Model-based texture classification under varying illumination

RM/02/8

Andreu Gonzalez — September 2002

Contents

List of figures

List of tables

Principal symbols and abbreviations

Symbol	Description	Section first introduced
λ	spectral-dependency of the reflectance function	2.2.1
$\mathbf{n}$	unit vector normal to the surface	2.2.2
p	facet slope in the x-direction	2.2.2
q	facet slope in the y-direction	2.2.2
$\mathbf{l}$	unit vector pointing at the light source	2.2.3
$s(x, y)$	surface height map	2.2.3
$\mathbf{d}(x, y)$	surface derivative field vector	2.2.3
$i(x, y)$	normalised intensity image	2.2.3
τ	angle of tilt of the illuminant vector $\mathbf{l}$	2.2.3
σ	angle of slant of the illuminant vector $\mathbf{l}$	2.2.3
$b(x, y)$	Gaussian point spread function	2.3.1
σ_b	blur parameter of $b(x, y)$	2.3.1
σ_s	root mean square roughness	2.4.1
R_{cla}	centre line average roughness	2.4.1
$s(x)$	surface height profile in the x-direction	2.4.1
$\mathbf{f}$	feature vector	3.2.2
$g(x, y)$	Gabor elementary function	3.3.3
u	horizontal frequency-domain coordinate	3.3.3
u_0	centre frequency of a Gabor filter in the u-axis	3.3.3
v	vertical frequency-domain coordinate	3.3.3

v_0	centre frequency of a Gabor filter in the v-axis	3.3.3
ϕ	rotation of spatial rectilinear coordinates	3.3.3
$e(x,y)$	Gaussian envelope of a Gabor filter	3.3.3
σ_x	standard deviation of $e(x,y)$ in the x-direction	3.3.3
σ_y	standard deviation of $e(x,y)$ in the y-direction	3.3.3
ω	radial frequency	3.3.3
ω_0	radial centre frequency of a Gabor filter	3.3.3
θ	polar frequency angle	3.3.3
θ_0	orientation of a Gabor filter	3.3.3
$G(u,v)$	Fourier domain representation of $g(x,y)$	3.3.3
σ_u	extent of the Gabor filter in the u-direction	3.3.3
σ_v	extent of the Gabor filter in the v-direction	3.3.3
G_r	class identifier of class r	3.4.2
$d(\mathbf{f})$	discriminant function	3.4.2
K_r	r-th disjoint feature subspace	3.4.2
$\mathbf{m}_r$	vector of feature measure mean for group G_r	3.4.2
$\mathbf{C}_r$	covariance matrix of group G_r	3.4.2
$\mathbf{C}_p$	pooled covariance matrix	3.4.2
$\hat{m}_j$	mean estimator for j-th feature	3.4.3
$\hat{C}_{jk}$	covariance estimator between j and k-th features	3.4.3
$R(p,q)$	reflectance map	4.4.2
$a(x,y)$	intensity image representation of albedo texture	5.2.1
γ	incident light flux	5.3.1
ρ	variation in reflectance property	5.3.1
$i_{0,90,180}(x,y)$	intensity images at τ =0°,90°,180° and same σ	5.3.1
$h(x,y)$	smoothing post-filter	5.4.1
$o(x,y)$	Gabor-filtered intensity image	5.4.1
$\aleph$	a generic non-linear operator	5.4.1
$m(x,y)$	non-linear energy estimation of $o(x,y)$	5.4.1
$f(x,y)$	feature texture	5.4.1

$I(u,v)$	Fourier representation of the image	5.4.2
$O(u,v)$	Gabor-filtered image in the frequency domain	5.4.2
$G_{even}(u,v)$	Real and even-symmetrical part of $G(u,v)$	5.4.2
$G_{odd}(u,v)$	Real and odd-symmetrical part of $G(u,v)$	5.4.2
$I_R(u,v)$	Real and even-symmetrical part of $I(u,v)$	5.4.2
$I_I(u,v)$	Imaginary and odd-symmetrical part of $I(u,v)$	5.4.2
$o_R(x,y)$	Real part of $o(x,y)$	5.4.2
$o_I(x,y)$	Imaginary part of $o(x,y)$	5.4.2
α	saturation parameter of the rectified sigmoid	5.4.3
σ_L	standard deviation of a lowpass Gaussian $h(x,y)$	5.4.3
B_ω	frequency bandwidth of a Gabor filter	5.5.1
B_θ	orientation bandwidth of a Gabor filter	5.5.1
$\hat{i}_{\tau,\sigma}(x,y)$	rendered intensity image at tilt τ and slant σ	6.1
$i_{\tau,\sigma}(x,y)$	actual intensity image at tilt τ and slant σ	6.1
k_1, k_2, k_3	adjustment constants for $MSE(\tau)$ best fit curves	6.3.2
$\mathbf{L}$	photometric stereo illuminant matrix	6.3.3
$\det \mathbf{L}$	determinant of the illuminant matrix	6.3.3
n_1, n_2	white noise	6.3.3

Abbreviation	Description	Section first introduced
AASR	absolute average slope ration	2.4.1
ACF	autocorrelation function	2.4
ADC	analogue to digital converter	2.3.1
ARMA	autoregressive moving average (model)	3.3.1
BRDF	bidirectional reflectance distribution function	2.2.1
CCD	charge-coupled device	2.3.1
FFT	fast Fourier transform	3.3.1
FIR	finite impulse response	3.3.1
GEF	Gabor elementary function	3.3.3
IFFT	inverse fast Fourier transform	5.4.3
LPF	low pass filter	5.4.3
MSE	mean square error	6.1
MTF	modulation transfer function	3.3.3
PDF	probability density function	5.4.3
PSD	power spectral density	2.4
PSF	point spread function	2.3.1
RGB	red, green, blue (channels)	2.2.1
S/N	signal to noise (ratio)	5.5.2
TEC	total error of classification	3.4.2

Acknowledgements

I would especially like to thank my supervisor, Dr Mike Chantler, for his ideas, support, and guidance throughout this dissertation.

I would like to extent my thanks to the International Relations Office of E.T.S.E.T.B., particularly Dr Josep Vidal and Patricia Duarte, and Dr Mike Chantler, Heriot-Watt University, for making the Socrates exchange possible between both universities.

I would also like to express my gratitude to all the people who work in the texture lab for their welcome and willingness to help me in my work. Special thanks are due to Cristina Gullon for her time, advice and significant help.

I wish to express my gratitude to my parents, grandparents, Tieta, and Antoni for all their invaluable support and encouragement.

Finally, I would like to say that I am especially indebted to Rosa for all her company, backing, support, and smile.

Abstract

This dissertation presents a complete texture classification system to overcome the problem induced by changes in the angle of illumination incident upon a 3D surface. The system works on the basis of a surface model, previously formed by means of a photometric stereo technique. From this model, the system is able to render a 2D image of the surface at any particular illuminant direction, thus providing a more appropriate data for training the classifier.

Many laboratory experiments are carried out in order to assess the accuracy of image prediction as an individual component. The investigation considers a large diversity of textures, including challenging situations such as rough, specular and anisotropic surfaces. It is concluded that the predicted images, yet not being perfectly accurate, are in all cases a much more reliable training data than a merely single image from a single illuminant direction.

The technique is then evaluated into a supervised statistical classification, which combines a bank of Gabor filters for feature extraction with a linear Bayes classifier. The classification performance is tested for different composite images, consisting of a varying number of disjoint textures and configurations. It is shown that our approach significantly reduces the misclassification rate, when compared with a naive classification system. Furthermore, in some cases it even reaches the level of accuracy that one would obtained with the proviso that training and classification were performed under invariant illumination.

Chapter 1

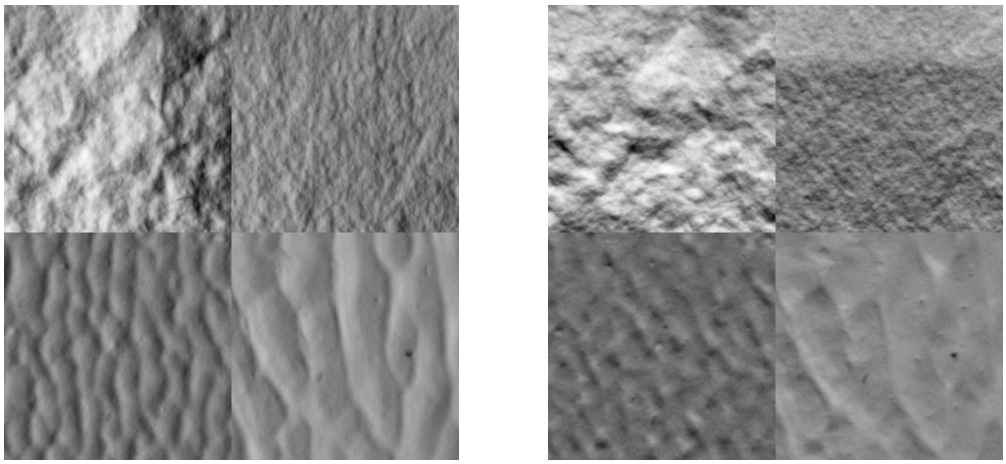
Introduction

1.1 Motivation

The motivation of this dissertation stemmed from the necessity to find reliable methods of reducing classification errors caused by changes in illuminant direction. The effect of varying lighting conditions on supervised texture classification was first investigated by [Chantler94]. It was observed that the direction of illumination affects the directionality of an image obtained from a given surface and indeed modifies its appearance. These changes can induce critical misclassification rates, providing that the illumination is altered between training and classification.

In spite of being a serious problem, nearly all of the classification approaches proposed in the literature do not take into account the effect of illumination on the imaged scene, or it is simply considered to be constant. Nevertheless, there are a wide range of applications in which texture classification may have to be performed under varying lighting conditions. For instance, close proximity point lighting, often used for inspection purposes, provides illumination at varying angles over the scene. Remote sensing systems that provide their own artificial lighting such as active sonar or radar are non-stationary, and hence the illuminant direction is dependent on the orientation of the survey platform itself. Other remote sensing devices using natural illumination are also affected by changes in illuminant vector according to the time of day.

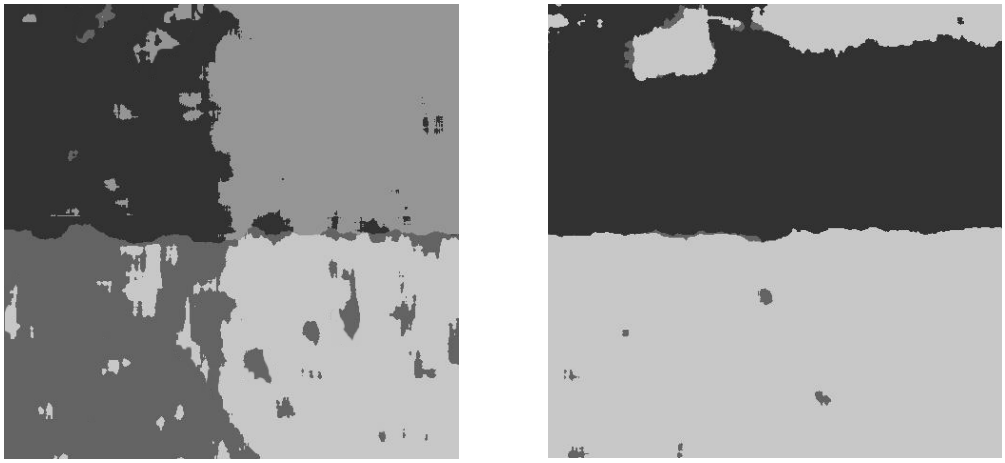
In order to illustrate the problematic, let us consider the following example. Figure 1.1 shows a composite image consisting of four textures illuminated from two different tilt angles and same slant angle[1]. Note that the pair of textures both on top and on bottom resemble each other closely. Observe how a shift in the tilt angle is clearly manifested in the perceived image of each surface (comparing Figure 1.1.a with Figure 1.1.b): *illumination variation attenuates or accentuates the directional information of the image texture*. It is then not surprising that a classifier struggles to recognise the images as belonging to the same textured surface. It follows that a classifier, which has been trained using the first set of images and performs well under these conditions, is not able to accurately classify the second one (see Figure 1.2). As a result, a substantial increase in classification error is experimented, leading to grave consequences as the classifier is not longer capable of discriminating between both pairs of alike textures, and only two regions are identified.



(a) **(b)**

Figure 1.1 Montage of four textures illuminated from **(a)** tilt angle of 0°, **(b)** tilt angle of 90°.

[1] The direction of the illuminant with respect to a texture is commonly defined by two polar co-ordinates: tilt and slant. As referred to here, tilt is the angle that the projection of the illuminant vector onto the texture reference plane makes with an axis in that plane. Its companion, slant angle, is the angle that the illuminant vector makes with a normal to the reference plane (see Figure 2.2).



(a) **(b)**

Figure 1.2 Classification results for montage of Figure 1.1: **(a)** training and classification are carried out under same illuminant direction (tilt of 0°), **(b)** training at tilt of 0°, whereas classification is performed at tilt of 90°.

It is therefore desirable to find a mode of maintaining classification rates as constant as possible regardless of the illumination condition. This text works on the basis of a model-based approach suggested by [McGunnigle98], which was originally intended to discriminate between rough, textured surfaces. In relation to McGunnigle's proposal, the aims of this dissertation are:

(i) to present the integration of a photometric model-based approach into a complete classification system,

(ii) to assess the accuracy of this photometric technique with respect to surface characteristics and illuminant,

(iii) to evaluate the model-based classification performance for a wide range of real textures under varying lighting conditions, and

(iv) to investigate a possible relationship between photometric inaccuracy and classification error, i.e. between (ii) and (iii).

1.2 Scope of the research

This work is concerned with the application of texture analysis techniques to the classification of multitextured images, i.e. those composed of several disjoint textures. The texture classification system, as referred to in this research, involves three main processes, as illustrated in Figure 1.3. The image acquisition is implemented using an imaging device—specifically a still camera and frame store—and directional lighting. The digitised image is then fed into an algorithmic classifier which produces a set of numerical descriptors of the textures. Finally, decision rules are applied to classify the image into classes.

For reasons of brevity not all the restrictions made in this research are described here (further details will be given in the appropriate chapter). Among other restrictions:

- The surface is considered to be globally flat, i.e. each of its partial derivative fields sums approximately to zero.
- The illumination is assumed to be unidirectional and of equal intensity across the imaged region.
- The topography of the surface is expected to be small relative to the distance between the camera and the physical sample, and the projection of the surface onto the camera array is orthographic.
- The viewer's position is held constant and fixed vertically above the physical texture which lies upon a horizontal plane.
- The image texture is assumed to contain both topological and albedo texture[2].

[2] The term topological texture is used solely to refer to the three-dimensional variation, or relief of a physical surface. In contrast the term albedo texture is used to refer only to surface markings. Image texture consists of intensity variations in the image plane and can be due to either topological or albedo texture or combination of the two.

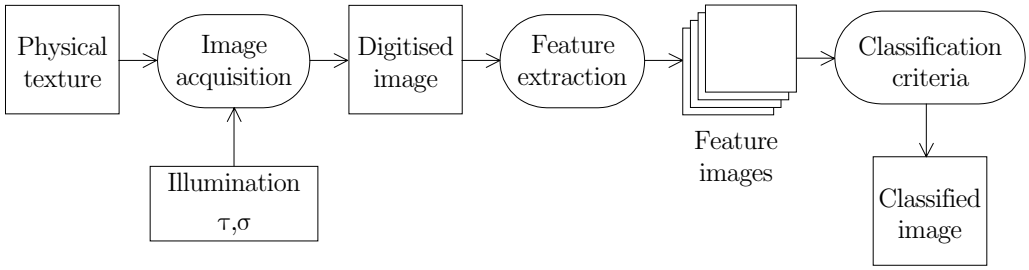


Figure 1.3 Typical processes in texture classification.

1.3 Dissertation organisation

This dissertation may be divided into two main parts. In the first part, comprising chapters 2, 3, and 4, a theoretical background concerning the nature of texture classification is presented, drawing special attention to image formation from physical surface and how to reverse the process towards image prediction. In the second part, comprising chapters 5, 6, and 7, the problem of illuminant dependence is explicitly faced, a consistent solution based on revised theory is suggested, and a methodical evaluation by simulation of the solution is carried out. Figure 1.4 traces the development of the research from theory to solution, and the subsequent evaluation and analysis of results.

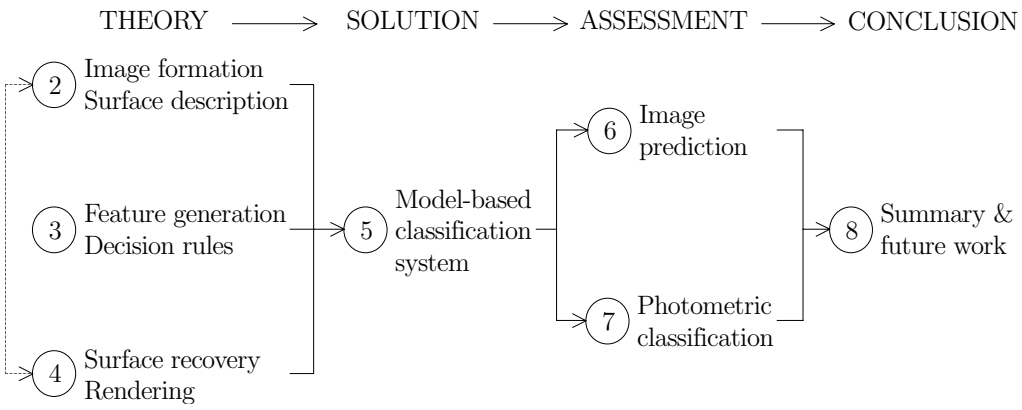


Figure 1.4 Progression of the investigation. Circled numbers symbolise the chapter where each topic can be found.

Explicitly, Chapter 2 is concerned with the image acquisition process; that is, given a scene description consisting of surface, reflectance, light, and imaging device, it describes the formation of an image from the surface. It also introduces two levels of surface description: single parameter description, and histogram representation techniques. Chapter 3 explains the fundamentals of classification, and focuses on the selection of a popular feature measure and discriminant function. Chapter 4 links the learning of Chapter 2 with the purpose of surface recovery, suggests shape from photometric stereo as a good alternative, and identifies the rendering process. Chapter 5 integrates a photometric technique into a complete model-based classification system, detailing the implementation of every component of the system. In Chapter 6 the accuracy of the model-based rendered images is subject to appraisal, evaluating the impact of the assumptions made in previous chapters. Chapter 7 assesses the performance of photometric classification by comparison with that of a naive classifier and best case classification. Finally, in Chapter 8 the work is summarised, final conclusions are drawn, and future work is recommended.

With the object of facilitating the reading, we have spared no effort in trying not to make reference to material in later chapters or sections, but when this has been done an understanding of material in hand does not necessarily require an understanding of that which comes later.

1.4 Original work

It is believed that this dissertation contains new findings concerning the work due to [McGunnigle98], and extends his prior experiments to a larger range of textures and conditions. Specifically, this work makes the following main contributions to texture classification under varying lighting conditions:

- In addition to the use of surface derivative fields, the albedo texture is also considered in the formation of the surface model.

- Detailed implementation of a complete texture classification system which integrates a photometric stereo scheme to reduce misclassification due to varying illuminantion.

- Assessment of image prediction and classification not only with respect to varying tilt angle but also with respect to varying slant angle.

- Determination of an empirical expression which approximately model the error power in the predicted images as a quadratic function of illuminant tilt angle (valid for isotropic surfaces).

- Having predetermined the tilt angle of the photometric image set, an optimal slant angle range, which is averagely better able to predict images of surfaces under any novel lighting conditions, is recommended.

- Development of supporting theory for the evaluation of the model-based classification approach.

- Investigation into the relationship between image prediction inaccuracies and classification errors on the model-based classifier.

Chapter 2

Image acquisition and illumination

This chapter is concerned with the first process of the texture classification system defined in the introductory chapter. The aim is to identify the phenomena underlying the generation of textured image from physical surface in the presence of illumination. Firstly, it describes the characterisation of the incident image onto the imaging device. Secondly, it overviews the main components of the imaging device and the associated degradation of the signal. And finally, it surveys methods for describing attributes of a surface such as roughness.

2.1 Introduction

In the context of this dissertation, the motivation for identifying surface properties and understanding how they affect the image space in the presence of light is threefold:

- Firstly, recognizing surface characteristics such as roughness or isotropy may help us understand the character of the experimental results and their validity.

- Secondly, extraction of surface geometry and reflectance properties is crucial for the success of surface modelling. Chapter 4 will show how

these properties can be extracted and used as input data for creating virtual images.

- Thirdly, understanding the mechanism of light reflection is an important issue for many computer vision algorithms. In particular, for photometry-based methods (photometric stereo, shape from shading, etc), which extract differential shape parameters from the intensity of reflected light, the need for correct interpretation of a pixel brightness is evident.

