

Chapter 7

Texture

7.1 Introduction

Texture plays an important role in many machine vision tasks such as surface inspection, scene classification, and surface orientation and shape determination. For example, surface texture features are used in the inspection of semiconductor wafers, gray-level distribution features of homogeneous textured regions are used in the classification of aerial imagery, and variations in texture patterns due to perspective projection are used to determine three-dimensional shapes of objects.

Texture is characterized by the spatial distribution of gray levels in a neighborhood. Thus, texture cannot be defined for a point. The resolution at which an image is observed determines the scale at which the texture is perceived. For example, in observing an image of a tiled floor from a large distance we observe the texture formed by the placement of tiles, but the patterns within the tiles are not perceived. When the same scene is observed from a closer distance, so that only a few tiles are within the field of view, we begin to perceive the texture formed by the placement of detailed patterns composing each tile. For our purposes, we can define texture as repeating patterns of local variations in image intensity which are too fine to be distinguished as separate objects at the observed resolution. Thus, a connected set of pixels satisfying a given gray-level property which occur repeatedly in an image region constitutes a textured region. A simple example is a repeated pattern of dots on a white background. Text printed on white paper such as this page also constitutes texture. Here, each gray-level primitive is formed

by the connected set of pixels representing each character. The process of placing the characters on lines and placing lines in turn as elements of the page results in an ordered texture. There are three primary issues in texture analysis: texture classification, texture segmentation, and shape recovery from texture.

In texture classification, the problem is identifying the given textured region from a given set of texture classes. For example, a particular region in an aerial image may belong to agricultural land, forest region, or an urban area. Each of these regions has unique texture characteristics. The texture analysis algorithms extract distinguishing features from each region to facilitate classification of such patterns. Implicit in this is the assumption that the boundaries between regions have already been determined. Statistical methods are extensively used in texture classification. Properties such as gray-level co-occurrence, contrast, entropy, and homogeneity are computed from image gray levels to facilitate classification. These are discussed in Section 7.2. The statistical methods are particularly useful when the texture primitives are small, resulting in *microtextures*. On the other hand, when the size of the texture primitive is large, it becomes necessary to first determine the shape and properties of the basic primitive and then determine the rules which govern the placement of these primitives, forming *macrotextures*. Such structural methods are briefly discussed in Section 7.3. As an alternative to the bottom-up analysis of image pixels to determine texture properties for classification, model-based methods to synthesize texture have been studied. In these methods a model for texture is first assumed and its parameters are then estimated from the image region such that an image synthesized using the model closely resembles the input image region. The parameters are then useful as discriminating features to classify the region. These are discussed in Section 7.4.

As opposed to texture classification, in which the class label of a single homogenous region is determined using properties computed from the region, texture segmentation is concerned with automatically determining the boundaries between various textured regions in an image. Although quantitative texture measures, once determined, are useful in segmentation, most of the statistical methods for determining the texture features do not provide accurate measures unless the computations are limited to a single texture region. Both region-based methods and boundary-based methods have been attempted to segment textured images. These methods are analogous

to those used for object-background separation methods discussed in earlier chapters. Texture segmentation is still an active area of research, and numerous methods, each designed for a particular application, have been proposed in the literature. However, there are no general methods which are useful in a wide variety of situations. Thus, we do not cover texture segmentation methods in this book.

Image plane variations in the texture properties, such as density, size, and orientation of texture primitives, are the cues exploited by shape-from-texture algorithms. For example, the texture gradient, defined as the magnitude and direction of maximum change in the primitive size of the texture elements, determines the orientation of the surface. Quantifying the changes in the shape of texture elements (e.g., circles appearing as ellipses) is also useful to determine surface orientation. These are discussed in Section 7.5.

7.2 Statistical Methods of Texture Analysis

Since texture is a spatial property, a simple one-dimensional histogram is not useful in characterizing texture (for example, an image in which pixels alternate from black to white in a checkerboard fashion will have the same histogram as an image in which the top half is black and the bottom half is white). In order to capture the spatial dependence of gray-level values which contribute to the perception of texture, a two-dimensional dependence matrix known as a *gray-level co-occurrence* matrix is extensively used in texture analysis. Another measure that has been used extensively is the *autocorrelation* function. These are discussed briefly in this section.

