientation. However, human visual system can also handle other variations of texture properties such as randomness and 3-D rotation. The current similarity measure does not model human's texture perception. Studies of human's judgement of texture similarity may help to define a better similarity measure. A neural network can be trained to produce similarity judgement that is closer to that of human.

Chapter 6

Conclusions

This thesis describes a novel texture segmentation method that is invariant to the gradual change of texture scale and orientation. Two key features are studied to make the algorithm scale- and orientation-invariant. First, the method adopts a distributed representation of texture features. Unlike existing methods which use only some of the Gabor channels' outputs, the invariant algorithm uses the pattern of outputs from all the channels as the texture feature. The half-peak supports of the channels are overlapped to reduce the variation of feature pattern due to changes in texture scale and orientation. Second, the invariant similarity measure performs individual shift-and-match of each local peak in the feature pattern. This similarity measure takes into account the complex changes in output pattern due to perspective distortion.

The invariant segmentation algorithm has been shown to perform well on images containing natural scenes with multiple complex texture patterns. It tolerates rotation and non-isotropic scale change of texture due to perspective distortion. In comparison with existing methods, the invariant algorithm produces segmentation results that are more consistent with human perception.

Bibliography

[1] J. Aloimonos. Shape from texture. *Biological Cybernetics*, 58:345–360, 1986.

[2] J. Bigün and J. M. H. du Buf. N-folded symmetries by complex moments in Gabor space and their application to unspervised texture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):80–87, 1994.

[3] A. C. Bovik, M. Clark, and W. S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):55–73, 1990.

[4] P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover, New York, 1966.

[5] J. L. Chen and A. Kundu. Rotation and gray scale transform invariant textur idendification using wavelet decomposition and hidden markov model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2), 1994.

[6] F. S. Cohen, Z. Fan, and M. A. Patel. Classification of rotated and scaled textured images using Gaussian Markov random field models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(2):192–202, 1991.

[7] J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America .A*, 12(7):1160–1169, 1985.

[8] J. M. H. Du Buf. Abstract processes in texture discrimination. *Spatial Vision*, 6(3):221–242, 1992.

[9] Marie-Poerre Dubuisson and Richard C. Dubes. Efficacy of fractal features in segmenting images of natural textures. *Pattern Recognition Letters*, 15(4):419–431, 1994.

[10] D. F. Dunn, W. E. Higgins, and J. Wakeley. Texture segmentation using 2-D Gabor elementary functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):130–149, 1994.

[11] J. M. Francos, A. Zvi Meiri, and B. Porat. A unified texture model based on a 2-D Wold like decomposition. *IEEE Transactions on Signal Processing*, pages 2665–2678, Aug. 1993.

[12] D. Gabor. Theory of communication. *Journal of Institute of Electrical Engineering*, 93:429–457, 1946.

[13] C. C. Gotlieb and H. E. Kreyszig. Texture descriptors based in co-occurrence matrices. *Computer Vision, Graphics, and Image Processing*, 51, 1990.

[14] R. M. Haralick. Image texture survey. In P. R. Krishnaiah and eds. L. N. Kanal, editors, *Handbook of Statistics*, volume 2. North-Holland, 1982.

[15] G. E. Hinton, J. L. McClelland, and D. E. Rumelhart. Distributed representation. In David E. Rumelhart and James L. McClelland, editors, *Parallel Distributed Processing*. MIT Press, Cambridge, Massachusetts, 1986.

[16] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[17] B. Julesz. Visual pattern discrimination. *IRE Trans. Information Theory*, 8:84–92, 1962.

[18] B. Julesz, H. L. Frisch, E. N. Gilbert, and L. A. Shepp. Inability of humans to discriminate between visual textures that agree in second-order statistics-revisited. *Perception*, 2:391–405, 1973.

[19] H. Kaizer. Texture measurements for the automatic classification of imagery. *IEEE Transactions on Electromagnetic Compatibility*, 18(1):38–42, 1976.

[20] R. L. Kashyap and A. Khotanzad. A model-based method for rotation invariant texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):472–481, 1986.

[21] J. Kender. *Shape from Texture*. PhD thesis, Carnegie Mellon University, 1980.

[22] F. Liu and R. W. Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):722–733, 1996.

[23] C. S. Lu, P. C. Chung, and C. F. Chen. Unsupervised texture segmentation via wavelet transform. *Pattern Recognition*, 30(5):729–742, 1997.

[24] S. V. R. Madiraju, T. M. Caelli, and C.-C. Liu. On the covariance technique for robust and rotation invariant texture processing. In *ACCV'93 Asian Conference on Computer Vision*, pages 171–174, 1993.

[25] S. V. R. Madiraju and C.-C. Liu. Rotation invariant texture classification using covariance. In *Proceedings of International Conference on Image Processing*, volume 2, pages 655–659, 1994.

[26] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1996.

[27] H. H. Nguyen and P. Cohen. Gibbs random fields, fuzzy clustering, and the unsupervised segmentation of textured images. *CVGIP: Graphical Models and Image Processing*, 55(1):1–19, 1993.

[28] D. K. Panjwani and G. Healey. Markov random field models for unsupervised segmentation of texture color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):939–954, 1995.

[29] O. Pichler, A. Teuner, and B. J. Hosticka. A comparison of texture feature extraction using adaptive gabor filtering, pyramidal and tree structured wavelet transforms. *Pattern Recognition*, 29(5):733–742, 1996.

[30] K. Shanmugam R. M. Haralick and I. Dinstein. Texture features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6), 1973.

[31] J. S. Weszka, A. Rosenfeld, E. J. Carton, R. L. Kirby, and J. M. Mohr. A comparitive study of texture measures for terrian classification. Technical Report TR361, Computer Vision Laboratory, Computer Science Center, University of Maryland, March 1975.

[32] Guang-You Xu and King-Sun Fu. Natural scene segmentation based on multiple threshold and textural measurement. In *Proceedings of the Seventh International Conference on Pattern Recognition*, pages 1111–1113, 1984.

[33] J. You and H. A. Cohen. Classification and segmentation of rotated and scaled textured images using texture "tuned" masks. *Pattern Recognition*, 26(2):245–258, 1993.