The nature of the process from physical surface to digitised image[1] may be very well illustrated by Figure 2.1. Such a task is defined as follows: given the scene description — (1) the surface texture and its topology, (2) the surface reflectance function, and (3) the light source position — determine the incident image, that is to say, how the object appears to the viewer. Finally, given the imaging device — (4) position and other parameters — determine the data image.

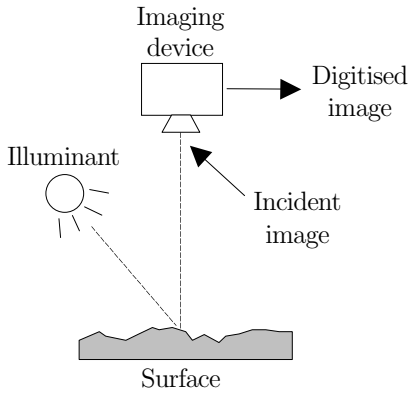


Figure 2.1 Scenario of image acquisition of an illuminated physical surface.

[1] The digitised image is the final result of all the acquisition process, thus this image may be interpreted as a degraded version of the incident image which has been affected by the imaging device. It is important to be aware that this image forms the basis of all subsequent processes, and therefore it is often referred to as data image.

2.2 Characterisation of incident image

The aim of this section is to model the transition from physical surface to incident image. The light intensity of a point (or pixel) in a 2D projection of a physical surface is dependent on:

- The lighting; this is an obvious observation—we are not surprised when a view changes with extra, reduction or modified illumination.
- The reflectance properties of the surface in view; such mirror-like surfaces are called *specular* and in fact the light that they reflect off depends solely on the angle of the incident light. Alternatively, matte surfaces are called *diffuse* and reflect light equally in all directions of an intensity that depends on the angle of incident light.
- Surface orientation; again, it is an obvious observation that inclining a surface will usually alter the light reflected towards an observer.

Herein, the last of these point will not be considered; surface's plane is therefore assumed to always have the same inclination, and specifically it will be held perpendicular to the imaging device (see Figure 2.1).

2.2.1 The reflectance function

The central term in the scene description represented by Figure 2.1 is the *surface reflectance function*. Roughly speaking, it describes how an elementary surface patch reflects the incident light. It is expressed in terms of the *bidirectional reflectance distribution function* (BRDF), which is constructed from ratios of patch radiance to its irradiance. Thus, it is generally a 4-dimensional function

$$BRDF_{\lambda}(\tau_{i},\sigma_{i},\tau_{r},\sigma_{r}) \tag{2.1}$$

describing how much light, incident from direction described by spherical angles (τ_{i},σ_{i}), is reflected into the viewer's direction (τ_{r},σ_{r}). The BRDF of a given

surface is spectral-dependent, denoted by subscript λ, but in practice it is often approximated by independent BRDFs per colour channel for a RGB image. For grey-level images, there is no need to take into account more than one BRDF since only one channel is used.

The phenomenon of reflection can be produced by one or more different physical processes. These underlying processes, which are not of interest here, lead to various generic behaviours, depending on how a surface reflects off the incident light. It is interesting to mention two particular cases which illustrate two extreme behaviours. A perfect mirror, or completely *specular* material, has a reflectance function equal to one in one direction and zero in all others. Conversely, a perfectly matte or *Lambertian* surface reflects equally in all directions, with the amount of reflected light depending only on the incident light. Many surfaces with less simple behaviours can have their reflectance described as having both Lambertian and specular reflectance properties, and are commonly known as *hybrid surfaces*.

As it was previously mentioned, the reflectance surface measures the fraction of emitted light in the observer's direction. However, all the facets reflect the incident light not only towards the viewer, but in other directions as well, including those towards another surface facets. Thus, the light reflected on each facet represents possible source of secondary illumination for other facets. The effects of illuminating a given surface facet by light reflected on other surface facets are called *interreflections* and correspond to the moment when the tight connection of local surface properties with a pixel value disappears; depending on entire surface properties interplay, a pixel value contains some contribution driven by properties of other surface facets. Fortunately, interreflections have usually small contribution to a pixel value, and in almost all applications, it is assumed that they do not exist at all. The consequences resulting from this fact depend on the technique used and the desired accuracy of the task, but at any rate, it always entails errors.

2.2.2 Gradient space

Gradient space is a way of describing surface orientation, and is a concept of general use in many analytical problems. We proceed by noting that at nearly every point a surface has a unique normal, which may be described by a vector $\mathbf{n} = (n_x, n_y, n_z)$. Since only the orientation of the surface is of interest, only the direction of this vector need to be considered, which, assuming $n_z \neq 0$, is given by

$$\left(\frac{n_x}{n_z}, \frac{n_y}{n_z}, 1\right)$$

or, without any loss of information,

$$\left(\frac{n_x}{n_z}, \frac{n_y}{n_z}\right) = (p, q)$$

The pair (p,q) is the two-dimensional gradient space representation of the orientation. Interpreting the image plane as $z = 0$, the origin of gradient space corresponds to the vector $(0,0,1)$, normal to the image, implying a surface parallel to the image plane.

2.2.3 The Lambertian image

The reflection model which is most often used for its simplicity and linearity is the Lambertian model. In intuitive terms, it states that the radiance in a point on a surface falls with the cosine between the surface normal and the viewer direction. Consequently, the camera pixel brightness value is viewpoint invariant due to foreshortening, and is dependent only on the relative relation between surface normal $\mathbf{n}$ and light source $\mathbf{l}$.

Kube and Pentland's model [Kube88] of an illuminated fractal surface provides an analytical expression for the statement mentioned above. If we drop the assumption that the surface is fractal, we may extend their model to any band limited surface with height map $s(x, y)$.

Let $\mathbf{d}(x, y)$ be the derivative field vector of the band limited scalar field

$s(x, y)$ such that

$$\mathbf{d}(x, y) = \nabla s(x, y) = \begin{bmatrix} \frac{\partial z}{\partial x} & \frac{\partial z}{\partial y} & \frac{\partial z}{\partial z} \end{bmatrix} \tag{2.2}$$

If illumination is produced by a point source located an infinite distance from the scene, the magnitude and direction of the vectors will be uniform throughout the scene, hence the illumination vector field is given by

$$\mathbf{l}(x, y) = \begin{bmatrix} \cos\tau \sin\sigma & \sin\tau \sin\sigma & \cos\sigma \end{bmatrix} \tag{2.3}$$

Assume that the surface has a Lambertian reflectance function which is homogeneous over the surface. Furthermore, assume that the surface is no significant affected by cast or self shadowing[2]. Adopting all these restrictions, Kube's model state that the image field is the normalised scalar product of the surface derivative field and the illumination vector field, that is

$$i(x, y) = \frac{\mathbf{l}(x, y) \cdot \mathbf{d}(x, y)}{\left|\mathbf{l}(x, y) \cdot \mathbf{d}(x, y)\right|} \tag{2.4}$$

Thus, the normalised intensity image $i(x, y)$ is given by

$$i(x, y) = \frac{-p(x, y)\cos\tau \sin\sigma - q(x, y)\sin\tau \sin\sigma + \cos\sigma}{(p(x, y)^2 + q(x, y)^2 + 1)^{1/2}} \tag{2.5}$$

where τ and σ are the illuminant vector's tilt and slant angles as defined in Figure 2.2, and

$$p(x, y) = \frac{\partial}{\partial x} s(x, y)$$

$$q(x, y) = \frac{\partial}{\partial y} s(x, y)$$

are the surface partial derivatives with respect to x and y respectively.

---

[2] A cast shadow occurs where one part of the surface prevents another from being illuminated, by blocking the direct path between light source and shadowed area. Self shadowing occurs when a facet is oriented such that it does not present any area on which light is incident.

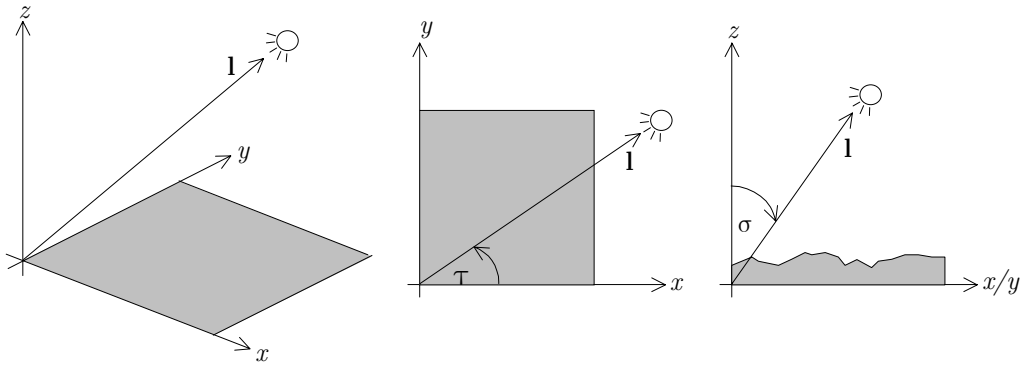


Figure 2.2 Definition of axis and illumination angles: (a) Perspective view, (b) height-map (plan), (c) height-map (elevation).

2.3 The imaging process

Section 2.2 has considered the interaction between rough surface and light to form a textured image. However, strictly, texture analysis algorithms do no operate on incident image but on a data set, which is altered by the so-called *imaging process*. Although it is not our intention to investigate the effects of this process, which was largely evaluated by [McGunnigle98], we must take into consideration the experimental findings if we are to draw more honest conclusions.

2.3.1 Overview

Three are the principal mechanisms which form components in the transition from incident image to data set: system optics, the CCD[3] array, and the frame store (see Figure 2.3). Each of these mechanisms has associated problems of different nature which may degrade the quality of the final data image.

The case of sub-optimal focusing is the predominant problem of the optical system discussed in the literature. It is common to use the term Point Spread Function (PSF) to represent the image resulting from a single point source imaged by the camera. Normally, the PSF is modelled as a two-dimensional

[3] Charge-coupled device, a high-speed semiconductor used frequently in image detection.

Gaussian function $b(x, y)$ [Nayar94] given by

$$b(x,y) = \frac{1}{2\pi\sigma_b^2} \exp\left[-\frac{x^2+y^2}{2\sigma_b^2}\right] \tag{2.6}$$

where σ_b^2 is a spatial constant proportional to the diameter of the blur circle.

Like with any silicon device at ambient temperature, the CCD array is subject to dark current noise which is considered to be stochastic as the generation of an electron is a random event. This effect contribute to both temporal and fixed pattern noise, and is highly dependent on changes in temperature (each 6.7° C rise in temperature doubles the dark current). Another fundamental limitation of all imaging devices is the quantum nature of light which may be described as shot noise. Shot noise is considered as being temporal, conforming to a Poisson process which tends towards a Gaussian distribution (with variance equal to its mean) as the irradiance of electrons increases.

The frame store is concerned with the transfer of discrete image intensity to numerical representation. The process involves a serial read-out of the rows, with low pass filtering of the pulse train and addition of synchronisation signal. Afterwards, the signal is stripped of synchronisation information, resampled and quantised before being stored in memory. The following three problems are typically associated with this process:

- Serial filtering affects the transfer function, but the effect may be neglected if optical blur prevails in the system.

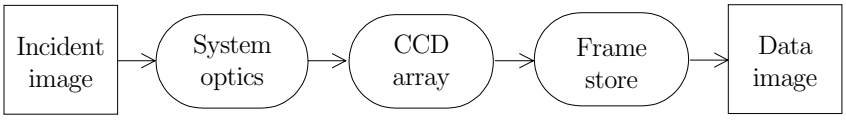


Figure 2.3 Mechanisms under the imaging process.

- If the rows are not perfectly synchronised, sampling points will be out of alignment with those of adjacent rows (the error is reported to be within the bounds ± 0.4 pixels) [Wang96].

- For a 8-bit video analogue to digital converter (ADC), a random error is uniformly distributed between $\pm 1/512$ of the full scale deflection. The quantisation noise is normally treated as a white noise process with variance $0.5^2 / 12$ [Healey94].

2.3.2 Implications for texture classification

Regarding the evaluation carried out by [McGunnigle98], blurring was found to be the dominant artefact, and a Gaussian model of the imaging system transfer function was adopted. He also noted that the common assumption of white noise is not completely justified, at least without down-sampling.

In any case, the observed effects of blur and white noise discourage the use of high frequencies as consistent sources of discriminatory information. While the blur function reduces the feature mean and variance, particularly at high frequencies, the additive white noise increases the texture's variance in the feature space. The magnitude of degradation in classification will depend on the proximity of the textures' statistics in the feature space as well as the level of noise itself.

It is also worthwhile to notice that while the absolute power of noise seems to be similar in all images, the power relative to the texture's variance may be significant depending on which texture is used. Consequently, it is not recommended to apply any technique of spectral subtraction such as Wiener filtering prior to classification.

2.4 Surface description

A description of a surface can be made on several levels. A single parameter may be sufficient to characterise a surface for some purposes, whereas in other cases a much greater degree of description is required. The descriptors introduced here will form the basis for modelling all the surface textures used throughout this dissertation. A first level of description seeks to estimate some property of the surface, e.g. height or gradient, with a single parameter. On a second level, a statistical model such as the histogram, is applied to the variation of height or gradient, which provides a more visual comprehension of the surface's characteristics. At a third level, there are those techniques which incorporate spatial interaction such as the Power Spectral Density (PSD) or the Autocorrelation Function (ACF). This third level of representation will not be considered here.

2.4.1 First order statistics

The most basic form of profile description requires the use of only one parameter. Two of the most common measures of roughness are the *root mean square roughness* σ_s and the *centre line average* or *average roughness* R_{cla}. Both of them are defined with respect to surface profile (see Figure 2.4) in either x or y axis; considering, for instance, the former

$$\sigma_s = \sqrt{\frac{1}{n}\sum_{x=0}^{n-1}\Big[s(x) - \overline{s(x)}\Big]^2} \tag{2.7}$$

$$R_{cla} = \frac{1}{n}\sum_{x=0}^{n-1}\big|s(x)\big| \tag{2.8}$$

where $s(x)$ represents the height of the surface at a point x along the profile, and n is the number of columns of the surface height map.

However, this work concentrates on real surfaces and implicit in the definition is the necessity of knowing or experimentally calculating $s(x, y)$, the surface height map. This is a complicated task with some associated problems,

as the measures of these parameters depend on the instrument and the separation of sampling points [Bennet89].

More suitable in the context of this dissertation is the use of parameters that concentrates on the slope of the facets rather than its height. Of particular interest is one parameter known as *absolute average slope ratio* (AASR), which provides an easy way to evaluate the level of irregularities of a certain surface. Understanding that we own a method to estimate the derivative fields of the surface, $AASR$ may be calculated as

$$AASR = \frac{1}{2n} \sum_{x} \sum_{y} \left|\hat{p}(x,y)\right| + \left|\hat{q}(x,y)\right| \tag{2.9}$$

where n is the number of samples contained in each partial derivative. This parameter will be used in Chapter 6 to characterise the degree of roughness of a given surface texture.

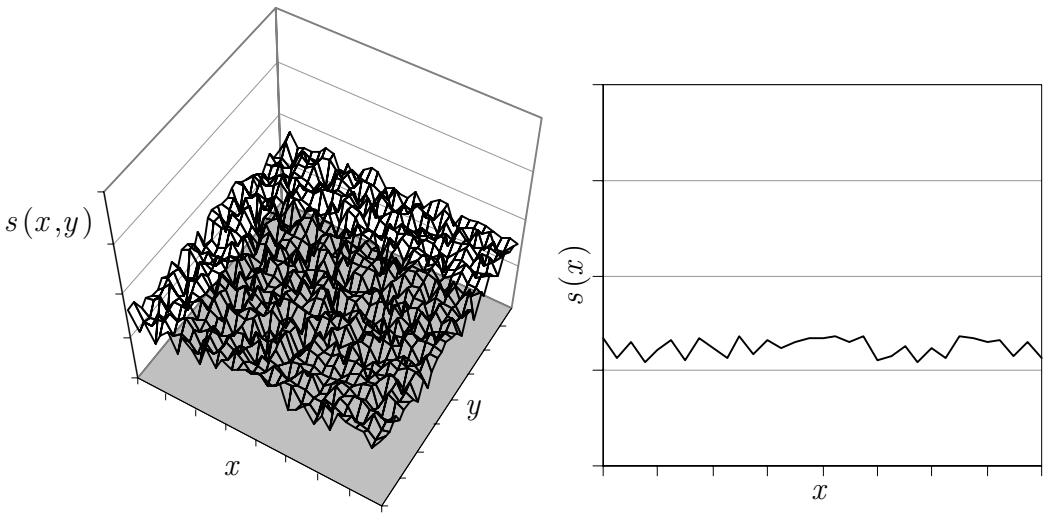


Figure 2.4 **(a)** Surface height map of a rough surface; **(b)** corresponding profile of a vertical cross section.

2.4.2 Histograms

The histogram description can be thought as an extension of the single parameter descriptor; while the parameters in Section 2.4.1 concentrate in estimating only the standard deviation or mean of the heights of the surface, the histogram represents a statistical model of height distribution. In some ways the use of histogram description not only implies a new degree of visual discrimination between surfaces but also a certain degree of modelling.

The histogram does give an insight into the nature and history of a surface. For example, surfaces which are the result of a large number of random events tend to approximate to Gaussian distributions. Natural phenomena such as abrasion, erosion, and friction, have the tendency to wear down and deform summits leaving valleys largely unaffected, which is reflected in a histogram taking on a negative skew [Bennett89]. Some milling operations and the presence of relatively large particulate matter on an otherwise smooth surface result in a histogram taking on a positive skew. However, as we work on the basis of photographic images, the presence of non-symmetrical statistical distributions is generally attributable to artefacts in the sampling process.

Chapter 3

The classification process

It has been already mentioned that texture classification, as proposed in this dissertation, normally involves three processes: image acquisition, feature generation, and discrimination. Chapter 2 examined the first of them, where the texture under consideration is illuminated and its image captured. This chapter concentrates on the other two processes, which constitute the classification process proper. In a sense, the nature of classification is concerned with the selection of suitable features which will allow different classes to be separated, gathering statistics from these features, and finally assigning probabilities and thus classifying each area.

3.1 Introduction

Classification is such a broad ranging field that a description of all the algorithms could fill several volumes of text. The theory of classification, and pattern recognition in general, is fully discussed in several references such as [Devijver86, Pavel89, Fukunaga90]; here only fundamental ideas will be given.

No recognition is possible without knowledge. Decisions about groups into which recognized objects are classified are based on such knowledge. Experience shows that a good knowledge representation design is the most important part of solving the classification problem. Normally, a certain description can be used for representing knowledge as a part of a more complex representation structure. Descriptions usually depict some properties of objects and are called

features. Typically, a single description is insufficient, therefore the descriptions are combined into *feature vectors* which are inputs for statistical classification techniques.

In the context of texture classification, the objects are textured images which can be divided into disjoint subsets that, from the classification point of view, have some common features and are called *classes*. The definition of how the images are divided into classes is ambiguous and depends on the classification goal. The number of classes is usually known beforehand, and typically can be derived from the problem itself. Nevertheless, there are also approaches, called cluster analysis, in which the number of classes is unknown.

The combination of algorithms which assign classes to textures is called the *classifier*; as refer to in this dissertation, the classifier consist of

1. a means of extracting the relevant image components,

2. a mechanism to process these components, and

3. a discriminatory method to classify on the basis of this information.

Even though (1) is particularly refer to as texture measure and (2) is refer to as post-processing, in many occasions both of them are indistinctly gathered within the feature generation system. This feature generation process is discussed in Section 3.3, while fundamentals concerning (3) are examined in Section 3.4. First, next section introduces a brief theoretical background of texture classification drawing special attention to terminology and functionality.

3.2 Texture classification

3.2.1 Brief description

In its simplest form, classification is the process of assigning similar regions of an image together. Texture classification includes a wide range of decision-theoretic approaches to the identification of images (or parts thereof). All

classification algorithms are based on the assumption that the image in question depicts one or more features (e.g. geometric parts in the case of a manufacturing classification system, or spectral regions in the case of remote sensing). Each of these features should belong to one of several distinct and exclusive classes.

There are two types of classification, namely *supervised* and *unsupervised*. In the former, classes are specified a priori by an analyst; in the latter, classes are automatically clustered into sets of prototype classes—where the user merely specifies the number of desired categories. In the following text, all the texture classification tests reported are of the supervised type and this process will be referred to simply as classification.

3.2.2 How it works

Texture classification analyses the numerical properties of various texture features and organises the data into categories. Classification algorithms typically employ two phases of processing: *training* and *testing*. In the initial training stage, characteristic properties of typical texture features are isolated and, based on these, a unique description of each classification category, i.e. *training class*, is created. In the subsequent testing stage, these feature-space partitions are used to classify texture features.