Gray-Level Co-occurrence Matrix

The gray-level co-occurrence matrix $P[i, j]$ is defined by first specifying a displacement vector $\mathbf{d} = (dx, dy)$ and counting all pairs of pixels separated by \mathbf{d} having gray levels i and j . For example, consider the simple 5×5 image having gray levels 0, 1, and 2 as shown in Figure 7.1(a). Since there are only three gray levels, $P[i, j]$ is a 3×3 matrix. Let the position operator be specified as $(1, 1)$, which has the interpretation: one pixel to the right and one pixel below. In a 5×5 image there are 16 pairs of pixels which satisfy this spatial separation. We now count all pairs of pixels in which the first

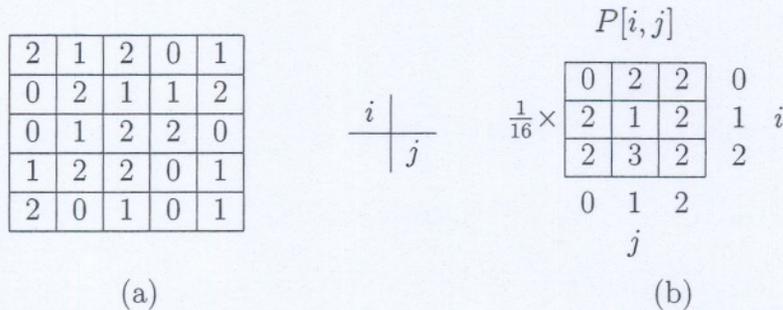


Figure 7.1: (a) A 5×5 image with three gray levels 0, 1, and 2. (b) The gray-level co-occurrence matrix for $\mathbf{d} = (1, 1)$.

pixel has a value of i and its matching pair displaced from the first pixel by \mathbf{d} has a value of j , and we enter this count in the i th row and j th column of the matrix $P[i, j]$. For example, there are three pairs of pixels having values $[2, 1]$ which are separated by the specified distance, and hence the entry $P[2, 1]$ has a value of 3. The complete matrix $P[i, j]$ is shown in Figure 7.1(b).

Note that $P[i, j]$ is not symmetric since the number of pairs of pixels having gray levels $[i, j]$ does not necessarily equal the number of pixel pairs having gray levels $[j, i]$. The elements of $P[i, j]$ are normalized by dividing each entry by the total number of pixel pairs. In our example, each entry is divided by 16. This normalized $P[i, j]$ is then treated as a probability mass function since the entries now add up to 1.

It is easy to illustrate that the gray-level co-occurrence matrix captures the spatial distribution of gray levels with the following simple example. Consider the 8×8 binary image of a checkerboard shown in Figure 7.2(a), where each square corresponds to a single pixel. Since there are only two gray levels, $P[i, j]$ is a 2×2 matrix. If we define $\mathbf{d} = (1, 1)$ as before, we get the normalized $P[i, j]$ shown in Figure 7.2(b). Notice that the only pairs that occur are $[1, 1]$ and $[0, 0]$ because of the well-defined structure of pixels; the off-diagonal elements are zero. Similarly, if the vector \mathbf{d} is defined as $(1, 0)$, the only entries will be those corresponding to $[0, 1]$ and $[1, 0]$, as shown in Figure 7.2(c), and the diagonal elements are zero.

In the above example, if the black pixels are randomly distributed throughout the image with no fixed structure, the gray-level co-occurrence matrix

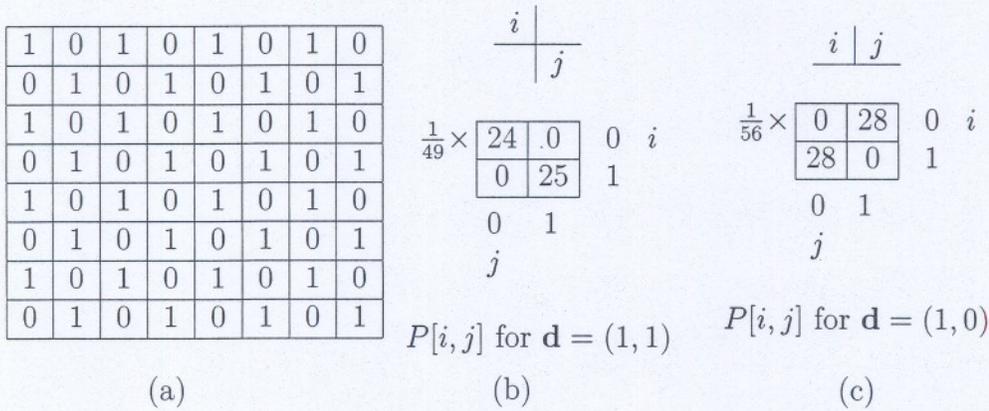


Figure 7.2: (a) An 8×8 checkerboard image. (b) The gray-level co-occurrence matrix for $\mathbf{d} = (1, 1)$. (c) The gray-level co-occurrence matrix for $\mathbf{d} = (1, 0)$.