The description of training classes is an extremely important component of the classification process. In supervised classification, statistical processes[1] or distribution-free processes can be used to extract class descriptors. Unsupervised classification relies on clustering algorithms to automatically segment the training data into prototype classes. In general, unsupervised clustering techniques are used less frequently, as the computation time required for the algorithm to learn a set of training classes is usually prohibitive. However, in applications where the features or/and relationships between them

[1] Unlike distribution-free processes, statistical processes are based on a priori knowledge of probability distribution functions. They are commonly refer to as parametric methods whereas distribution-free processes are also known as non-parametric methods.

are not well understood, clustering algorithms can provide a viable means for partitioning a sample space. In either case, the motivation criteria for constructing training classes is that they are:

- independent—a change in the description of one training class should not change the value of another,
- discriminatory—different texture features should have significantly different descriptions, and
- reliable—all texture feature within a training group should share the common definite descriptions of that group.

A convenient way of building a parametric description is via a feature vector, $\mathbf{f} = (f_1, f_2, \ldots, f_n)$ where n is the number of attributes which describe each texture feature and training class. This representation allow us to consider each texture feature as occupying a point, and each training class as occupying a subspace within the n-dimensional space. Viewed as such, the classification problem is that of determining which subspace each feature vector belongs to.

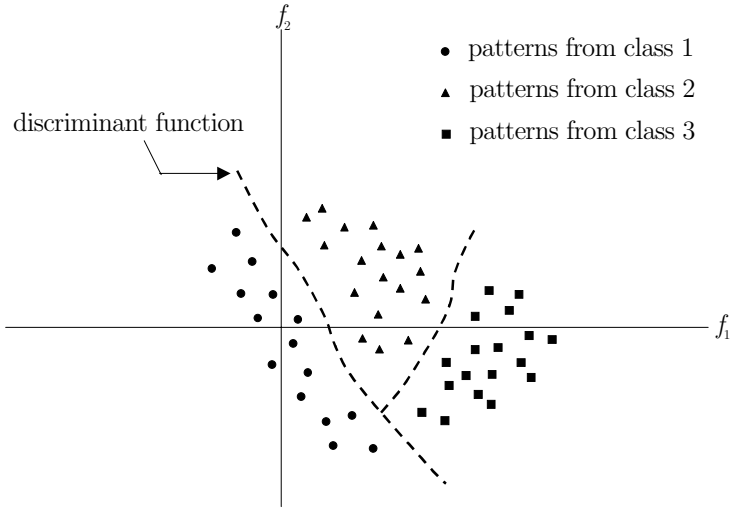


Figure 3.1 2-D feature space representation of hypothetical data. Each training class forms clusters, i.e. a representative point (or mean) surrounded by some spread (or deviation).

For example, consider an application where we must distinguish among three different types of textures based upon a set of two attribute classes. If we assume that we have a vision system capable of extracting these two features from the set of training images, we can plot the result in the 2-D feature space shown in Figure 3.1.

At this point, we must decide how to numerically partition the feature space so that if we are given the feature vector of a test image, we can determine, quantitatively, to which class belongs. To achieve this, a set of discrimination rules should be applied to the probability density functions of the dataset.

3.2.3 Differences between classification and segmentation

As classification and segmentation are sometimes misunderstood, it is helpful to clarify what these terms, as used in this text, refer to. Classification and segmentation have in fact closely related objectives as the former is another form of component labelling that can result in segmentation of various features in a scene. Precisely, texture segmentation is used to refer to the process of dividing an image up into homogeneous regions according to some *homogeneity* criterion. It is therefore intimately concerned with establishing the boundaries between these regions without regard to the type of class. On the other hand, texture classification refers to the process of grouping test samples of texture into classes, where each resulting class contains similar patterns according to some *similarity* criterion.

Note that classification tests may be performed on separate samples of textures, in which case segmentation is not required. Alternatively, a single image containing multiple textures may be presented, requiring segmentation prior to classification[2]. In this text, we define the final purpose of our system to be classification on multi-textured images. Hence, segmentation is considered to be the global effect of the classification process occurring at pixel level.

---

[2] If classification is performed on a pixel by pixel basis within a single multi-textured image then, as a by-product, segmentation also occurs.

3.3 Feature extraction

Before either classification or segmentation can take place, some homogeneity or similarity criterion must be defined. These criteria are normally specified in terms of a set of *features measures*[3], which each provides a quantitative measure or quality of a certain texture characteristic. In a multiclass problem it is habitual to use more than just one feature; groups of features associated with each pixel are treated as being orthogonal and assembled together as a *feature vector* $\mathbf{f}(x, y)$.

Feature extraction is concerned with the detection and localization of particular image patterns which represent significant features of the image. These important features are dependent on the application and they are generally of two different origin: a global image property or a region of the image with a relevant measurable property.

There are many researchers in image processing and computer vision areas who have considered the concept of feature measure to cope with texture classification and segmentation problems. In the area of database retrieval, for instance, texture features are often used to search an image database to find similar images to the image submitted by the user [Ashley95][Pentland96]. In remote sensing and radar applications, texture features have been used to identify forest regions and their boundaries, and to identity and analyse various crops [Fukuda99][Wu99]. The use of feature textures have been even reported in analysing seismic signals [Randen98].

3.3.1 Overview of feature measures

Many types of representations and features have been proposed in the past few decades, attempting to maximise the classification task with a minimal set of compact discriminates. Classically, two major categories of texture measure methods has been identified: structural and statistical [VanGool85][Wechsler80].

[3] These *feature measures* are alternatively referred to in this document as *texture measure* or just simply *features*.

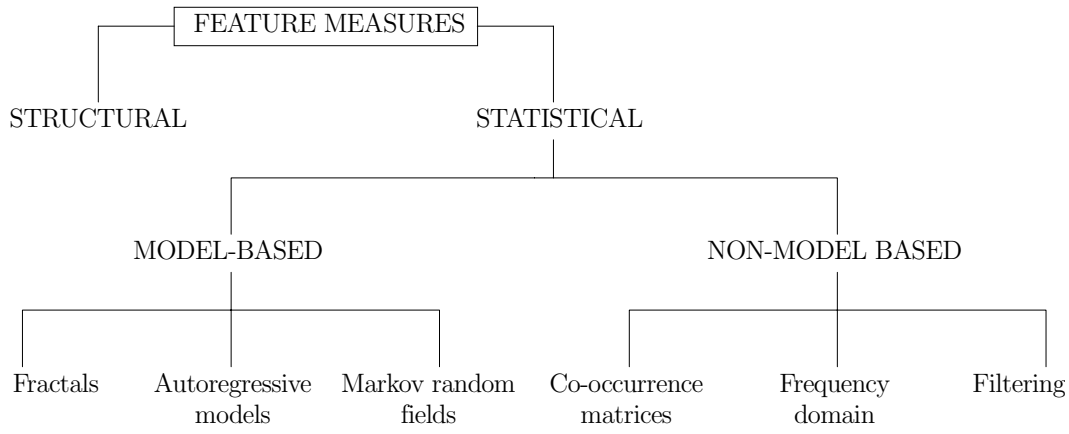


Figure 3.2 Major categories of features for texture identification.

Structural techniques use primitives to describe both texture elements and placement rules between them. This approach is therefore better suited to textures that exhibit a regular macrostructure, i.e. composite textures where primitives placed at various locations can be recognized. On the contrary, statistical techniques are more appropriate for micro textures in which there is no obvious primitive pattern. That is the case of the kind of textures used in this research, thus structural approaches will not be considered further.

Figure 3.2 shows an incomplete taxonomy of texture measure methods drawing special attention to statistical approaches. We distinguish between model-based features, which are based upon parameter estimation techniques, and non-model based features. In the former, a number of random field processes are used for modelling and synthesis of texture. If this model is shown to be capable of representing a range of textures, then its parameters may provide a suitable feature set for classification or/and segmentation of the textures. Popular model-based approaches are Markov random fields, autoregressive models (e.g. AR, SAR, MA, ARMA, fractional difference), and fractals. Non-model based techniques comprise a wider range of features of different nature; they include frequency domain analysis (e.g. FFT, Wigner-Ville), signal processing approaches (Laws' masks, Gabor functions, wavelets),

and pure statistical methods (e.g. grey-level co-occurrence matrices, grey-level differences).

The criteria used for the selection of features are: (i) sensitiveness to changes in illuminant slant and tilt, and (ii) popularity in the literature. In the last few years, signal processing approaches have been one of the most common and productive areas of research in the context of classification. They normally employ directional FIR filters designed exclusively for the purposes of discriminating between textures, and indeed are affected by changes in lighting conditions. It is our impression that they are suitable for the framework of this investigation and therefore considered in next section.

3.3.2 Filtering for texture measure

A common denominator for most signal processing approaches is that the textured image is submitted to a linear transform, filter, or filter bank, followed by some energy measure [Randen99] (Figure 3.3). Due to the inherent similarities between these approaches, they will be referred to as filtering approaches.

The objective of the energy function is to estimate the energy of the filtered image in a local region. Unfortunately, as [Gabor46] formally proved for 1D signals and [Daugman85] extended to the 2D case, a signal's specificity simultaneously in both spatial and frequency is limited by a lower bound on the product of its frequency bandwidth and spatial extent. Therefore, accurate edge preservation and energy estimation are conflicting goals. For edge localization, high spatial resolution is desired, whereas for energy estimation, high spatial frequency resolution is desired. This trade-off need to be balanced in the smoothing filter. The design of the smoothing filter as well as the non-linearity will be discussed later in Chapter 5.

Regarding the second non-linearity in Figure 3.3, [Unser90] studied several combination of the first and second non-linearities. He concluded that squaring together with a logarithmic normalising non-linearity was the best combination.

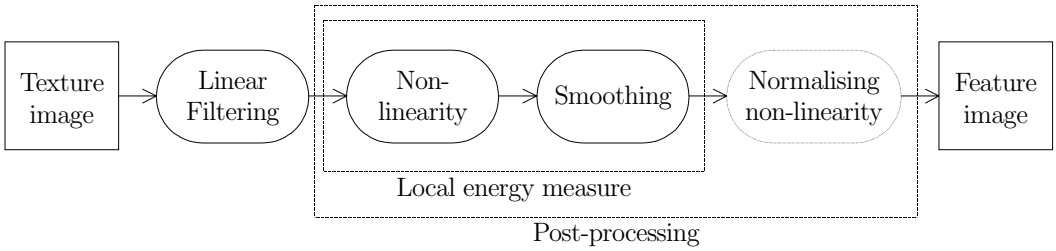


Figure 3.3 Typical block diagram of filtering approaches for feature extraction.

Nevertheless, this second non-linearity is hardly ever used, and hence it will be not considered.

In the literature, many different choices for all the components of Figure 3.3 have been reported. In any case, it is always desirable to achieve the best possible performance with the smallest possible number of features in order to keep low computational and storage requirements. This issue is directly linked with the design of the filtering stage, and specially the filter bank in a multichannel approach. Therefore, we are of the belief that filtering is certainly the most important element.

One of the pioneering filtering techniques was the approach by Laws [Laws79], where a bank of band pass filters was applied. Despite their simplicity and lack of theoretical background, they sometimes form a highly effective approach, which in few cases have a classification performance comparable with modern techniques. Their ease of implementation and very modest computational expense have make them a popular choice for many researchers as a test bed for novel classification systems and practical applications.

Subsequent works have focused on different filter bank families, different subband decompositions, and optimisation of the filter for texture feature separation. One family of functions, the 2D Gabor family, has particularly become an important and popular area of research within texture analysis thanks to two important properties:

- the space/frequency characteristic of the filter, and
- the similarity to the operation of the early human visual system.

The complex form of Gabor filters achieved the lower limit for the joint entropy, or uncertainty, of the resolution for orientation, spatial frequency and 2D spatial position. They are therefore of interest to analysts attempting to localise textures in the spatial domain using their spectral characteristics. Furthermore, [Daugman85] presented evidence that the 2D receptive-field profiles of simple cells in mammalian visual cortex are well described by member of this optimal 2D filter family.

In conclusion, it is believed that Gabor filters can be easily integrated into our system and provide the following advantages:

- they can be used to extract and resolve magnitude and phase components of the image, thus making explicit the use of magnitude or power at the expense of phase;
- they can be well defined both in the spatial and the polar frequency domain, so they may be implemented in the frequency domain in order to avoid costly convolutions; and
- because of their popularity, by establishing the tilt and slant dependency of the Gabor-based classifier, the results of this work will be relevant to a wide area of texture research.

3.3.3 Gabor functions

The Gabor elementary functions (GEF's) were first defined by Dennis Gabor in his classic monograph on the "Theory of Communication" [Gabor46]. Daugman was the first to extend the definition to the two-dimensional space [Daugman85]. The family of 2D Gabor functions is given by

$$g(x,y) = e(x',y') \cdot \exp\left[j2\pi(u_0 x + v_0 y)\right] \tag{3.1}$$

where $(x', y') = (x\cos\phi + y\sin\phi, -x\sin\phi + y\cos\phi)$ represent rotated spatial-domain rectilinear coordinates. Letting (u, v) denote frequency-domain rectilinear coordinates, (u_0, v_0) represents a particular 2D frequency. The function $e(x, y)$ is a 2D Gaussian envelope

$$e(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right] \tag{3.2}$$

where σ_x and σ_y characterize the spatial extent and bandwidth of the filter. Therefore, the GEF is an elliptical Gaussian with aspect ratio (σ_x / σ_y) multiplied by a complex exponential representing harmonic modulation with radial centre frequency $\omega_0 = (u_0^2 + v_0^2)^{1/2}$ and orientation $\theta_0 = \tan^{-1}(v_0 / u_0)$.

In most cases, letting $\sigma_x = \sigma_y$ is a reasonable design choice; then ϕ need not to be specified since $e(x, y)$ is circularly symmetric. However, Dunn et al. have found that when a texture contains texture not arrange in a square lattice, asymmetric filters are useful [Dunn95]. In this dissertation, we assume asymmetric filters $(\sigma_x \neq \sigma_y)$; however, we do not consider the rotation of the coordinates $(\phi = 0)$ because it simplifies the implementation and its application is not clearly defined. Hence, the GEF simplifies to

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right] \cdot \exp\left[j2\pi(u_0 x + v_0 y)\right] \tag{3.3}$$

It can be shown that the Fourier domain representation of Eq.(3.3) is given by

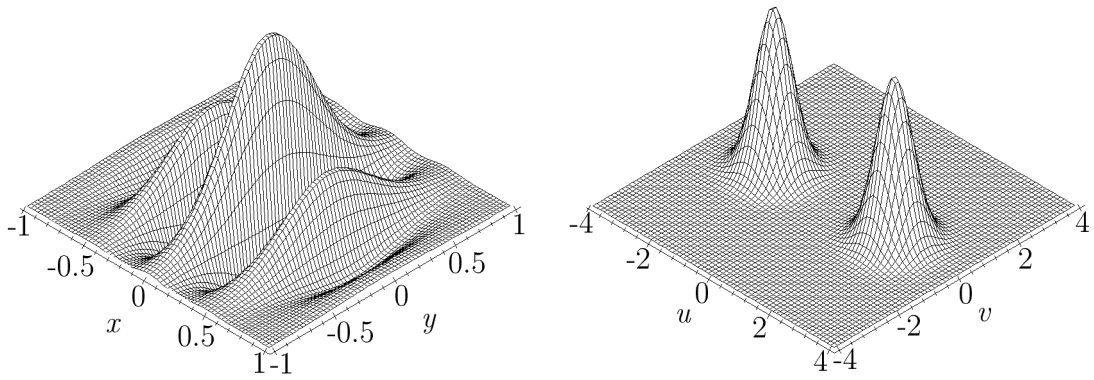
$$G(u,v) = \exp\left[-\frac{1}{2}\left(\frac{(u-u_0)^2}{\sigma_u^2} + \frac{(v-v_0)^2}{\sigma_v^2}\right)\right] \tag{3.4}$$

where $\sigma_u = 1/(2\pi\sigma_x)$ and $\sigma_v = 1/(2\pi\sigma_y)$. The Fourier transform specifies the amount by which the filter modifies or modulates each component of the input image. Such representations are therefore referred to as modulation transfer

functions (MTF).

It is common in texture analysis to use pairs of Gabor filter with quadrature phase relationships (cosine and sine parts). [Daugman85] gives some justification to that fact, arguing that actual filter impulse-response functions and neural receptive-field profiles are real functions, which can be regarded as containing in quadrature both the even- and the odd-symmetric versions. If $g(x,y)$ is in pure cosine phase, then its transform can be obtain as $½[G(u,v)+G(-u,-v)]$; whereas, if $g(x,y)$ is in pure sine phase, then its transform is determined by $½[-jG(u,v)+jG(-u,-v)]$. Intermediate phases represented by the appropriate mixture of these cases would generate filter profiles with neither symmetry nor anti-symmetry.

A plot of a representative even-symmetric member of the 2D Gabor family is shown in Figure 3.4. Referring to the parameters in Eq.(3.3), this particular filter has $\sigma_x^2 = \sigma_y^2 = 1/8$, $u_0 = 2$, and $v_0 = 0$ cycles/image-width. Observe that the number of significant sidelobes in the spatial filter profile inversely determines the filter's spatial-frequency bandwidth and orientation bandwidth; the spatial periodicity and orientation of the lobes specifies the filter's preferred spatial frequency and orientation.



(a) **(b)**

Figure 3.4 (a) An even-symmetric Gabor filter with unity aspect ratio and centroid located at (0,0), **(b)** corresponding MFT.

3.4 The discriminant function

3.4.1 Theoretical framework

In statistical terms, the discriminant function is that of several variates used to assign items into one of two or more groups. The function for a particular set of items is obtained from measurements of the variates of items which belong to a known group.

If a discrimination curve (or hypersurface in a multidimensional feature space) exists which separates the feature space such that only objects from one class are in each separated region, the problem is called a classification task with *separable classes*. If the discrimination hypersurfaces are hyperplanes, then it is called a *linearly separable* task. Intuitively, we may expect that separable classes can be identified without errors. However, the majority of texture classification problems do not have separable classes in which case the locations of the discrimination hypersurfaces in the feature space can never separate the classes correctly, thus some areas will always be misclassified.

One of the most simple (although not the most computationally efficient) method is to employ a supervised, distribution-free approach known as the *minimum mean distance classifier*. This classifier works well when the distance between means is large compared to the spread of each class with respect to its mean. It is simple to implement and is guaranteed to give and error rate within a factor of two of the ideal error rate—obtainable with the statistical, supervised *Bayes classifier*. The Bayes classifier is a more informed algorithm as the frequencies of occurrence of the features of interest are used to aid the classification process. Without this information the minimum mean distance classifier can yield biased classifications, thus a Bayes classifier will be adopted.

3.4.2 A Bayes classifier

Before dealing with the Bayes classifier principles, some nomenclature used in this section is introduced. A statistical classifier is a device with n inputs

and 1 output. Each input is used to enter the information about one of n features $f_1, f_2, \ldots, f_n$ that are measured from a classified object. An R class classifier will generate one of R groups or *class identifiers* $G_1, G_2, \ldots, G_R$ as an output, which may be interpreted as a decision about the class of the processed object. The function $d(\mathbf{f}) = G_r$, known as *decision rule*, divides the feature space into R disjoint subsets K_r, $r = 1, \ldots, R$ each of which includes all the feature representation vectors $\mathbf{f}$.

The Bayes classifier is a simple statistical classifier, relatively straightforward to implement, and offers reasonable performance [Linnett91][Clarke92]. Bayes rule provides the basis for probabilistic classifiers that seek to minimise the *Total Error of Classification* (TEC) [James85][Tou74]. The goal of this rule may be expressed as follows:

Assign the pixel with feature vector $\mathbf{f}$ to group G_r for which

$$P(G_r|\mathbf{f}) > P(G_s|\mathbf{f}) \quad \forall s \neq r \tag{3.5}$$

where $P(G_r|\mathbf{f})$ is the conditional probability that the pixel with feature vector $\mathbf{f}$ belongs to group G_r. Unfortunately these conditional probabilities are difficult to obtain. A posteriori probability may be expressed by means of a priori probability using the Bayes theorem

$$P(G_r|\mathbf{f}) = \frac{P(\mathbf{f}|G_r)P(G_r)}{P(\mathbf{f})} \tag{3.6}$$

Thus the maximum likelihood classification rule is defined in terms of conditional probabilities, where $P(\mathbf{f} \mid G_r)$ is the probability of a pixel from group G_r having a feature vector of $\mathbf{f}$, and $P(G_r)$ is the a priori probability of a pixel belonging to group G_r.

We now require a suitable representation for the class conditional probability $P(\mathbf{f} \mid G_r)$. One of the ways to achieve this representation is to assume that $P(\mathbf{f} \mid G_r)$ has a parametric form. The most common model is the multivariate Gaussian distribution, which may be expressed as

$$P(\mathbf{f} \mid G_r) = \frac{1}{(2\pi)^{n/2} \left|\mathbf{C}_r\right|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{f}-\mathbf{m}_r)^T \mathbf{C}_r^{-1}(\mathbf{f}-\mathbf{m}_r)\right] \tag{3.7}$$

where

n is the number of feature measures,

$\mathbf{f}$ is the $n \times 1$ feature vector of sample to be classified,

$\mathbf{C}_r$ is the $n \times n$ covariance matrix of group G_r, and

$\mathbf{m}_r$ is the $n \times 1$ vector of feature measure mean for group G_r.

Substituting Eq.(3.6) and Eq.(3.7) into Eq.(3.5), and taking natural logarithms gives the following rule:

$$\begin{aligned} &-\ln\left|\mathbf{C}_r\right| - (\mathbf{f}-\mathbf{m}_r)^T \mathbf{C}_r^{-1}(\mathbf{f}-\mathbf{m}_r) + 2\ln(P(G_r)) > \\ &-\ln\left|\mathbf{C}_s\right| - (\mathbf{f}-\mathbf{m}_s)^T \mathbf{C}_s^{-1}(\mathbf{f}-\mathbf{m}_s) + 2\ln(P(G_s)) \quad \forall s \neq r \end{aligned} \tag{3.8}$$

The terms in the left-hand side of Eq.(3.8), with the exception of the a priori group probability[4], are often collected together in one function $d_r(\mathbf{f})$ called the *discriminant function*. Hence,

$$d_r(\mathbf{f}) = -\frac{1}{2}\ln\left|\mathbf{C}_r\right| - \frac{1}{2}(\mathbf{f}-\mathbf{m}_r)^T \mathbf{C}_r^{-1}(\mathbf{f}-\mathbf{m}_r) \tag{3.9}$$

Expanding Eq.(3.9) gives

$$d_r(\mathbf{f}) = -\frac{1}{2}\ln\left|\mathbf{C}_r\right| - \frac{1}{2}\mathbf{f}^T \mathbf{C}_r^{-1}\mathbf{f} + \mathbf{f}^T \mathbf{C}_r^{-1}\mathbf{m}_r - \frac{1}{2}\mathbf{m}_r{}^T \mathbf{C}_r^{-1}\mathbf{m}_r \tag{3.10}$$

which is known as the *quadratic discriminant function* (due to the $\mathbf{f}^T\mathbf{C}_r^{-1}\mathbf{f}$ term). If the covariance matrices of all classes are identical then the quadratic and natural logarithm terms may be eliminated to give a *linear discriminant*

$$d_r(\mathbf{f}) = -\frac{1}{2}\mathbf{m}_r{}^T \mathbf{C}_r^{-1}\mathbf{m}_r + \mathbf{f}^T \mathbf{C}_r^{-1}\mathbf{m}_r \tag{3.11}$$

The value of a discriminant function for a sample $\mathbf{f}$ is known as

[4] In practice the group probabilities are unknown, so they are considered equal and may be ignored.

discriminant score, and is a univariate number which represents the likelihood of membership of the sample to that class. Note that this value is not a probability, due to the cancelling of common terms and the taking of logarithms from the original expression.

In sum, assuming equal a priori probabilities, the classification rule may be stated as follows: assign the pixel with feature vector $\mathbf{f}$ to the group G_r with the highest discrimination score $d_r(\mathbf{f})$. Therefore, the design of the decision rule must satisfy the following formula for all $\mathbf{f} \in K_r$ and for any $s \in [1, R],\ s \neq r$

$$d_r(\mathbf{f}) > d_s(\mathbf{f}) \tag{3.12}$$

This analysis effectively partitions the feature space into different regions of class membership (see Figure 3.5). The discrimination hypersurface between class regions, e.g. K_i and K_j, is defined by

$$d_r(\mathbf{f}) - d_s(\mathbf{f}) = 0 \tag{3.13}$$

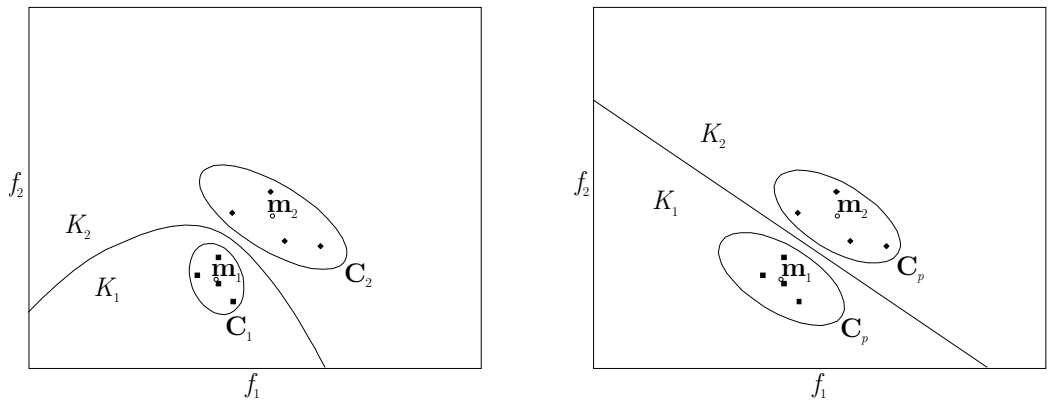


Figure 3.5 Decision surface between two class regions $(R = 2)$ in a two-dimensional feature space $(n = 2)$: **(a)** Quadratic Bayes classifier (different covariance matrices), **(b)** Linear Bayes classifier (both groups have the same pooled covariance $\mathbf{C}_p$).

The simpler linear discriminant function will be used herein as it is straightforward to implement and because of its reported robustness and performance [James85]. It assumes a multivariate normal distribution defined by Eq.(3.7) and identical covariance matrices $\mathbf{C}_r$. However, these covariance matrices are normally slightly different and therefore they are often replaced by the so-called pooled covariance matrix $\mathbf{C}_p$, in which each element is the average of the corresponding elements of the individual group covariance matrices $\mathbf{C}_r$.

3.4.3 Practical considerations

In practice, the population parameters of $P(\mathbf{f} \mid G_r)$ will be unknown, and sample estimates from the training set need to be obtained. The maximum likelihood estimates for the mean and covariance matrix of a multivariate normal are given respectively as

$$\hat{m}_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij} \tag{3.14}$$

$$\hat{C}_{jk} = \frac{1}{N-1}\sum_{i=1}^{N}(x_{ij} - \hat{m}_j)(x_{ik} - \hat{m}_k) \tag{3.15}$$

where

N is the number of samples (of a training set for a particular feature),

x_{ij} is the ith sample of jth feature,

$\hat{m}_j$ is the estimate of the mean for jth feature, and

$\hat{C}_{jk}$ is the estimate of the covariance between the jth and kth features.

Discriminant functions based upon sample estimates are known as *plug-in estimators*. Since the sample estimates of the parametric model are themselves random variables, they are not guaranteed to satisfy Bayes rule, and hence they will not necessarily yield minimum error. However, such discriminant functions are still useful, and the linear discriminant function is particularly robust in operation when the data is not ideally normal [Linnet91].

Chapter 4

Surface recovery and rendering

The main components of a standard classification system has been presented thus far. This system, just as it is, will suffer misclassification due to changes in illuminant direction. Having observed the effect in Chapter 1, the idea is to incorporate into the system surface modelling and rendering techniques to overcome such errors. This chapter overviews techniques used in computer vision to extract surface shape and optical parameters. It focuses on shape from photometric stereo, model-based rendering, and how to combine both methods to produce images of a surface under novel lighting conditions.

4.1 Introduction

The recovery of shape and material information from images is extremely challenging and is far from being solved in complete generality. A number of approaches that promise to be useful, albeit not completely general, have been pursued over a long time. The primary focus of computer vision has been to develop lighting models and rendering procedures to produce synthetic images, which are visually and measurably indistinguishable from real-world images. Computer models of architectural scenes have been especially popular subjects, and have proven to be entertaining virtual environments as well as valuable visualization tools.

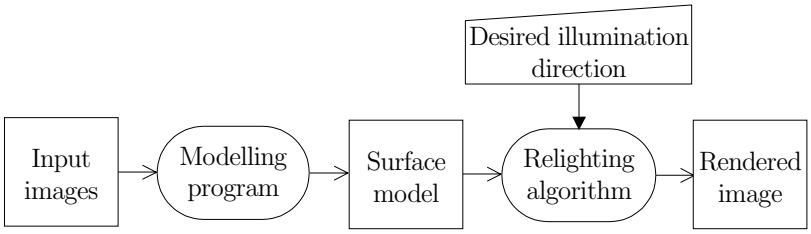


Figure 4.1 Block diagram of a model-based rendering system.

While computer vision usually aims to derive a 3D scene description from its 2D images, it is important to clarify that our approach is not in that direction. Certainly, it is aimed to reverse the process of surface recovery towards image prediction, i.e. use a surface model to render textured images of that surface under novel lighting conditions; afterwards, these images will become the input data for the classifier training stage and, if accurate enough, will provide more reliable decision rules, and therefore better classification results.

Figure 4.1 shows a schematic of how our approach combines modelling and rendering in order to produce images of a surface under a desired illuminant direction. The first part of the system seeks light reflection phenomena and reflectance models that could better capture the relation intensity-shape. The second part uses that model to render the appearance of the surface for a given illumination direction.

Chapter 2 analysed the physical processes of image formation in order to understand how intrinsic surface properties propagate into visual appearance of the surface. We learnt that the interaction between surface and light is completely determined by the surface properties, namely *surface geometry* and *surface optical parameters*. The goal of next section is to find how such properties can be determined by processing of the captured images. We anticipate that the extraction of surface properties is highly non-trivial. The visual appearance of a surface varies with both the viewpoint and the light-point position. In addition, the task is complicated by interreflections.

4.2 Surface recovery

4.2.1 Shape from X

Different methods to extract important information in images have been largely studied for surface recovery purposes during the last three decades. Such techniques are commonly known as 'Shape from X', where X is one of a number of options. This family of techniques includes methods such as shape from motion, shape from stereo, shape from focus, shape from contour, and shape from intensity[1]. The last of these is particularly appealing since it does not require additional hardware beyond that used for classification. Shape from intensity extracts shape information from a series of intensity images, assuming each is generated by a single light source. It can be further divided into three subcategories, namely shape from shading, shape from photometric stereo, and shape from photometric sampling. Basically, the differences between them are in the number of images and arrangement of light sources.

Shape from shading uses a single light source, i.e. one image as input, to recover the shape information [Healey88, Pentland84, Horn75]. It has the advantage that requires the least amount of input, however it also introduces evident disadvantages. Since it has less image information available, it is less accurate: at each pixel, intensity provides only one constraint, whereas the description of surface 'shape' (surface gradient or surface normal) requires at least two parameters. Therefore, many shape from shading techniques introduce limitations, such as smoothness of the surface, and use optimization methods to estimate shape. Shape from shading is often not as reliable as other 'Shape from' techniques since it is so easy to confuse with reflections, or for it to fail through poorly modelled reflectance functions.

To overcome some of these problems, shape from photometric stereo was introduced [Clark92, Lee92, Tagare90, Coleman82, Ikeuchi81, Woodham80].

[1] The reader should be aware that many others exist, which we do not mention, especially considering the important spread of such technologies in the last few years.

The main idea behind photometric stereo is to take multiple images of a scene with different light source directions for each image, while keeping viewing point constant. Each image of the scene provides one constraint on the surface shape. Hence, multiple images of the same scene create an overconstraint system, which is solved for the surface shape in order to minimize total cost.

Another technique, which is similar to photometric stereo, is shape from photometric sampling [Ikeuchi90, Sanderson88]. It usually uses many light sources, instead of a few, and a sequence of images corresponding to the light sources. The use of extra light sources eliminates the inaccurate results caused by improper choices of light position in classical photometric stereo. However, it shares so many similarities with photometric stereo that they are commonly considered the same in the literature, and so do we here.

4.2.2 Photometric techniques

The technique of photometric stereo allows us to form surface description from several images of the same surface imaged under various illumination directions. Therefore, it seems ideally suited to our purposes since our problem is itself caused by variations in illuminant direction. Each illumination condition will have its own unique reflectance map, and a given point's intensity will vary accordingly. Therefore, each image defines a unique set of possible orientations. If three or more images are used then their intersection will solve the problem giving only one possible orientation.

In order to exemplify the idea, consider a Lambertian surface illuminated from a given illumination direction, which defines one reflectance map $r(p,q)$. If we are given a facet's intensity under these conditions, we may only conclude that its surface derivatives lie on a particular contour within the p/q plane. This is essentially the fundamental problem of shape from shading; single image techniques need to impose constraints in the spatial domain to resolve this ambiguity. On the other hand, photometric techniques use several images, imaged under different illumination conditions, with their own specific

reflectance map (see Figure 4.2). The facet's derivatives, which are invariant, will lie at the intersection of this set of contours. Since two contours may overlap at more than one point, three images are required to resolve ambiguities in all cases. Mathematically, this may be expressed as follows: consider Lambertian reflectance

$$i = \mathbf{n} \cdot \mathbf{l} \tag{4.1}$$

where i is the grey level intensity, $\mathbf{n}$ is the unit surface normal, and $\mathbf{l}$ is the unit light source direction. Now, consider the same facet illuminated three times with different illuminant directions, and express the surface normal in terms of surface gradient.

$$\begin{pmatrix} i_1 \\ i_2 \\ i_3 \end{pmatrix} = \begin{pmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \\ l_{31} & l_{32} & l_{33} \end{pmatrix} \begin{pmatrix} p \\ q \\ 1 \end{pmatrix} \tag{4.2}$$

Invert Eq.(4.2) in order to isolate the partial derivatives

$$\begin{pmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \\ l_{31} & l_{32} & l_{33} \end{pmatrix}^{-1} \begin{pmatrix} i_1 \\ i_2 \\ i_3 \end{pmatrix} = \begin{pmatrix} p \\ q \\ 1 \end{pmatrix} \tag{4.3}$$

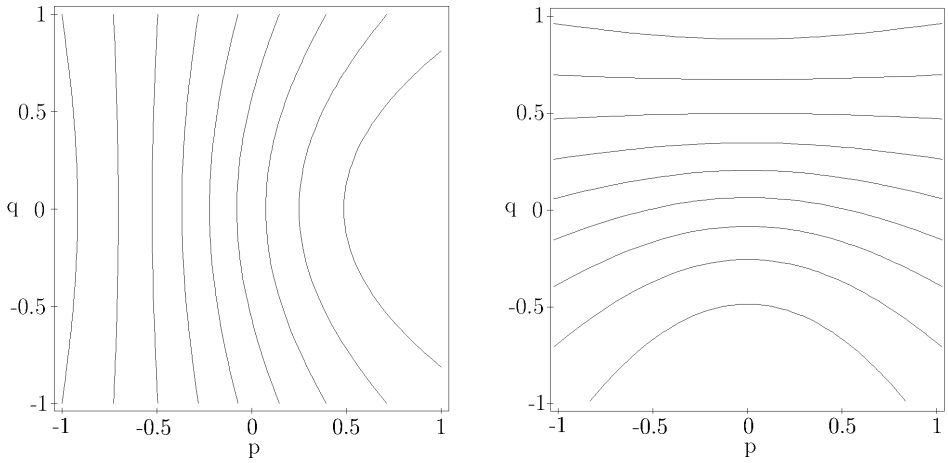


Figure 4.2 Reflectance maps of a matte material at two different illuminant directions; tilt angle rotated 90°.

4.2.3 Related work in photometric stereo

Shape from photometric stereo was first introduced by Woodham in the early eighties [Woodham80]. Woodham proposed that the surface gradient could be solve by using two input images, if the albedo at each surface point is known. In case the albedo were not known, both gradient and reflectance factors could be solved by using an additional image. The method was simple, sensitive to noise, and efficient only for Lambertian surfaces.

Ikeuchi [Ikeuchi81] was the first to obtain the shape of a specular surface employing a photometric stereo technique. In his research, he used a distributed light source obtained by uneven illumination of a diffusely reflecting planar surface and three input images. His solution involved solving a set of non-linear equations; a look-up table, made from the reflectance map, was used to perform the numerical inversion of the three reflectance maps. This method assumed a known object position, and required accurate measurements of reflected brightness.

Tagare and de Figueirdo [Tagare90] estimated the shape of hybrid surfaces. An energy function was minimized with respect to the surface normal and the weights of the Lambertian and specular components. They proved that ten light sources were needed to get a unique solution. This method was based on the assumption that the Lambertian and specular components could be pre-separated.

Coleman and Jain [Coleman82] solved the shape from hybrid surfaces using four light sources. It was based on the assumption that only one of the light sources caused specular reflection; they used relative deviation to determine the specular source.

Solomon and Ikeuchi [Solomon92] extended Coleman's solution by dividing the surface into different areas, depending on the number of light sources illuminating them. The areas illuminated by four sources were solved by Coleman's methods; three sources areas were solved by adding the constraint that the surface normal was a unit vector. Two sources areas could only be

solved if any light source caused specular reflection.

Lee and Kuo [Lee92] introduced the concepts of parallel and cascade photometric stereo. In their paper, they showed that the accuracy of shape from shading algorithms was related to the slope of the reflectance map defined on the gradient space. They suggested two different photometric approaches: first, parallel photometric stereo would take all the images together to produce the best estimation of the surface; second, cascade photometric stereo would take the images one after the other, and the estimated shape from the previous image—computed using triangular element surface approximation—was used as input for the initial estimate of next image. They used a two source photometric stereo, and concluded that the best results could be obtained for orthogonal light sources.

Clark [Clark92] proposed an active photometric stereo approach, which modelled the motion of light source in infinitesimal steps. He was the first to use perspective, instead of orthographic projection, thus removing the necessity to assume the light coming from infinity. The computation was local, non-iterative, and directly solved depth in a closed form equation. In order to measure the infinitesimal image gradient with respect to change in lighting, seven images were required to provide an acceptable discrete approximation.

It is a common practice to avoid interreflections, i.e. the mutual illumination between surface facets, and indeed none of the above methods dealt with them. Nayar et al [Nayar90] challenged the interreflection problem applying photometric stereo to Lambertian surfaces. They observed that erroneous shape in the presence of interreflections was a little bit shallower than the real shape, and therefore it could be iteratively refined.

4.3 Rendering

Traditionally, rendering is the processing of a scene's image using colour and shading to make it appear solid and three-dimensional. However, it must be clarified that the rendering process is here understand in a simpler, different manner. Precisely, here rendering only consists of a relighting algorithm whose input is the surface model and the desired illumination direction and whose output is the simulated image (see Figure 4.1). With that respect, it can be said that our rendering method is model-based, in contrast to image-based rendering methods. The basic difference between them is found in the approach used to obtained the virtually generated image. In traditional model-based rendering, a geometric model of a scene, together with surface reflectance properties and lighting parameters, are used to generate an image of the scene from a desired light point. In image-based rendering a set of images are taken from (possibly) known viewpoints and they are directly processed to create new images.

Chapter 5

Approach to the problem

In the preceding chapters some theoretical background concerning the main topics of this research was presented. Firstly, Chapter 2 examined possible descriptions of physical surfaces, specially rough ones, and the interaction with light in the imaging process. Secondly, Chapter 3 described the classifier as a combination of three mechanisms: texture measure, post-processing, and discrimination. Finally, Chapter 4 surveyed popular surface recovery techniques and identified the relighting process. This investigation through literature should be perceived as a supportive tool when approaching the current issue. In this chapter we state the problematic, use both theory and previous work to recommend a model-based solution, and describe a relatively simple and speed-efficient implementation.

5.1 Problematic

It has been shown that changes in the angle of illuminant incident upon a surface texture can significantly modify its appearance. These alterations can affect the output of texture features to such an extent that they may cause complete misclassification. Essentially, this occurs because a side-lighting enhances the appearance of the surface texture producing an image which is a directionally filtered version of the surface height function. Furthermore, as Kube and Pentland [Kube88] predicted in their pioneering work, the axis of this filter is a function of the illumination's tilt angle. In the context of texture

classification, this is very unfortunate as many classifiers employ directional filters in their feature measures —like e.g. Gabor filters, which are used in this research.

Notwithstanding the evidence, very little work has been published on this subject. [Chantler94] studied the effect caused by variations in illuminant direction on images of physical texture, investigated the impact of this effect on texture classification and proposed methods of reducing classification errors under this situation. Leung and Malik [Leung99,01] developed a texture classification scheme that identifies 3D textons[1] for the purposes of illumination and view-point invariant classification. They tackled the difficulties of recognising that samples of the same material under different viewpoint/lighting settings actually belong to the same type. They made the point that variations due to surface relief cannot be treated with simple brightness normalisation or intensity transforms. [McGunnigle98] confirmed by simulation and experimentation that the features on which classification is based are functions, not only of the surface, but also of the illuminant tilt angle. He evaluated several proposals in the literature for reducing the tilt induced effect. However, he encountered them inefficient for his data set and proposed a new method based on photometric stereo. His evaluation showed that it was able to accurately model real textures and to significantly reduce the effect of tilt variation on classification.