will not have any preferred set of gray-level pairs. In such a case the matrix is expected to be uniformly populated. Thus, a feature which measures the randomness of gray-level distribution is the *entropy*, defined as

$$\text{Entropy} = - \sum_i \sum_j P[i, j] \log P[i, j]. \quad (7.1)$$

Note that the entropy is highest when all entries in $P[i, j]$ are equal; such a matrix corresponds to an image in which there are no preferred gray-level pairs for the specified distance vector \mathbf{d} . The features of *energy*, *contrast*, and *homogeneity* are also defined using the gray-level co-occurrence matrix as given below:

$$\text{Energy} = \sum_i \sum_j P^2[i, j] \quad (7.2)$$

$$\text{Contrast} = \sum_i \sum_j (i - j)^2 P[i, j] \quad (7.3)$$

$$\text{Homogeneity} = \sum_i \sum_j \frac{P[i, j]}{1 + |i - j|} \quad (7.4)$$

The choice of the displacement vector \mathbf{d} is an important parameter in the definition of the gray-level co-occurrence matrix. Occasionally, the co-occurrence

matrix is computed for several values of \mathbf{d} and the one which maximizes a statistical measure computed from $P[i, j]$ is used. The gray-level co-occurrence matrix approach is particularly suitable for describing microtextures. It is not suitable for textures comprising large area primitives since it does not capture shape properties. Gray-level co-occurrence matrices have been used extensively in remote sensing applications for land-use classification.

Autocorrelation

The autocorrelation function $p[k, l]$ for an $N \times N$ image is defined as follows:

$$p[k, l] = \frac{\frac{1}{(N-k)(N-l)} \sum_{i=1}^{(N-k)} \sum_{j=1}^{(N-l)} f[i, j] f[i+k, j+l]}{\frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N f^2[i, j]}, \quad 0 \leq k, l \leq N-1. \quad (7.5)$$

For images comprising repetitive texture patterns the autocorrelation function exhibits periodic behavior with a period equal to the spacing between adjacent texture primitives. When the texture is coarse, the autocorrelation function drops off slowly, whereas for fine textures it drops off rapidly. The autocorrelation function is used as a measure of periodicity of texture as well as a measure of the scale of the texture primitives.

7.3 Structural Analysis of Ordered Texture

When the texture primitive is large enough to be individually segmented and described, then structural methods which describe the primitives and their placement rules are useful. For example, consider a simple texture formed by the repeated placement of homogeneous gray-level discs in a regular grid pattern as shown in Figure 7.3(a). Such a texture can be described by first segmenting the discs using a simple method such as connected component labeling, described earlier, and then determining the regular structure formed by the centroids of these connected components. For more general binary images the primitives can be first extracted using morphological methods and then their placement rules determined. Such morphological methods are particularly useful when the image is corrupted by noise or other nonrepeating random patterns which would be difficult to separate in a simple connected component method. For example, when the image shown in Figure 7.3(a) is

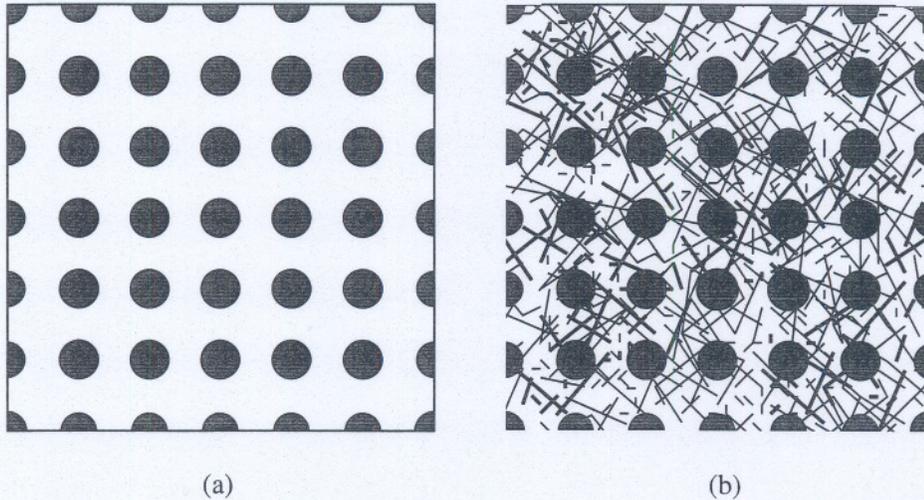


Figure 7.3: (a) A simple texture formed by repeated placement of discs on a regular grid. (b) Texture in (a) corrupted by random streaks of lines.