5.2 Model-based solution

The ability to set classification parameters from a set of examples is very important. Clearly, the quality of the classifier setting depends on the quality and size of the training set, which is always finite. Therefore, to design and set a classifier, it is not possible to use all the textures which will later need

[1] Fundamentally, a 3D texton is a vocabulary of prototype tiny surface patches with associated local geometric and photometric properties.

classifying. The classifier setting methods must be inductive in the sense that the information obtained from the elements of the training set must be generalised to cover the whole feature space, implying that the classifier setting should be near optimal for all feasible textures, even for those which it had never 'seen' before.

When tackling classification under varying lighting condition, the problem is considerably increased. Not only should the classifier be able to discern among different textures, but it also should be able to perform robustly against changes in the appearance of identical textures. In that case, it is obviously not viable to train at every single illuminant direction and store up infinite number of training statistics to secure a more successful classification.

One potential solution to that difficulty is a model-based method. It takes advantage on the fact that it is not based on the actual surface but on its model. Therefore, it is a much more reasonable approach because it merely requires a finite and rather small data set to model the surface. Essentially, this technique aims to anticipate the feature space distributions by modelling the physical and analytical processes of imaging and feature extraction. It forms a spatial model of the training surfaces in a primary training stage. The classification process proper begins with secondary training, when the recovered surfaces are synthetically rendered under the specific lighting condition, and the resulting images are the basis of training. We hypothesise that if the model components, i.e. the reflectance function and the surface characterisation, are satisfactorily accurate, the classification rates will come close to those of the *best case*[2] classification.

Figure 5.1 illustrates the general set-up of the system identifying three main steps: recovery or primary training, secondary training and classification. In following subsections, each phase is individually explained highlighting the functionality of relevant components.

---

[2] *Best case* is referred to that situation where training and classification are performed under the same lighting conditions.

5.2.1 Recovery stage

Firstly, the procedure need a structured set of data held in a repository—e.g. a database for a computer-based application. This database should be accessible from the software in order to restore the data. Generally, the data set consists of several images obtained at different illuminant angles. The number of required images relies entirely on the technique used for the implementation. Specifically, we adopted a photometric technique due to [McGunnigle98] which requires just three images as data set. This technique is explained later on in Section 5.3. In any case, the resulting surface model is composed of three images, $p(x,y)$, $q(x,y)$, and $a(x,y)$, representing the partial derivatives and albedo respectively. This modelling process is without doubt the key component of the system; the final misclassification rate will mostly depend on the precision to imitate the real appearance of the materials.

5.2.2 Training stage

The surface model found in the previous stage is fed into the relighting algorithm so as to simulate the appearance of the actual image texture under a definite lighting condition. Basically, the algorithm applies Eq.(5.1) to the surface model in order to produce an estimation of the radiant intensity. Consequently, this procedure presumes that the reflectance function of the real surface can be modelled as Lambertian and the lighting conditions are known[3]. In the next step, a multichannel system measures the prior simulated image with the aim of extracting key features. This set of texture features is subsequently entered into the discriminant generator which separates the feature space so that only patterns from one class are in each separated region. The result is a set of statistics which will become the basis for posterior classification.

---

[3] This assumption confines the proposal to applications where the direction of illuminant during the classification stage can be either manually or automatically controlled.

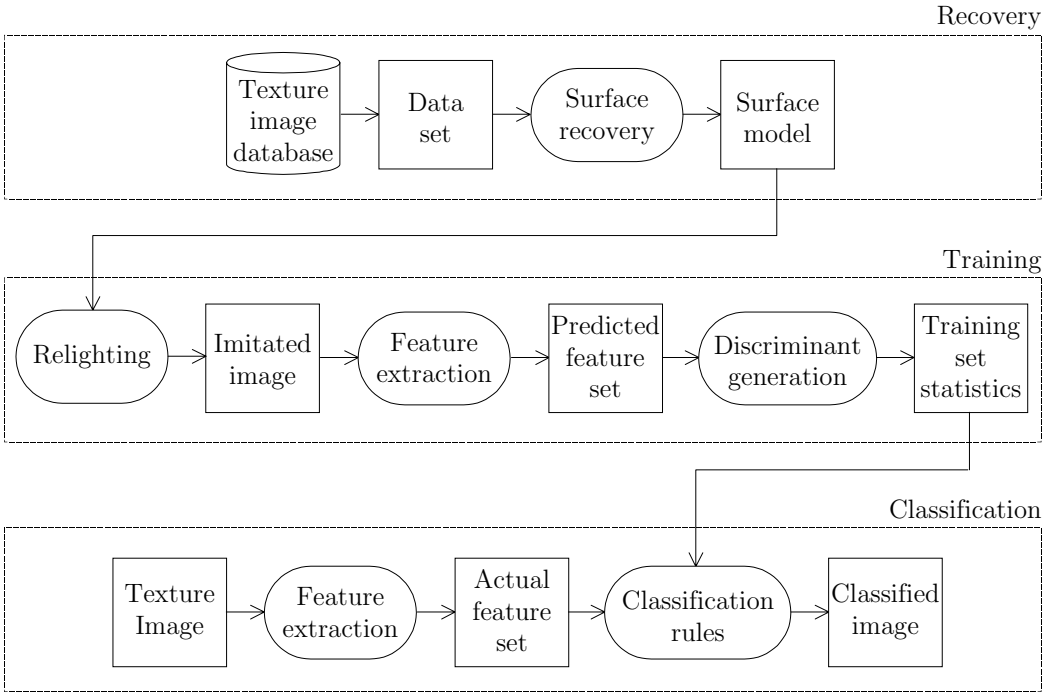


Figure 5.1 Block diagram of suggested model-based classifier.

5.2.3 Classification stage

Finally, the actual texture is submitted to the very same multichannel scheme used during secondary training. The feature images extracted are passed to a statistical classifier that uses the statistics obtained in the training session, and labels each pixel as belonging to a certain class on the basis of its feature vector and the a priori probability of that vector being a member of each class. The resulting output image is a class map in which the value of each pixel correspond to the group with the lowest discriminant score at that pixel.

5.3 Photometric implementation

In Section 4.2 the related literature was reviewed in order to identify appropriate implementations as well as the practical issues arising from the technique's use. From this study, we understand that accurate models imply difficult and less efficient systems. As noted in [McGunnigle98], in the context

of a model-based classifier the accuracy of surface recovery is of secondary importance to the accurate prediction of training images. This relaxation allows other factors to be taken into account. We also believe that the ease of implementation and the computational speed become a priority for our purposes.

5.3.1 A simple photometric stereo scheme

Assuming Lambertian surface reflection, and no significant shadowing or inter-reflection, the intensity of an illuminated surface may be expressed as:

$$i(x,y) = \gamma\rho \frac{-p(x,y)\cos\tau\sin\sigma - q(x,y)\sin\tau\sin\sigma + \cos\sigma}{(p^2(x,y) + q^2(x,y) + 1)^{1/2}} \tag{5.1}$$

where γ denotes the incident light flux, and ρ denotes the variations in reflectance property such as surface markings.

As noted in Section 4.2.2, three images under different illuminant conditions are required to avoid possible ambiguities. If the surface is illuminated from $\tau = 0^\circ$, 90°, and 180°, and same slant angle, Eq.(5.1) can be significantly simplified. For each given point (x,y) of the image the radiant intensity can be stated respectively as:

$$i_{_0} = \frac{\gamma\rho}{\left(p^2 + q^2 + 1\right)^{1/2}}(-p\sin\sigma + \cos\sigma) \tag{5.2}$$

$$i_{_{90}} = \frac{\gamma\rho}{\left(p^2 + q^2 + 1\right)^{1/2}}(-q\sin\sigma + \cos\sigma) \tag{5.3}$$

$$i_{_{180}} = \frac{\gamma\rho}{\left(p^2 + q^2 + 1\right)^{1/2}}(p\sin\sigma + \cos\sigma) \tag{5.4}$$

Adding Eq.(5.2) and Eq.(5.4) will produce a non-linear function of the surface derivates:

$$i_{_{NL}} = i_{_0} + i_{_{180}} = 2\frac{\gamma\rho}{\left(p^2 + q^2 + 1\right)^{1/2}}\cos\sigma \tag{5.5}$$

Now dividing Eq.(5.2) and Eq.(5.3) by Eq.(5.5) will generate two linear functions mapping surface slope to image intensity:

$$i_p = \frac{i_0}{i_0 + i_{180}} = \frac{1 - p\tan\sigma}{2} \tag{5.6}$$

$$i_q = \frac{i_{90}}{i_0 + i_{180}} = \frac{1 - q\tan\sigma}{2} \tag{5.7}$$

These equations may be transposed to give the desired derivative fields:

$$p = \frac{1 - 2i_p}{\tan\sigma} = \frac{1 - 2\dfrac{i_0}{i_0 + i_{180}}}{\tan\sigma} \tag{5.8}$$

$$q = \frac{1 - 2i_q}{\tan\sigma} = \frac{1 - 2\dfrac{i_{90}}{i_0 + i_{180}}}{\tan\sigma} \tag{5.9}$$

Knowing the gradient fields one can use any equation ranging between Eq.(5.2)-(5.5) to isolate the albedo of the surface texture, yet it is advantageous to choose Eq.(5.5) since it produces less computational cost. In that case, the albedo is given by

$$a \triangleq \gamma\rho = \frac{i_0 + i_{180}}{2\cos\sigma}\left(p^2 + q^2 + 1\right)^{1/2} \tag{5.10}$$

Due to all the assumptions, i.e. Lambertian reflection, no shadowing, and neglectful interreflections, this model is sub-optimal and certainly inferior, in terms of accuracy, to some techniques surveyed in Chapter 4. Even if the reflectance function is near Lambertian, surfaces with high slope angles may perform comparatively worse than those of low slope angles, since the former may present major levels of shadows and interreflections. Despite inadequacies, this scheme does provide a fast and simple implementation and will be adopted during experimentation. The performance of this scheme, hereafter refer to as *simple photometric stereo*, will be assessed carefully in Chapter 6.

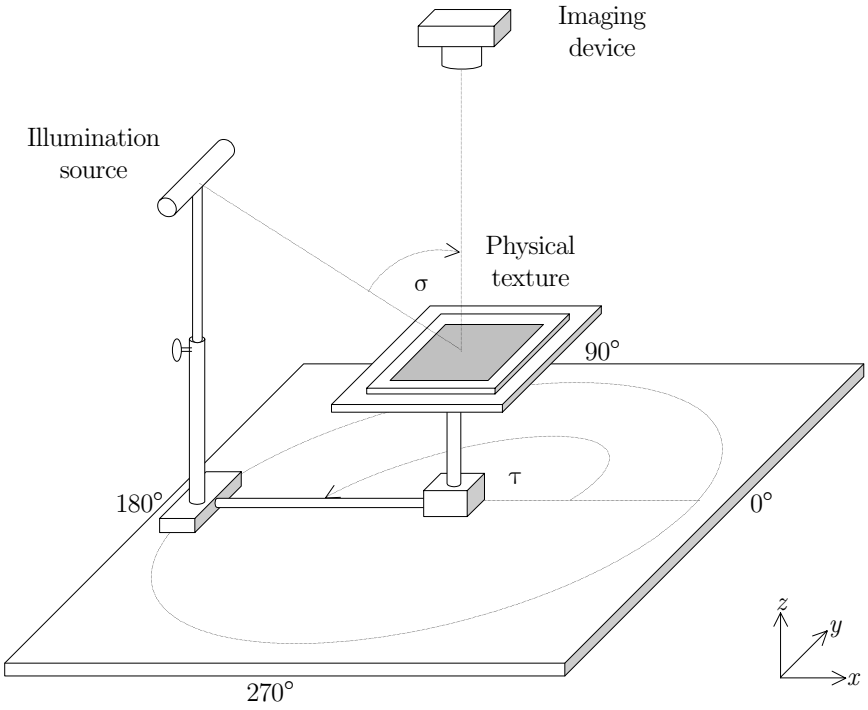


Figure 5.2 Laboratory apparatus to collect photometric stereo data.

5.3.2 Collecting photometric data

The required input data is a set of images of test surfaces, each one lit from a different known direction, all captured from the same view point. In the particular case of the simple photometric stereo scheme (see Section 5.3.1), only three images are acquired at tilt angles of 90° increments starting at 0°. Implicit in this approach is the assumption that the illumination's slant angle is held constant for all three images — see Eq.(5.2)–(5.4). As done generally in the field of photometric stereo, we collect multiple images of a static surface texture with a static camera under varying lighting conditions. Figure 5.2 shows the device constructed to assist with this process. The sample is placed on a platform which is situated perpendicular to the camera's line of sight. A single light source is mounted on an arm which can be positioned to the specific zenith and azimuth. In the context of lighting, azimuth is commonly referred to

as the illuminant tilt angle; analogously, zenith is referred to as the illuminant slant angle. The light source is a single filament tube with a mask. The mask gives the source a near Gaussian which helps to smear specular highlights and reduces aliasing of the azimuth function. The imaging device is an CCD camera which is fixed in the apex of the apparatus and is connected to a frame store mounted in a workstation.

5.4 Feature extraction

Section 3.3 reviewed different approaches to texture feature extraction and justified the use of signal processing techniques. Specifically, motivated by studies of human perception, many computer vision researchers have proposed a paradigm based on a filter-bank model. Among several categories of operators which can perform a joint space/spatial-frequency decomposition, the Gabor functions have been shown to be optimally localized per the uncertainty principle in both the space and spatial-frequency domains.

This section presents a bank of Gabor filters for the extraction of texture features from multi-textured images. The approach includes an integrated design for the filtering process as well as a post-processing scheme to adapt the filtered images for classification.

5.4.1 Multichannel scheme

The multichannel filtering approach is intuitively appealing because it allows us to analyse differences in sizes and orientations of different textures. Unlike other approaches to texture analysis, the multichannel filtering approach is inherently multiresolutional. Another important advantage is that one can decompose the original image into several filtered images with limited spectral information.

Figure 5.3 illustrates a general k-channel filtering scheme, where M is the number of texture features. The input image $i(x, y)$ is assumed to be composed of D disjoint textures $t_1, t_2, \ldots, t_D$ with $D \geq 2$. Each single filter channel in the

scheme consists of a bandpass Gabor filter $g_k(x,y)$, a non-linear operator, and a smoothing post-filter $h_k(x,y)$. The subscript k denotes a particular filter channel with $1 \leq k \leq M$.

The output of a first filtering stage $o_k(x,y)$ is the convolution of the input image with the 2D Gabor filter

$$o_k(x,y) = i(x,y) * * g_k(x,y) \tag{5.11}$$

where $**$ denotes convolution in two dimensions. The non-linear operator $\aleph$ is computed in the following stage as

$$m_k(x,y) = \aleph\Big\{o_k(x,y)\Big\} = \aleph\Big\{i(x,y) * * g_k(x,y)\Big\} \tag{5.12}$$

where $m_k(x,y)$ is an energy estimation in the local region. However, $m_k(x,y)$ has been shown to fail in accomplishing the discriminant criteria (see Section 5.4.3 for discussion). Thus, a smoothing post-filter is then applied to $m_k(x,y)$ yielding the final feature image

$$f_k(x,y) = m_k(x,y) * * h_k(x,y) \tag{5.13}$$

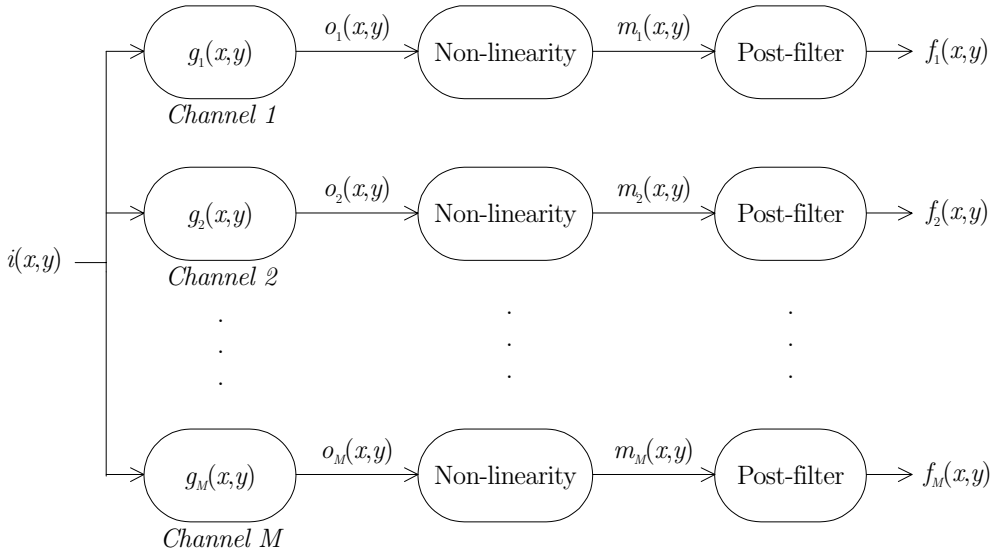


Figure 5.3 Multichannel filtering scheme for feature extraction.

5.4.2 Filtering in the frequency domain

This section focuses on the filtering stage of a particular channel. We believe it is desirable to perform the filtering in the frequency domain because of the computational difficulty of a two-dimensional convolution in the spatial domain. Then, the complex filtering expressed by Eq.(5.11) is transformed into a product in the frequency domain

$$O_k(u,v) = I(u,v) \cdot G_k(u,v) \tag{5.14}$$

where $G_k(u,v)$ is the Fourier transform of the Gabor filter in the k-th channel and $I(u,v)$ is the Fourier representation of the image. Figure 5.4 depicts a detailed implementation of a Gabor filter in the frequency domain.

Some theory is now introduced that justifies the implementation adopted. First, we need to rewrite some formulae in a proper manner. The GEF defined by Eq.(3.3) can be seen as a sum of two functions: one is real and even-symmetrical, and the other one is imaginary and odd-symmetrical. This is summarised by Eq.(5.15)

$$g(x,y) = \underbrace{e(x,y)}_{\text{even, real}} \cdot \left\{ \underbrace{\cos[2\pi(u_0 x + v_0 y)]}_{\text{(1) even, real}} + \underbrace{j \sin[2\pi(u_0 x + v_0 y)]}_{\text{(2) odd, imaginary}} \right\} \tag{5.15}$$

where, for simplicity, $e(x,y)$ represents the Gaussian envelope. Thereby, when transformed into the frequency domain, see Eq.(5.16), we can distinguish even-symmetrical part from odd-symmetrical part.

$$\begin{aligned} G(u,v) = \underbrace{E(u,v)}_{\text{even,real}} ** \Bigg[& \underbrace{\frac{1}{2}\delta(u-u_0, v-v_0) + \frac{1}{2}\delta(u+u_0, v+v_0)}_{\text{(1) even, real part}} + \\ & + \underbrace{j\frac{1}{2j}\delta(u-u_0, v-v_0) - j\frac{1}{2j}\delta(u+u_0, v+v_0)}_{\text{(2) odd, real part}} \Bigg] \end{aligned} \tag{5.16}$$

Note that Eq.(5.16) is, in effect, equivalent to Eq.(3.4). Let us denote the convolution between $E(u,v)$ and component (1) as $G_{even}(u,v)$, and the

convolution between $E(u,v)$ and component (2) as $G_{odd}(u,v)$:

$$\begin{cases} G_{even}(u,v) = \frac{1}{2}\exp\left[-\frac{(u-u_0)^2}{2\sigma_u^2} - \frac{(v-v_0)^2}{2\sigma_v^2}\right] + \frac{1}{2}\exp\left[-\frac{(u+u_0)^2}{2\sigma_u^2} - \frac{(v+v_0)^2}{2\sigma_v^2}\right] \\ G_{odd}(u,v) = \frac{1}{2}\exp\left[-\frac{(u-u_0)^2}{2\sigma_u^2} - \frac{(v-v_0)^2}{2\sigma_v^2}\right] - \frac{1}{2}\exp\left[-\frac{(u+u_0)^2}{2\sigma_u^2} - \frac{(v+v_0)^2}{2\sigma_v^2}\right] \end{cases} \quad (5.17)$$

An image function $i(x,y)$ is always real valued, thus according to the symmetry property its Fourier transform satisfies

$$I(u,v) = I^{\#}(-u,-v) \quad (5.18)$$

where $\#$ denotes complex conjugate. Eq.(5.18) implies that $I(u,v)$ is a complex function consisting of even-symmetric real part $I_R(u,v)$ and odd-symmetric imaginary part $I_I(u,v)$.

To calculate the complex output of the filter in the spatial domain, the result of correctly computing the above four spectrums is applied to the inverse Fourier transform. Among all the possible combinations (see Table 5.1), it can be demonstrated that only a complex spectrum consisting of even real part and odd imaginary part—products (a) and (c)—gives a real image $o_R(x,y)$ in the spatial domain. Conversely, a complex spectrum consisting of odd real part and even imaginary part—products (b) and (d)—provides an imaginary image $o_I(x,y)$ in the spatial domain.

Image spectrum		Gabor spectrum		
Part	Properties	Part	Properties	Product
$I_R(u,v)$	real, even	$G_{even}(u,v)$	real, even	(a) real, even
$I_R(u,v)$	real, even	$G_{odd}(u,v)$	real, odd	(b) real, odd
$I_I(u,v)$	imaginary, odd	$G_{even}(u,v)$	real, even	(c) imaginary, odd
$I_I(u,v)$	imaginary, odd	$G_{odd}(u,v)$	real, odd	(d) imaginary, even

Table 5.1 Summary of combinations between image and Gabor filter.

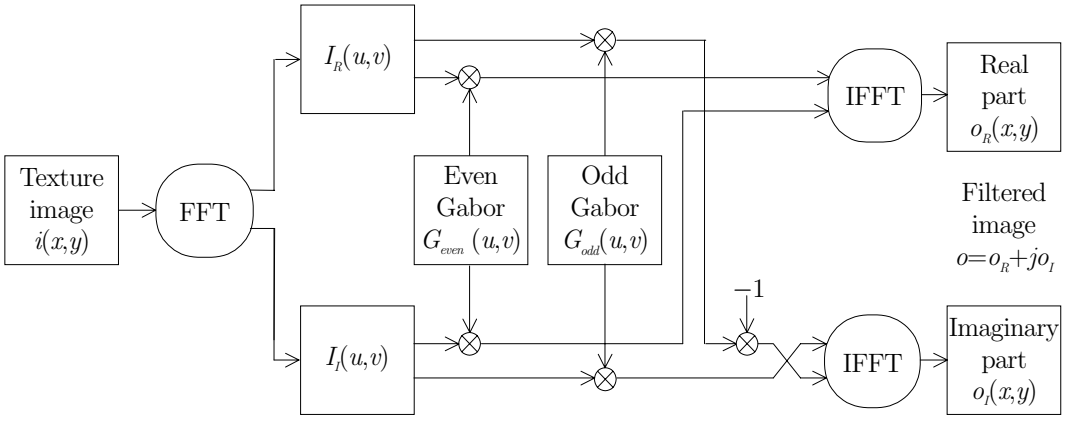


Figure 5.4 Block diagram of Gabor filtering in the frequency domain.

It is worthy of note that the IFFT software was originally devised to render only the real part. Nevertheless, this program can be used to obtain the imaginary part by multiplying the complex input image by $-j$. It is equivalent to swap the input images and multiply the real part by -1.

5.4.3 Post-processing

Although the form of the Gabor filter itself is relatively standard, it is generally advocated a subsequent processing of the signal prior to classification. This post-processing stage aims to estimate the energy in the filter output in the local region. In the majority of approaches, the local energy function takes the form of a non-linearity followed by a smoothing filter (Figure 3.3).

Commonly applied non-linearities are the magnitude $|\cdot|$, the squaring $(\cdot)^2$, and the rectified sigmoid $|\tan(\alpha\cdot)|$ [Randen99]. We avoid using the rectified sigmoid because it requires tuning of the saturation parameter, α. Between the other two parameter free non-linearities, we chose signal magnitude since it is in a range which is more comparable to that of the original image. This means that numerical stability is less of a problem than for signal power, and outlying samples are less significant. Typical approaches to the calculation of the signal magnitude are the quadrature filtering and the absolute value estimator. We adopt the quadrature filtering scheme for the following reasons:

1. For pixel-based classification, which may be adversely affected by smeared transitions between areas of different textures caused by low pass filtering, quadrature filters are more appropriate [Aach95].

2. This scheme makes explicit the suppression of phase in favour of magnitude information as the basis for classification.

In that case, the final expression for Eq.(5.12) is given by

$$m_k(x,y) = \left| i(x,y) ** g_k(x,y) \right| \tag{5.19}$$

Another consideration is that the post-processing must perform reliably in conjunction with the discriminant function. Since the classifier used in the discriminant stage is optimal for Gaussian data only (see Section 3.4.2 for details) it is also necessary that the output feature image, $f_k(x,y)$, should have at least an approximately Gaussian distribution. It has been found that even when the texture is filtered in quadrature, the feature distributions have unacceptably large variances, presumably because of filtered images not being sufficiently narrowband [Kieran95]. In addition, the quadrature filter output histogram approximates a Rayleigh distribution. This result is supported by theory: "the PDF of the resultant of two uncorrelated Gaussian processes will follow a Rayleigh distribution" [Couch93]. After low pass filtering, the distribution more closely approximates the Gaussian case, reaching the optimality of the discriminant. Commonly applied smoothing filters in the local energy function are rectangular and Gaussian low pass filters. Experience has taught us that the Gaussian filter is the far better choice and will consequently be used in our implementation [Randen99]. The lowpass Gaussian filter in the k-th channel is given by

$$h_k(x,y) = \frac{1}{2\pi\sigma_{Lk}^2} \exp\left[-\frac{(x^2+y^2)}{2\sigma_{Lk}^2} \right] \tag{5.20}$$

where σ_{Lk} determines the Gaussian filter in the channel k.

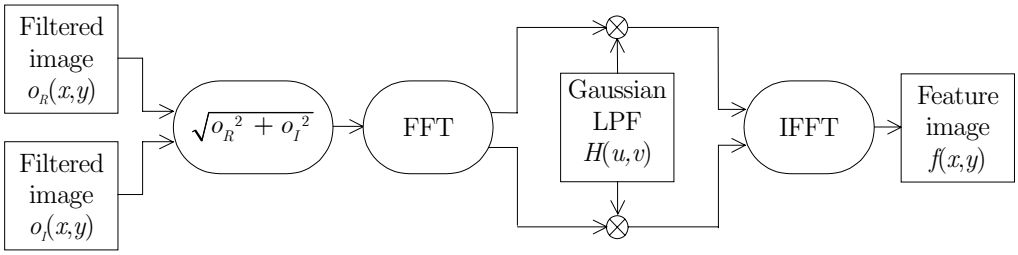


Figure 5.5 Post-processing scheme based on quadrature filtering.

The final implementation of the post-processing procedure is shown in Figure 5.5. Note that the Gaussian smoothing filter is also implemented in the frequency domain in order to avoid the costly convolution.

5.5 Gabor filters design

Many studies such as [Dunn95] and [Weldon96] have been devoted to the algorithmic design of optimal Gabor filters for a particular task. They have considered a supervised texture segmentation problem, i.e. samples of textures are provided to help in designing the filters. While former designs are more efficient, we prefer to propose a more standard filter implementation based on [Jain91] but with some modifications. This implementation, although suboptimal, will allow us to obtain more general conclusion and extend our results to a wider range of textures. Once again, we remark that we are interested in the coherence of the system as a whole in preference to separately optimise each component.

5.5.1 Filter characterisation

Our filtering technique uses a multichannel scheme based on a bank of Gabor filters. We now present a consistent way of specifying a Gabor filter in the polar and radial space. Existing techniques in the literature use either complex-valued Gabor filters [Bovik90], pairs of Gabor filters with quadrature phase relationships [Perry89, Tan90], or real-valued even-symmetric filters [Malik90]. Although Malik and Perona provide some explanation for using even-symmetric

filters only, our implementation in the frequency domain need the use of the complex form of Gabor filters—the optimal space/frequency characteristic of the Gabor filter only holds for the complex form of the function.

Clearly, to uniquely describe a Gabor filter in the spatial domain, four independent filter parameters σ_x, σ_y, u_0, and v_0 have to be specified. However, the frequency- and orientation-selective properties of a Gabor filter are more explicit in its frequency domain representation. Additionally, our filtering implementation is in the frequency domain. Therefore, we prefer to specify a Gabor filter by its spatial frequency location and bandwidth. The filter location is determined by the radial centre frequency ω_0 and orientation θ_0 given by

$$\omega_0 = (u_0^2 + v_0^2)^{1/2} \tag{5.21}$$

$$\theta_0 = \tan^{-1}\left(\frac{v_0}{u_0}\right) \tag{5.22}$$

where ω_0 is in cycles/image and θ_0 is in degrees measured from the u-axis. In addition to radial frequency and orientation, the frequency bandwidth B_ω and orientation bandwidth B_θ are also of interest. For the Gabor filters (either even- or odd-symmetrical part) defined by Eq.(5.17), the half-peak magnitude bandwidth are given by

$$B_\omega = \log_2\left(\frac{\omega_0 + (2\ln 2)^{1/2}\sigma_u}{\omega_0 - (2\ln 2)^{1/2}\sigma_u}\right) \tag{5.23}$$

$$B_\theta = 2\tan^{-1}\left(\frac{(2\ln 2)^{1/2}\sigma_v}{\omega_0}\right) \tag{5.24}$$

where B_ω is in octaves and B_θ is in degrees. Figure 5.6 shows an intuitive way to specify Gabor filters in terms of their spatial frequency location and bandwidths: ω_0, θ_0, B_ω, and B_θ.

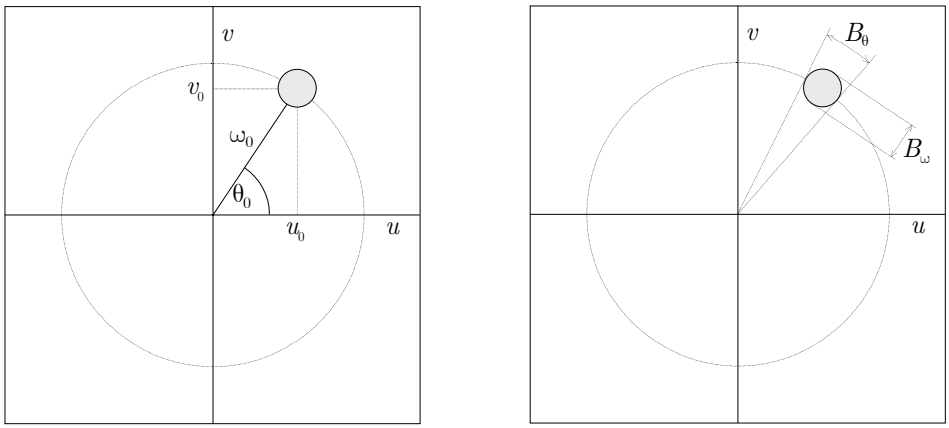


Figure 5.6 Gabor filter parameters in spatial frequency domain:
(a) Filter location, **(b)** Filter bandwidth.

To determine the optimal values of σ_x and σ_y (or analogically B_ω and B_θ), many algorithms need to know the textures in advance. Hence, the results obtained under these circumstances are merely valid for a few number of textures. As we aim to generalise our conclusion for an ampler range of textures, we assume a popular constraint introduced by Jain and Farrokhnia based on biological justifications which has been adopted by most non-design researchers. This restriction fixes all the filters with frequency and angular bandwidth of $B_\omega = 1$ octave and $B_\theta = 45$ degrees respectively. Several experiments have shown that the frequency bandwidth of simple cells in the visual cortex is about 1 octave. Taking these values, one simply must designate the filter location (ω_0, θ_0) and the space constants of the Gaussian envelope will be automatically calculated by using Eq.(5.23) and Eq.(5.24).

5.5.2 Selection of filters

As advocated by [Jain91], we opt for a systematic sampling of the image spectrum. Theoretically, the ideal set of filters used in our algorithm should result a nearly uniform coverage of the spatial frequency domain. Jain proposed sampling the image spectrum at intervals of 1 octave and 45 degrees. For instance, for a 512×512 image a total of 32 filters can be used—4 orientations

and 8 radial frequencies. We note here that a large number of filters can lead to associated disadvantages in computing a large number of filtered images and classifying a large-dimension feature space. On the other hand, experimental evidence suggests that classification error tends to increase as the number of filters decrease, particularly near texture boundaries. Attempting to solve this trade-off, we indicate the following points:

- For some textures, filters with very low radial frequency may not be very useful, since these filters capture spatial variations that are too large to explain textural variations in an image [Jain91].

- The radial spectra of different surface models (and consequently of their images) differ most markedly at low frequencies [McGunnigle98].

- The S/N ratio decreases with increasing frequency because of attenuation of high frequencies due to blurring.

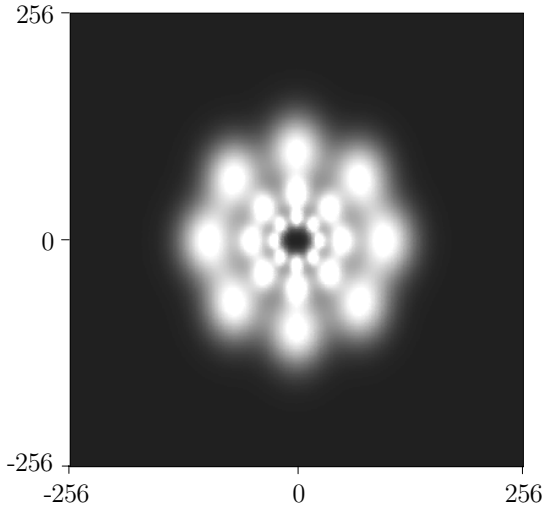


Figure 5.7 Plot of the filter set in the spatial-frequency domain (512×512). Only the even-symmetrical part of the frequency response is shown. The origin is at the centre of the image and the axes are in cycles/image-width.

This would suggest that it is the low frequencies that should be most closely scrutinised by the classifier. As an empirical compromise, we decided to use three sets of filters located at radial frequencies of 25, 50, and 100 cycles/image. As done by Jain, each frequency band will be sampled at four equidistant orientations, $\theta_0 = 0^\circ, 45^\circ, 90^\circ,$ and 135°. This leads to a total of 12 texture features instead of a 32-dimensional feature space, which represents a reduction of 62.5%. Figure 5.7 shows the frequency response of the bank of Gabor filters which ranges from the low frequencies to the mid band.

Chapter 6

Assessment of image prediction

Chapter 5 proposed a model-based scheme to overcome the problem of classifier failure induced by varying lighting direction. In an initial stage, the technique estimated the surface derivative field and albedo using photometric stereo in order to predict the observed texture under specified illumination conditions. This chapter is dedicated to the evaluation and verification of the technique's robustness in relation to different effects, specifically the influence of surface roughness and varying illumination angles. The primary target is to ascertain in what way, and to what extent, departures from the ideal Lambertian model and assumptions concerning shadowing and interreflections, affect image prediction.

6.1 Introduction

Previous chapter suggested a complete photometric classification system to reduce the effect of variations in lighting conditions on classification. Now, this chapter concentrates in the assessment of the most important component of the system: the surface recovery. Of particular interest is the degree to which the simulated image conforms to the correct value. With this goal, following sections present experimental studies that may help to discern in which situations the technique does perform reliably.

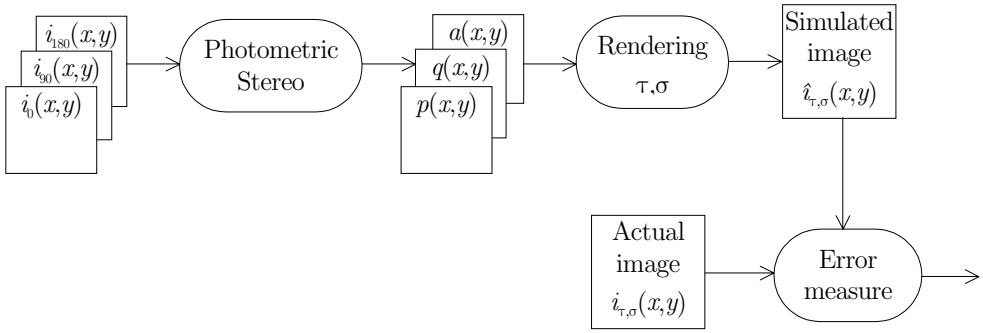


Figure 6.1 Experimental measure of image prediction.

Figure 6.1 outlines the experimental approach to measure the accuracy of image prediction in a controlled and analytical manner. A particular case of study can be simulated by changing some variables such as input image or lighting conditions, and then one can evaluate how these changes affect the performance of the algorithm. Specifically, we are interested in the technique's robustness under the influence of the following effects:

- *Surface roughness*: intuitively the more abrupt the surface is, the higher the slopes become. As slopes increase, more and more facets are affected by shadows. Since the recovery method does not consider shadowing, surface roughness will play an important role in the image prediction.

- *Varying illuminant tilt angle*: the algorithm interpolates between training images in order to imitate the intensity image of a surface captured under arbitrary tilt. Then, the inaccuracy of the image prediction will fluctuate throughout the tilt range, presumably taking minimum values at $\tau = 0^\circ$, 90°, and 180° (photometric data set) and maximum values in between.

- *Varying illuminant slant angle*: although a model-based scheme should be equally skilled at dealing with changes in slant, this represents a much more demanding task since now it is required the extrapolation among the training images. Besides, shadowing becomes more intense for higher values of slant, commonly for $\sigma \geq 45^\circ$.

In order to allow comparison between different cases, a single generic criterion of accuracy will be selected. Objective quantitative methods usually estimate the quality of an image by comparison with a known reference image. Popular methods use simple measures such as the maximal absolute error or correlation between images. With these methods it is possible to distinguish a few big differences from a lot small differences. However, we are more interested in calculating the average error rather than recognising particular errors. One suitable method is the *mean square error* (MSE); for a $N \times N$ image its definition is given by

$$MSE = \frac{1}{N^2} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} \left[i(m,n) - \hat{i}(m,n) \right]^2 \tag{6.1}$$

where $\hat{i}(x,y)$ denotes the estimated image and $i(x,y)$ is the reference image.

6.2 Limitations of the model

Before analysing the performance of the simple photometric stereo scheme in detail, we make some observations regarding the assumption that the rendering can be described by Eq.(5.1). The relighting process is a priori restricted by the following three limitations:

1. Although, in general, the reflectance function is approximately diffuse, we note that it is not perfectly Lambertian; in fact, we will intentionally challenge the method for non-Lambertian textures.

2. Choosing to describe the reflectance function using Eq.(5.1) implies that for certain facet orientations a negative intensity may be observed; this is clearly not the case, and

3. Eq.(5.1) makes no allowance for the non-local effect of cast shadows.

The degree to which the second and third points occur is highly dependent on the surface relief, whereas the first point relies mostly on the surface material.

6.2.1 Non-Lambertian reflectance

The reflection on an object surface is often partitioned into two components: *interface* reflection and *body reflection*, according to the dichromatic model proposed by [Shafer85]. Interface reflection, also known as specular, originates from light reflection on an optical boundary between the surface and the air; for smooth surfaces, it is mirror-like. Body reflection results from the light that penetrates the object body and is scattered by the object material particles back into the air. It is fairly diffuse, even for smooth surfaces, and actually it is the mechanism thanks to which objects can be seen from any direction.

Our approach only considered body reflection and used a Lambertian model to characterise it. Certainly, the Lambertian model is only a simplification of reflectance behaviour of common materials, but it is very attractive for its linearity and it is straightforward to implement. However, as the assumption of Lambertian reflectance breaks down, the method produces error. Sometimes the Lambertian assumption is only violated in a limited number of pixels, e.g. when some pixels accommodate the specular reflection. Specular reflections are frequent sources of errors for many computer vision algorithms.

6.2.2 Cast and self shadowing

We pointed before that in some cases the Lambertian function can return negative values. As a result, self-shadowed regions are assumed to have negative intensities since the technique does not clip[1] affected areas. The model-based technique also fails to account for cast shadows, thus the more abrupt a surface is the more difficult the prediction turns out to be. In this respect, it can be observed that for the higher values of slant, where shadowing is severe, the technique may also undergo more difficulties. Figure 6.13 is a good case in point that reflects the difficulty in predicting cast shadows, as observed in the lack of shadow behind the bump in the upper left-hand side of the surface.

[1] Process an image so as to remove the parts above or below predetermined thresholds.

6.3 Test textures

A total of thirty-five physical texture samples were used throughout the experimental work presented in this dissertation. These textures are used not only in this chapter but also in Chapter 7 to compose 'artificial' multitextured images for classification tests. 512×512 monochrome images (when presented to the method) were obtained from each sample using illumination tilt angles ranging between 0° and 360° incremented by either 10° or 30° degree steps. All textures were illuminated at a slant angle of 45°. In addition, twenty-two surfaces were also illuminated at slant angles of 30°, 60°, and 75°. Appendix A contains one example image of every texture (imaged at slant angle of 45° and tilt angle of 0°).