corrupted by noise resulting in random streaks of lines as shown in Figure 7.3(b), morphological techniques (see Chapter 2) can be used to locate all discs.

For gray-scale images we can define a predicate which is satisfied by all pixels within each blob corresponding to a primitive. A commonly used predicate is the gray-level homogeneity predicate. The image is initially processed using a Laplacian of Gaussian filter (see Chapter 5). Primitive regions are then identified by grouping all those pixels which are not on or near edge pixels. For homogeneous blobs properties such as size, elongation, and orientation are useful features. Measures based on co-occurrence of these primitives obtained by analyzing their spatial relationship are then used to characterize the texture.

7.4 Model-Based Methods for Texture Analysis

An approach to characterize texture is to determine an analytical model of the textured image being analyzed. Such models have a set of parameters.

The values of these parameters determine the properties of the texture, which may be synthesized by applying the model. The challenge in texture analysis is to estimate these model parameters so that the synthesized texture is visually similar to the texture being analyzed.

Markov random fields (MRFs) have been studied extensively as a model for texture. In the discrete Gauss-Markov random field model, the gray level at any pixel is modeled as a linear combination of gray levels of its neighbors plus an additive noise term as given by the following equation:

$$f[i, j] = \sum_{[k, l]} f[i - k, j - l] h[k, l] + n[i, j] \quad (7.6)$$

Here the summation is carried out over a specified set of pixels which are neighbors to the pixel $[i, j]$. The parameters of this model are the weights $h[k, l]$. These parameters are computed from the given texture image using least-squares method. These estimated parameters are then compared with those of the known texture classes to determine the class of the particular texture being analyzed.

When patterns forming texture have the property of self-similarity at different scales, fractal-based models may be used. A set is said to have the property of self-similarity if it can be decomposed as a nonoverlapping union of N copies of itself scaled down by a factor r . Such a texture is characterized by its fractal dimension D , given by the equation

$$D = \frac{\log N}{\log(\frac{1}{r})} \quad (7.7)$$

The fractal dimension is a useful feature for texture characterization. Estimation of D from an image is rather difficult because natural textures do not strictly follow the deterministic repetitive model of fractals assumed above, but have statistical variations.

7.5 Shape from Texture

Variations in the size, shape, and density of texture primitives provide clues for estimation of surface shape and orientation. These are exploited in *shape-from-texture* methods to recover 3-D information from 2-D images. As an illustration, consider the regular ordered texture shown in Figure 7.3(a) slanted

at an angle α such that the top of the surface is farther away from the camera than the bottom; for simplicity, let us assume that all points along a given horizontal line are at the same depth from camera (i.e., there is no tilt). This is illustrated in Figure 7.4(a). The corresponding image captured is shown in Figure 7.5. Note that the discs now appear as ellipses, which is a clue that the surface is not parallel to the image plane. The sizes of these ellipses decrease as a function of y' in the image plane. In other words, there are more ellipses for a unit area in the image plane near the top of the image compared with the center, resulting in a density gradient. Furthermore, the aspect ratio (ratio of minor to major diameters of an ellipse) does not remain constant, resulting in an aspect ratio gradient [36]. To show this, we first derive an expression for the major and minor diameters of the ellipse as a function of the slant angle and the position of the ellipse in the image plane.

Let the diameter of the disc be d . Consider the disc at the image center. The major diameter of the ellipse in the image plane corresponding to this disc is given by the perspective projection equation

$$d_{\text{major}}(0, 0) = \frac{df}{z} \quad (7.8)$$

where z is the distance of the disc from the camera center and f is the focal length of the camera. The minor diameter of this ellipse is affected not only by the perspective projection but also by the foreshortening effect due to the slant angle α . This is given by the equation

$$d_{\text{minor}}(0, 0) = \frac{df}{z} \cos \alpha. \quad (7.9)$$

Thus, the aspect ratio of the ellipse at the center of the image plane is equal to $\cos \alpha$. All ellipses along the same horizontal line in the image plane will have the same aspect ratio.