The test set includes a wide variety of textures which may be grouped into the following three classes according to their visual appearance:

- isotropic surfaces, e.g. plaster1 or deposit3, and in general all the textures made from plaster showed a highly uniform topology.
- directional surfaces, e.g. anaglypta3 or card, in which the directionality is approximately aligned with the y-axis; and
- repeating primitives, e.g. peanuts or peas, which show a great deal of shadowing and therefore they are a handicap to the photometric stereo scheme.

A histogram description was applied to the variation of surface gradient in order to characterise the nature of the textures. Different types of gradient distributions were identified, which corroborates the diversity of the dataset. For instance, let us consider deposit3, card, and peas, which provides a representative example of each texture class defined above. It was observed that surfaces which are formed by natural process, e.g. fracture, deposition, or wave action, are essentially random textures and most approximate Gaussian random

process—like deposit3, which was formed by depositing plaster powder on a flat plaster surface (see histogram for p and q in Figure 6.2.a and Figure 6.2.b respectively). Card had a typical distribution of a surface with sinusoidal height profile (see Figure 6.2.c and Figure 6.2.d). Other textures like peas, which are the result of uniformly placed repeating primitives, did not usually suit any particular distribution (see Figure 6.2.e and Figure 6.2.f); to a certain extent, they recall normal distributions which have been widened out and flattened, presumably because of the estimation being affected by severe shadowing.

In general, it was observed that textures which are isotropic in appearance do not necessary have to follow Gaussian processes; conversely, those textures which do follow Gaussian processes seemed to be highly uniform textured surfaces with isotropic appearance.

Furthermore, it was noted that the measure of the q distributions is biased, as they present a negative skew in some cases. This asymmetry might be explained by natural processes such as erosion or abrasion, but it is believed that the main reason is attributable to the estimation method itself. Note that whereas the p estimation is based in symmetric photometric data, images from tilts 0° and 180°, the q estimation is only based in one image from tilt of 90°.

A description of the methodology employed to obtained these histograms as well as a complete list of histograms for surface partial derivatives p and q can be found in Appendix B. Table B.1, at the end of the appendix, lists the AASR values for all the textures, as well as the mean values of the surface slopes. Observe that both p and q mean always tends to zero — not strictly true for peanuts and gravel q map — which validates the assumption made in Section 1.2 that the surfaces considered in this dissertation are globally flat.

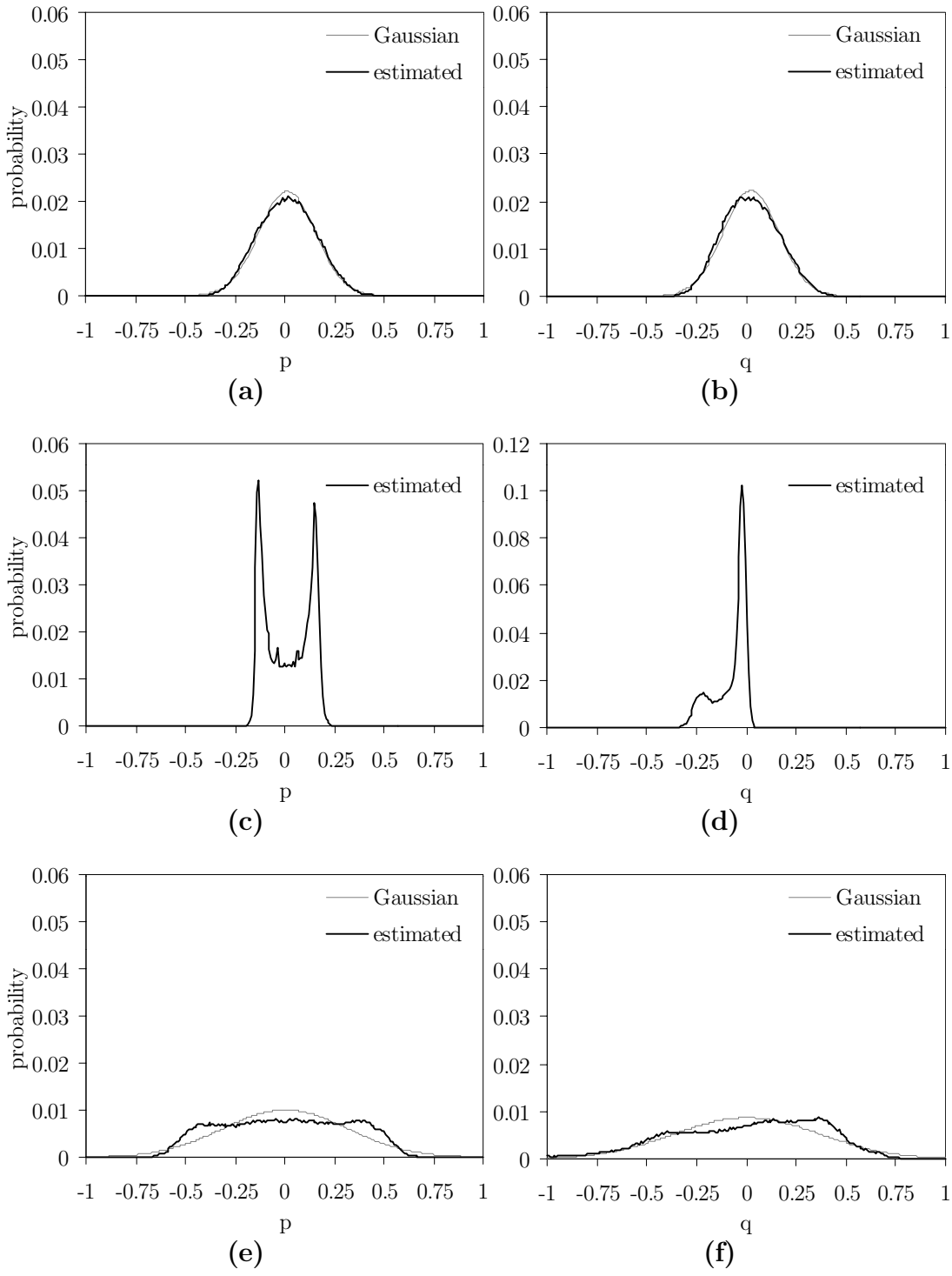


Figure 6.2 Histogram representation of the partial derivative fields for **(a)(b)** deposit3, **(c)(d)** card, and **(e)(f)** peas. Deposit3 and peas p–q distributions are compared with a best fit Gaussian. Card histograms are not compared since the non-Gaussian character is evident.

6.4 Accuracy of image prediction

The aim of the model-based scheme considered in the previous chapter is to predict the image, and ultimately the feature distributions, of a surface imaged under arbitrary lighting conditions. For natural textures the most restrictive of the assumptions made, in the author's opinion, is that shadowing is not significant. This section pays particular attention to this aspect, dividing the study up to three components corresponding to the response of image prediction to (i) changes in topological texture (or surface relief), (ii) changes in illuminant tilt angle, and (iii) changes in illuminant slant angle. All the experiments presented in this chapter were performed using modified versions of the shell script `pserror` (see Appendix C).

6.4.1 Influence of surface roughness

In this section, analysis of laboratory results are presented using two different criteria. First, in order to isolate the effect of surface roughness from other phenomena, investigation into different samples of the same material surface is carried out. Secondly, examination of different test surfaces attempts to discern whether the degree of roughness is enough to predict which one will perform worse.

Experiment 6.1

Four samples were formed by fracturing a large block of plaster. Different degrees of roughness were obtained for each one ranging from severe to smooth. A perspective view of the test surfaces can be consulted in Appendix A. Plaster1 is the roughest plaster fracture we obtained, and shadowing is severe at most slant angles. Plaster2 and plaster3 are intermediate in roughness between plaster1 and the later smoother fracture, plaster4, which was formed in the same way as the other three fractures, but the resulting block was also cured and smoothed, giving a ceramic type fracture. Several images of the plaster blocks were acquired for slant angles of 45°, 60°, and 75°. The

description and main statistics of the captured images into study are listed on Table 6.1. In this table, the value of AASR was estimated using Eq.(2.9).

Original images were compared with predicted ones for tilt angles varying between 0° and 360° in 30° steps at each slant case, testing a total of 144 cases[2]. The slant angle of the photometric image set was the same than that of the original image, i.e. 45°, 60° or 75°. An estimated error in the prediction of the images was computed using Eq.(6.1). As we were not interested in a particular case, but on the overall behaviour with respect to surface roughness, we averaged the errors along the tilt angle; we preferred, however, to plot the results differentiating between slant angles. Figure 6.3 shows the MSE of the test samples for different cases of slant angles. In all three sets, plaster1 was the most inaccurate, whereas plaster4 was the most accurate. The other two surfaces, plaster2 and plaster3, performed more closely since they are of similar roughness; as expected, plaster2 was slightly more difficult to predict.

To this point, we have shown the importance of roughness but we still need a direct link between roughness and error. In fact, certain resemblances between contours of each set illustrated in Figure 6.3 may be found. This would suggest that further study into a larger number of different textures could help to find a consistent relationship between surface roughness and error, if that is the case. This will be investigated in Experiment 6.2.

Description			Mean*			Standard deviation*		
Surface	Roughness	AASR	σ=45°	σ=60°	σ=75°	σ=45°	σ=60°	σ=75°
plaster1	severe	0.185	2907	2169	1320	668	831	804
plaster2	moderate	0.137	2978	2310	1363	503	635	649
plaster3	gentle	0.126	2626	1769	865	489	568	480
plaster4	smooth	0.0757	2888	2170	1327	287	349	367

* Depth: floating point (32 bits), Scaled maximum: 4096

Table 6.1 Characteristics of the test samples for Experiment 6.1.

[2] 12 tilt angles × 3 slant angles × 4 surfaces = 144 different cases.

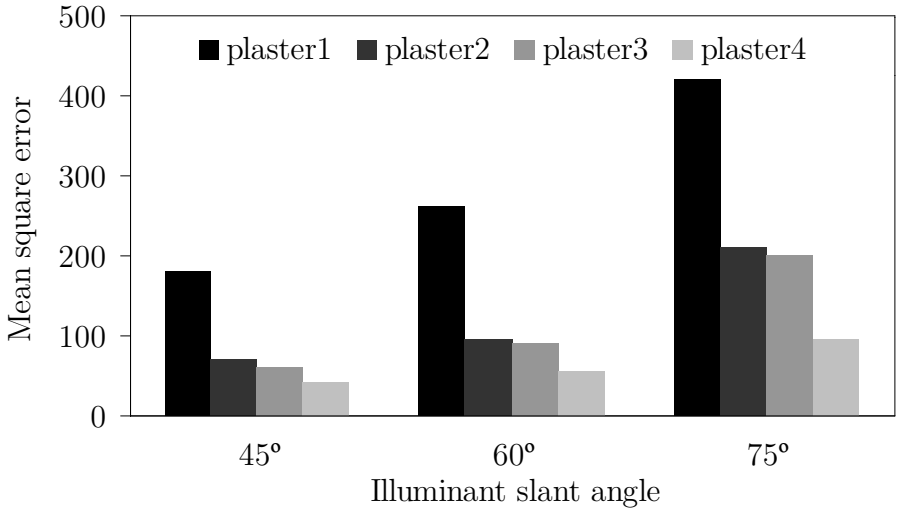


Figure 6.3 Mean square error of four plaster fractures with different degree of roughness at slant angles of 45°, 60°, and 75° (same slant angle of photometric data set and predicted image).

Experiment 6.2

The purpose of this experiment was to detect a possible relationship between the estimated slope angles and the error in the image prediction. As done in Experiment 6.1, three cases of slant (equal for photometric data and prediction) were considered and the computed MSE along the tilt angle was averaged. An estimation of the average slope angles was also calculated for each surface, again in terms of AASR. In order to find a possible correlation, we preferred to work with a linear error, so the square root values of MSE were considered in this case.

The experiment was repeated for 15 miscellaneous textures, providing three sets of 15 points shown in Figure 6.4. A sample image of the 15 textured surfaces can be found in Appendix A. Error curves were compared with typical tendencies, reaching a modest regression value of 0.90 (for slant of 60° and linear regression). There was therefore no clear relationship between these two variables, although the results indicated an upward trend in MSE with increasing AASR.

It is supposed that even though surface roughness is an important factor which always has a negative effect, there exists other secondary phenomena which also affect the accuracy of the prediction. Naturally, other surface properties, e.g. directionality, specularity, or even asymmetry, may also contribute to the error measure. If the premise is true, then the irregularly increasing error may be justified.

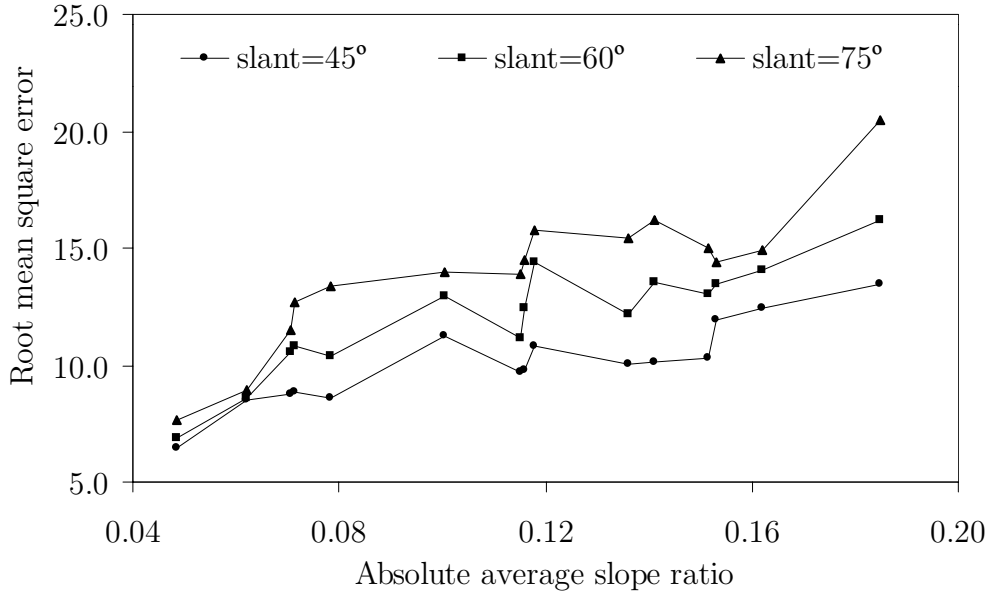


Figure 6.4 Root MSE vs. AASR for three cases of slant (45°, 60°, and 75°). Same slant angle of photometric image set and predicted images.

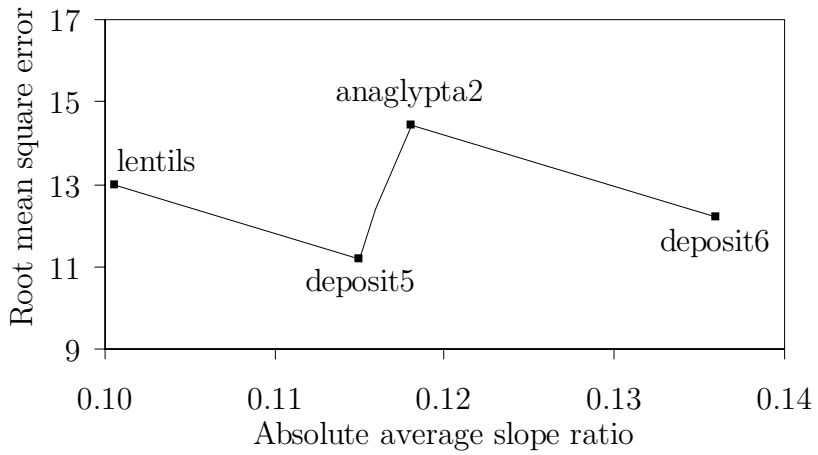


Figure 6.5 Irregular variation in the inaccuracy of image prediction (only slant angle of 60° is plotted).

In order to sustain the validity of this idea, let us consider the variation of error in the range of AASR from 0.10 to 0.14, which involves the following textures: lentils, deposit5, anaglypta2, and deposit6 (see Figure 6.5). Lentils is a texture formed by repeating primitives (red lentils) randomly placed over the scene; deposit5 and deposit6 are two isotropic surfaces formed by depositing plaster powder on a plaster surface of moderate and severe roughness respectively; anaglypta2 is a paper surface featuring irregular stripes that give a certain directionality align with the y-axis. Regarding the natural or artificial formation of the textures, Figure 6.5 illustrates the transition from repeating primitives to isotropic-like surfaces, and from directional to isotropic-like ones. In the first situation, despite the fact that lentils is not as much rough as deposit5, the former may be more affected by shadowing due to gaps between primitives, thus producing an additional error and being less accurate overall. The second situation showed that simulation of directional marks (like in anaglypta2) seems to be a more demanding task than the prediction of deposit6, which is the result of a number of random events and has a typical Gaussian distribution.

6.4.2 Effect of varying tilt

As it is aimed to design a model-based technique capable of enhancing the classification process under varying illuminant direction, the study of image prediction under arbitrary tilt conditions is extremely important. In this section the slant angle will be held constant at 45°, otherwise mentioned. Two different behaviours depending on the tilt angle into consideration are expected. For those values of tilt ranging from 0° to 180°, where the input data is contained, error curves will consist of two lobes with minimums at 0°, 90°, and 180°; for values of tilt from 180° to 360°, a parabola with maximum nearby 270° is very likely to occur. These two cases are investigated separately in the following experiments.

Experiment 6.3

The recovery images remained unchanged at 0°, 90°, and 180°, but the tilt of the illumination used in the image prediction was varied between 0° and 180° in 10° steps. Actual images were compared with those obtained by applying empirical rendering using the mean square error. After analysing the experimental results, three different types of curve were observed.

Case A: Isotropic surfaces

For those textures having a strong isotropic character — see Appendix A — the curves very likely followed a quadratic relationship with the tilt angle (see Figure 6.6). More specifically, the mean square error was found to be well approximated by the following equation:

$$MSE(\tau) \approx \begin{cases} k_1\tau(90-\tau) & 0 \le \tau \le 90 \\ k_2(\tau-90)(180-\tau) & 90 \le \tau \le 180 \end{cases} \tag{6.2}$$

where k_1 and k_2 adjust the curves to the correct fit, and are assumed to be independent of the tilt angle. This equation predicts that maximums occurs at tilt of 45° and 135°. Due to limitations on the sampling process, we did not have the opportunity to corroborate the MSE value for these two tilt angles, yet Figure 6.6 suggests that the proposition may be correct. Of all the surfaces, Eq.(6.2) was best satisfied by peanuts, gravel being the lowest accurate case.

In general, the constants k_1 and k_2 may not be identical, presumably because of the textural properties being not symmetrical. Again, gravel exhibits the most asymmetrical situation, having the biggest different between the two constants, whereas peanuts' MSE can be modelled by $k_1 = 0.405$, $k_2 = 0.414$, thus $k_1 \cong k_2$. In order to show the validity of Eq.(6.2) under proper conditions, i.e. nearly Lambertian reflection and isotropy, consider the peanuts texture since it was the most successful case. Figure 6.7 illustrates the goodness of fit between modelled and real power of error.

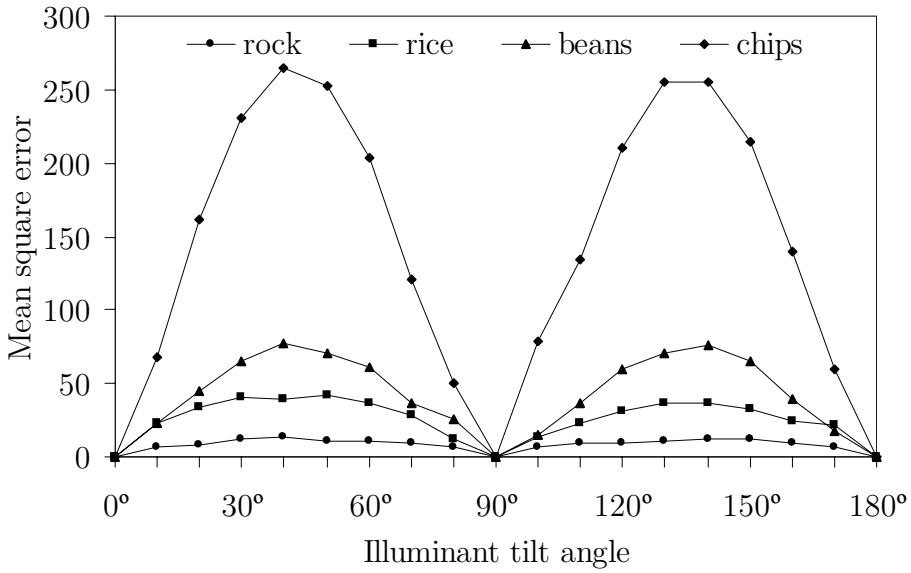
Case B: Anisotropic surfaces

On the contrary, it was observed that surfaces which are anisotropic in appearance have an important impact on the directional characteristics of their image textures. As a result, some orientations of the light source provoke more difficulties than others, and therefore the error of the image prediction does not follow the standard pattern obtained in case A. This unpredictable behaviour of the error is shown in Figure 6.8 for spaghetti1, card, ripple3, spaghetti2, and stucco. A sample of these textures can be consulted in Appendix A. The experiment was also performed on a paper texture, which showed similar error curve to that of Figure 6.8; however, it was decided not to show the curve because the MSE was much lower than for the order textures and the variation of error would not be observed.