Now consider an ellipse with its center at $(0, y')$ in the image plane. The disc corresponding to this ellipse is at an angle with respect to the optical axis as shown in Figure 7.4(b) and (c), where $\tan \theta = y'/f$. The major diameter of the ellipse is now given by Equation 7.8 with a slight modification. Since the disc is now located at a distance S away from the camera center, z must be replaced with S . From Figure 7.4(b) and (c):

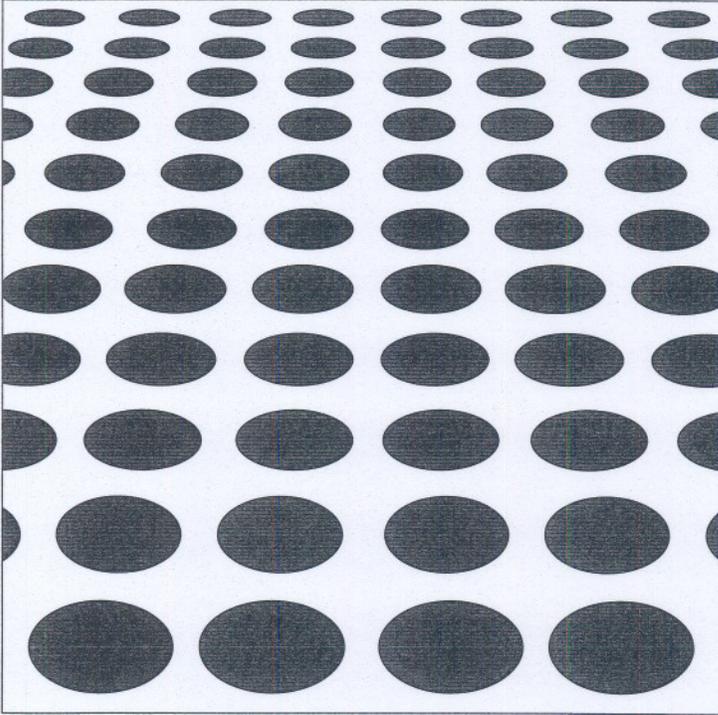


Figure 7.5: The image captured from the camera system in Figure 7.4.

$$\frac{S - z}{\tan \alpha} = S \tan \theta \quad (7.10)$$

$$S(1 - \tan \theta \tan \alpha) = z \quad (7.11)$$

$$S = \frac{z}{1 - \tan \theta \tan \alpha}. \quad (7.12)$$

Therefore,

$$d_{\text{major}}(0, y') = \frac{df}{z}(1 - \tan \theta \tan \alpha). \quad (7.13)$$

The derivation of $d_{\text{minor}}(0, y')$ is a bit more involved. First, from Figure 7.4(b),

$$S_1 = \frac{z}{1 - \tan \theta_1 \tan \alpha} \quad (7.14)$$

and

$$S_2 = \frac{z}{1 - \tan \theta_2 \tan \alpha}. \quad (7.15)$$

Now if we assume that the diameter of the disk d is very small so that it subtends a small angle at the camera center, we can approximate

$$\theta_1 \approx \theta_2 \approx \theta. \quad (7.16)$$

Therefore,

$$S_1 \approx S_2 \approx S \quad (7.17)$$

$$\approx \frac{z}{1 - \tan \theta \tan \alpha}. \quad (7.18)$$

Now from Figure 7.4(b) we know that

$$AC = d \cos \alpha. \quad (7.19)$$

However, we need to find the distance AB .

Noting that

$$S_1 - S_2 = d \sin \alpha, \quad (7.20)$$

we find

$$BC = d \sin \alpha \tan \theta. \quad (7.21)$$

Therefore,

$$AB = d(\cos \alpha - \sin \alpha \tan \theta) \quad (7.22)$$

$$= d \cos \alpha (1 - \tan \alpha \tan \theta). \quad (7.23)$$

Now, by the perspective projection,

$$\frac{d_{\text{minor}}(0, y')}{f} = \frac{d \cos \alpha (1 - \tan \alpha \tan \theta)}{\frac{z}{1 - \tan \theta \tan \alpha}} \quad (7.24)$$

Therefore, the minor diameter of the ellipse at $(0, y')$ is given by

$$d_{\text{minor}}(0, y') = \frac{d f}{z} \cos \alpha (1 - \tan \theta \tan \alpha)^2 \quad (7.25)$$

Thus, the aspect ratio given by $\cos \alpha (1 - \tan \theta \tan \alpha)$ decreases as θ increases, resulting in an *aspect ratio gradient*.

Variations in the image plane features such as size, shape, density, and aspect ratio of texture primitives can be exploited to recover the surface orientation (and ultimately the 3-D surface shape) of scene objects. To do this, however, we must have accurate methods to delineate each primitive in the image plane. For simple binary primitives such as the disc used in our illustration, fairly accurate segmentation of individual image plane primitives for measurement is possible. However, for more complex gray-level textures corrupted by noise, it is difficult to accurately estimate the image plane features.

7.6 Further Reading

There is a wealth of literature describing methods for texture modeling, synthesis, analysis, and segmentation. More details and references to many of the topics covered in this chapter may be found in [103, 239, 200]. Haralick, Shanmugam, and Dinstein [100] and Connors and Harlow [61] have used gray-level co-occurrence matrices to analyze textures in satellite images. Pentland [194] describes textures using fractals. Cross and Jain [64] describe a Markov random field model for texture. Methods for estimating the model parameters may be found in [138]. The shape-from-texture methods described in this chapter can be seen in more detail in [36]. Other approaches are given in [13, 252, 135]. Rao [200] includes a taxonomy for texture description and identification.

Exercises

7.1 What is texture? How is it defined for images?

7.2 How can you classify texture? Give some major classes and the characteristics that are used to define each class.

7.3 How are the following used in texture recognition:

- Fractals
- Co-occurrence matrices
- Fourier transforms

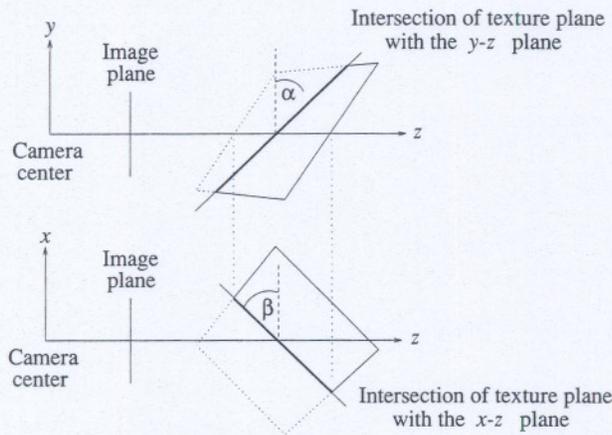


Figure 7.6: Diagram illustrating a textured plane with an arbitrary slant and tilt with respect to the camera axes.

- Markov random fields

Discuss their application and mention the types of texture where the above are most suitable.

- 7.4 How can images be segmented using texture features? Give an approach for image segmentation using texture.
- 7.5 How can the shape of a surface be determined by using its texture characteristics? Give details of an algorithm to determine shape from texture.
- 7.6 Mention three applications of a machine vision system where texture plays a prominent role.
- 7.7 For Figure 7.1(a), find the gray-level co-occurrence matrix for $\mathbf{d} = (0, 2), (2, 0), (2, 2)$. Repeat for Figure 7.2(a).
- 7.8 Derive the expressions for d_{major} and d_{minor} when the plane containing the texture pattern is tilted in addition to slant (i.e., oriented in any arbitrary direction as long as it is visible to the camera) as shown in Figure 7.6.

Computer Projects

- 7.1 Take an image of a textured plane (no tilt, only slant) and measure the major and minor diameters to verify their expressions.
- 7.2 Similar to above, now with unknown slant angle. Estimate α from the measured values of d_{major} and d_{minor} .
- 7.3 Same as above, now with both slant and tilt unknown.
- 7.4 Develop a fractal-based algorithm to discriminate among textures of different kind. Implement this algorithm and test its performance on several images. Relate the performance of this algorithm to judgments by humans about the similarity and ordering of texture.
- 7.5 Implement a texture recognition algorithm based on co-occurrence matrices. Apply it to several texture images to study its discrimination power.
- 7.6 Implement a texture segmentation algorithm. Test it on several images containing several regions of different types of texture.
- 7.7 Take a picture of a brick wall from the side so that the wall is at about 45 degrees from the optical axis of the camera. Develop a shape-from-texture algorithm and apply it to recover the structure of the surface.