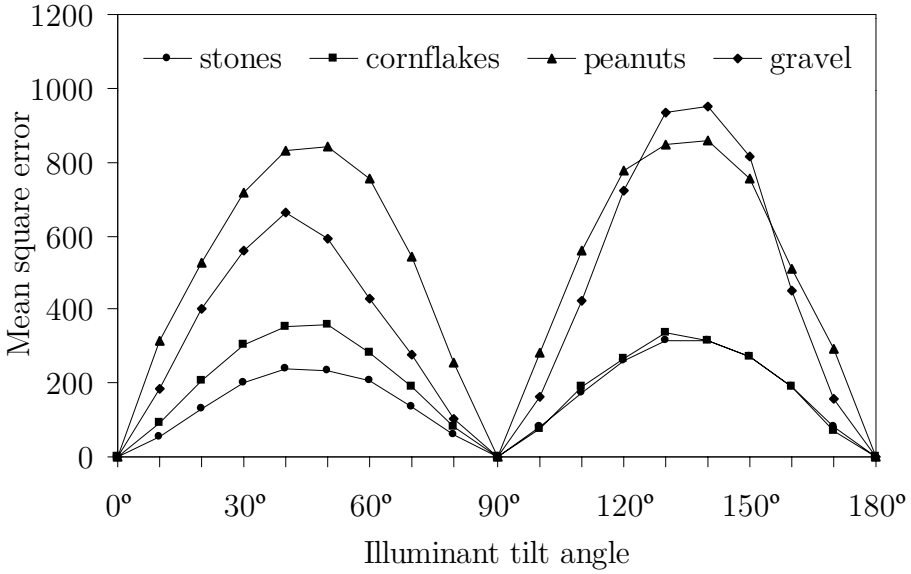
Case C: Specular particles

Like many other photometric methods, our *simple photometric stereo* technique gets confused in the presence of specular components. This effect was observed using a sandpaper texture, which is composed by a loose granular substance, typically resulting from the erosion of siliceous and other rocks. This texture has interesting optical properties since different facets become specular as the illuminant tilt angle changes. Given that the specular peaks were so narrow, the tilt function needed to be sampled densely. Figure 6.9 shows the MSE with respect to illuminant tilt angle sampled in 5° steps.

Note that sandpaper is the smoothest of all the surfaces, but its associated error is comparable to that of stones or chips, which are much rougher surfaces (see Table B.1 for exact values of $AASR$). The error may be then due to the specular particles of the surface, whose reflectance behaviour is very complicated to imitate. Initially, the sudden peak of error at tilt of 70° was attributed to a strong specularity at that angle, perceivable from the camera viewpoint. However, it may be also due to a photometric corrupted data obtained at that particular tilt angle.



(a)



(b)

Figure 6.6 Accuracy of image prediction for 8 isotropic surfaces under varying tilt angle: **(a)** MSE is low to moderate; **(b)** MSE is moderate to high. Although different level of inaccuracy is observed, the power of the error are similar to each other (in terms of shape).

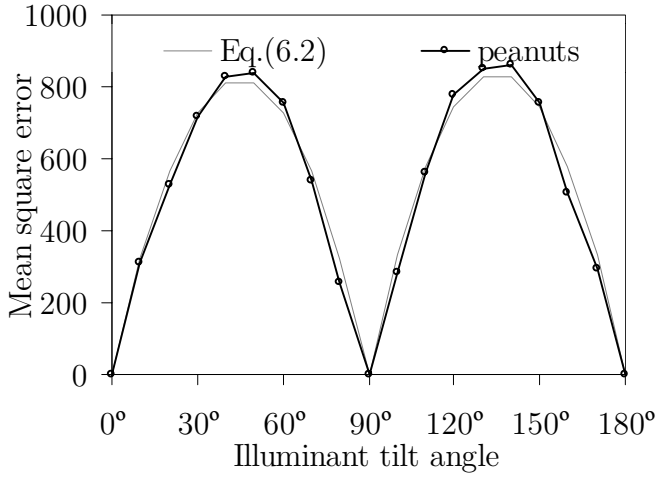


Figure 6.7 Best fit between Eq.(6.2) and peanuts' MSE (k_1=0.405, k_2=0.414).

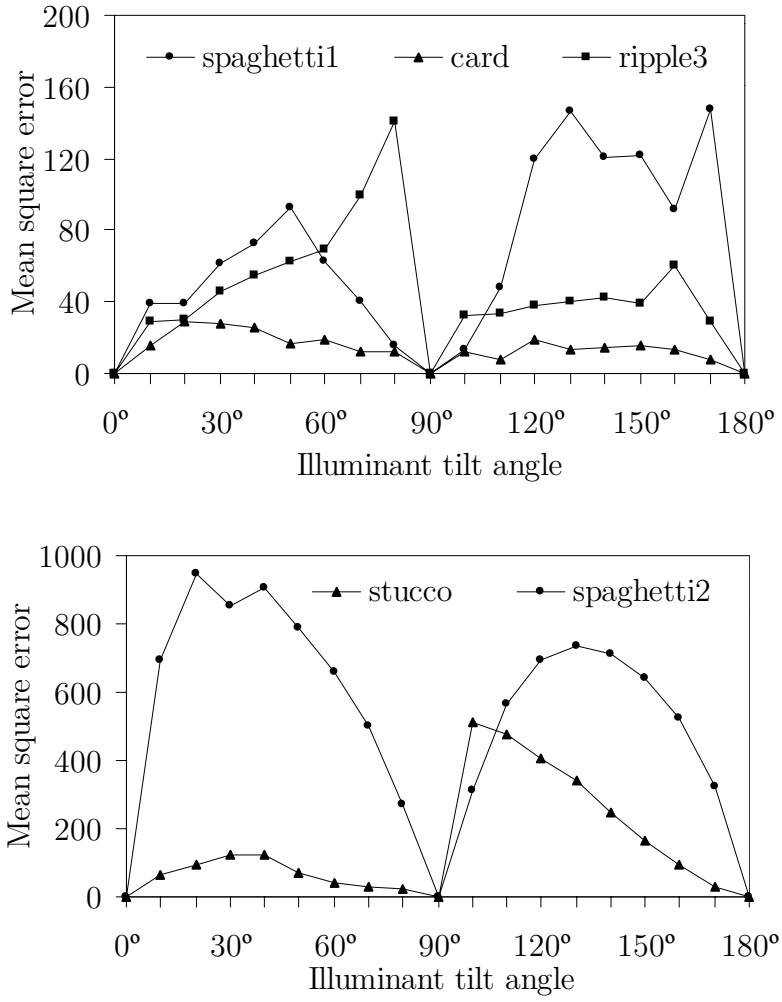


Figure 6.8 Accuracy of image prediction for 5 anisotropic surfaces.

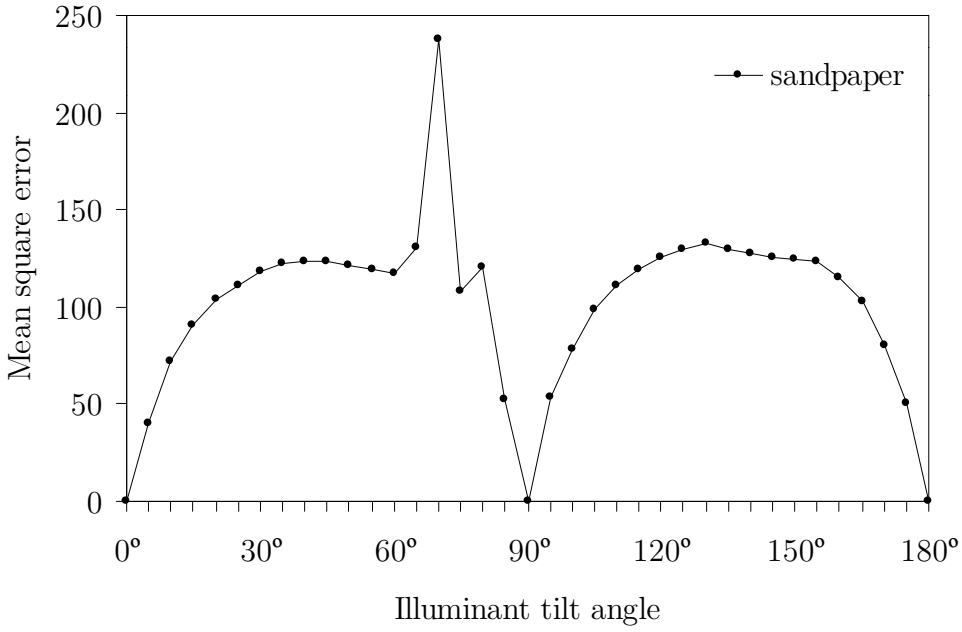


Figure 6.9 Image prediction in the presence of specular particles.

Experiment 6.4

This experiment studied the second range of tilt angles, i.e. from 180° to 360°. Three different textures were used, namely plaster2, ripple3, and sandpaper, as representative members of the isotropic, directional, and specular case respectively. The resulting errors in the simulated images are plotted versus illuminant tilt angle in Figure 6.10.

Once more, it is observed that for an isotropic surface with a diffuse reflectance function, the MSE can be approximated by an equivalent equation to that of Eq.(6.2). Specifically, the mean square error is given by

$$MSE(\tau) \approx k_3(\tau - 180)(360 - \tau), \quad 180 \leq \tau \leq 360 \tag{6.3}$$

where k_3 is an analogous constant to that of k_1 and k_2. Figure 6.10 shows the actual MSE of plaster2 prediction and best fit curved line modelled by Eq.(6.3) having an analytically estimated constant $k_3 = 0.022$.

Unlike the isotropic case, the error for ripple3 is highly unpredictable, depending chiefly on the particular directionality properties of the surface. Surprisingly, the image texture at tilt equal to 270° is not the less accurate, specially considering that the MSE is maximum at 260°. The error also suffers a sudden variation at tilt of 220°, being the range 210°–260° the more affected in general.

Finally, sandpaper had a typical error variation of isotropic surfaces, except that it might be more affected by specularities when illuminated from tilt of 255°. Due to specularities, the texture is again the worst precisely estimated, even though it is e.g. 30 times smoother than plaster2.

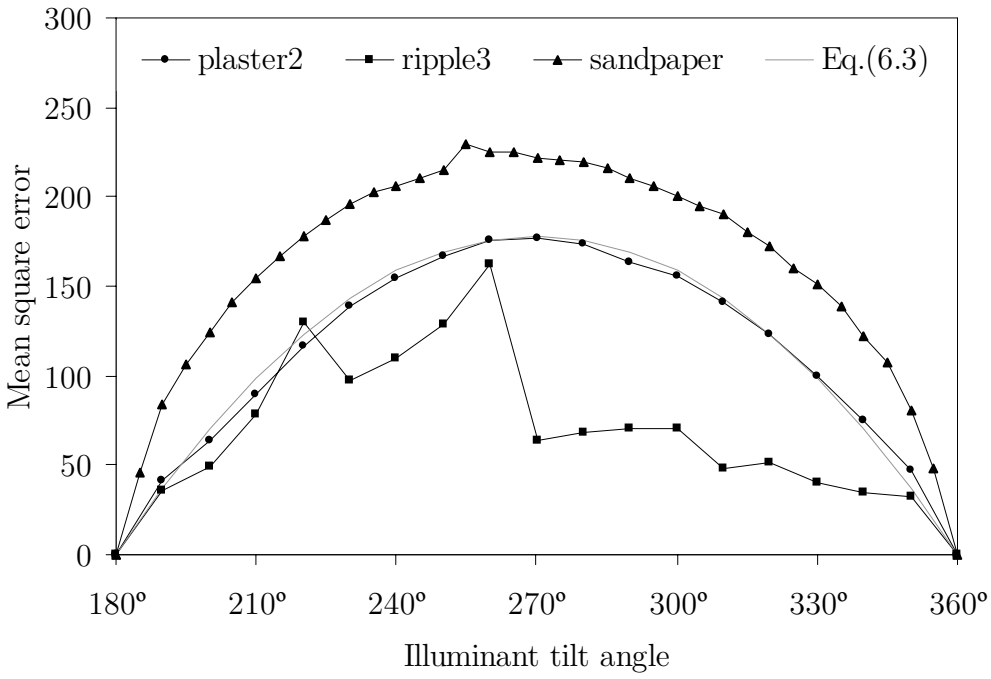


Figure 6.10 Inaccuracy of the image prediction for tilt angles ranging from 180° to 360°. Three different behaviours are observed: plaster2 is well fitted by the isotropic, diffuse case modelled by Eq.(6.3); ripple3 has an unpredictable error variation characteristic of directional surfaces; and sandpaper presents specular components.

6.4.3 Effect of varying slant

Although not directly, some previous experiments have already shown that higher slant angles tends to augment the inaccuracy of the estimation (recall e.g. Figure 6.3 or Figure 6.4). In this section, the impact of varying slant angle on image prediction is explicitly assessed.

In contrast to the tilt angle, the slant angle is not predetermined by the *simple photometric stereo* method, thus strictly it can take any value in the range 0°-90°. The method established, however, same slant angle for all three images constituting the dataset. To ease the discussion of the experiments, the slant angle used in the data set will be simply refer to as *input slant*, whereas the slant angle of the predicted images will be refer to as *output slant*. In this respect, two distinct situations are identified: (i) same input and output slant angles, and (ii) different input and output slant angles; both of them will be examined in Experiment 6.5 and Experiment 6.6 respectively.

The ultimate purpose of this section is to determine which case of input slant angle is the best option as a standard selection for the dataset. In other words, it is tried to find the input slant angle more capable on average of predicting any given output slant angle. This uncertainty leads to the following compromise: on the one hand, it is desirable the slant angle to be as large as possible, because illumination directions which are close to a perpendicular to the surface plane, i.e. low slant angles, provide less information since the perceived image textures are more similar. On the other hand, large slant angles, although providing more information to the photometric stereo, intensify the level of shadowing, thus also deteriorating the precision of the estimation. This idea, intuitively understandable, can be mathematically justified as follows.

Section 4.2.2 explained the fundamentals of photometric techniques and reduced the problematic to that of solving $\mathbf{L}^{-1} \cdot \mathbf{I} = \mathbf{D}$ —see Eq.(4.3) for correspondence. For a photometric stereo technique using three sources of light and assuming Lambertian reflection, the illuminant matrix $\mathbf{L}$ is given by

$$\mathbf{L} = \begin{pmatrix} \cos\tau_1 \sin\sigma_1 & \sin\tau_1 \sin\sigma_1 & \cos\sigma_1 \\ \cos\tau_2 \sin\sigma_2 & \sin\tau_2 \sin\sigma_2 & \cos\sigma_2 \\ \cos\tau_3 \sin\sigma_3 & \sin\tau_3 \sin\sigma_3 & \cos\sigma_3 \end{pmatrix} \tag{6.4}$$

In particular, this matrix is significantly simplified for the simple photometric stereo method, since it considers same illuminant slant angles and tilt angles of 0°, 90, and 180°. Then, it is easy to demonstrate that the determinant of that illuminant matrix is only a function of slant angle given by

$$\det\ \mathbf{L} = 2\sin^2(\sigma)\cos(\sigma) \tag{6.5}$$

Regarding Eq.(6.5), we make two observations: first, our problem of surface recovery cannot be solved for slant angles of 0° and 90°, because the determinant must be different from zero. Secondly, and more important, observe that the solution will be inversely proportional to the determinant of the illuminant matrix. Hence, noting that the images of the data set are always affected by noise, the lower the determinant is, the more sensitive the method is to noise. In effect, let I_1 and I_2 be two data images affected by noise such that

$$\begin{aligned} I_1 &= i_1 + n_1 \\ I_2 &= i_2 + n_2 \end{aligned} \tag{6.6}$$

where $n_{1,2}$ represents additive noise produced during the imaging process, and $i_{1,2}$ represents the incident image. Let us express the contribution of these two images to the solution as

$$\frac{1}{\det\ \mathbf{L}}(\ldots + i_1 - i_2 + n_1 + n_2 + \ldots) \tag{6.7}$$

It is clear that even the addition of terms with opposite sign will degrade the S/N, due to the additive character of the noise. In order to soothe this degradation, it is attractive to work only in the range of slant angles which maintains the factor $(1/\det\ \mathbf{L})$ as low as possible. Figure 6.11 illustrates the variation of this factor with respect to the illuminant slant angle.

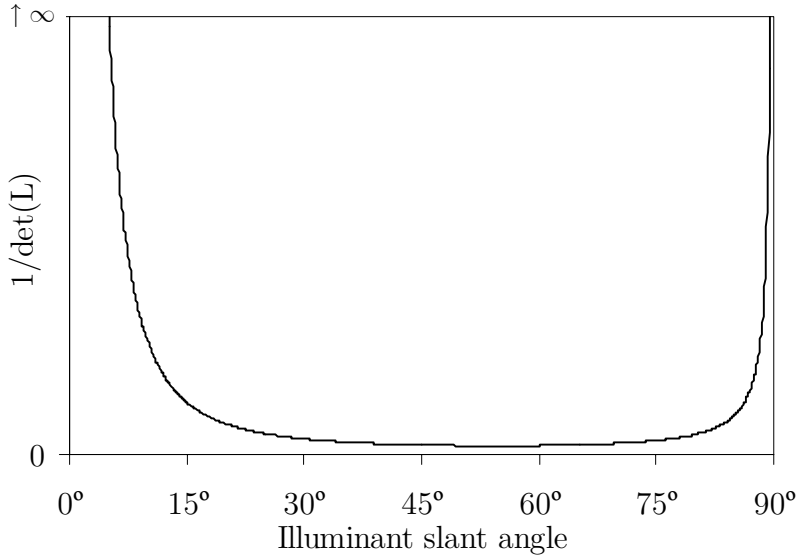


Figure 6.11 Variation of the illuminant matrix determinant with respect to slant angle. High values of 1/det(**L**) are identified as a source of degradation, whilst low values are more suitable for a successful surface recovery.

Approximately, the above premise is satisfied by slant angles between 30° and 75°, and therefore this range will be adopted as the starting point for further investigation.

Experiment 6.5

This experiment assessed the accuracy of the image prediction with respect to illuminant slant angle, restricting the study to those cases where the input slant angle is equal to the output slant. In order to examine the performance of the photometric technique, we decided to use a single tilt angle of 270° for the rendering, since it usually represents the hardest case to deal with. Consequently, note that the MSE curves in this section trace the maximum error limit for each particular texture, and that normally the error will be found below these curves.

Several physical texture samples were used in this experiments (see Appendix A). Images were obtained from each sample using illumination slant

angles ranging between 30° and 75° incremented by 15°. All textures were illuminated from tilt angles of 0°, 90°, and 180° to form the input data. In addition, another image at tilt angle of 270° was captured in order to calculate the MSE of the rendered image at that angle.

Generally, low slant angles provided image textures with small differences amongst the pixels' intensity. As the angle of illuminant became more and more shallow, shadowed areas became bigger, thus increasing the image contrast. However, this tendency started to saturate and even decrease from a certain slant angle, typically $\sigma > 60^{\circ}$, probably because of a general state of darkness affecting the whole surface scene. As we aimed to calculate an objective measure of error, it was recommendable to compensate for these changes in image variance; the original measure of MSE was therefore divided by image variance at each slant angle into study, giving what we call *normalised* MSE.

While most of the textures performed in a similar way to that illustrated in Figure 6.11.a —here only some of the results are shown to avoid redundancy— a few number of textures presented different MSE variation throughout the slant angle axis, as shown in Figure 6.12.b. From the results, it seems that surfaces such as plaster4 and deposit1, which are formed by a series of random processes, and consequently have nearly Gaussian distribution (see Appendix B), follow a monotonously ascending tendency.

Textures formed by repeating primitives, like mixture and peas also tend to correspond to similar error curves; however, when the size of the primitive patterns was large compared to the dimensions of the image, like in pebbles, the outcome was quite unpredictable, it depends on the case of study, and it may be due to the combination of many factors, chiefly severe shadowing.

Directional surfaces, e.g. ripple1 and specially ripple3, also showed different trends, and again trying to find a tendency towards a particular type of behaviour was very complicated.

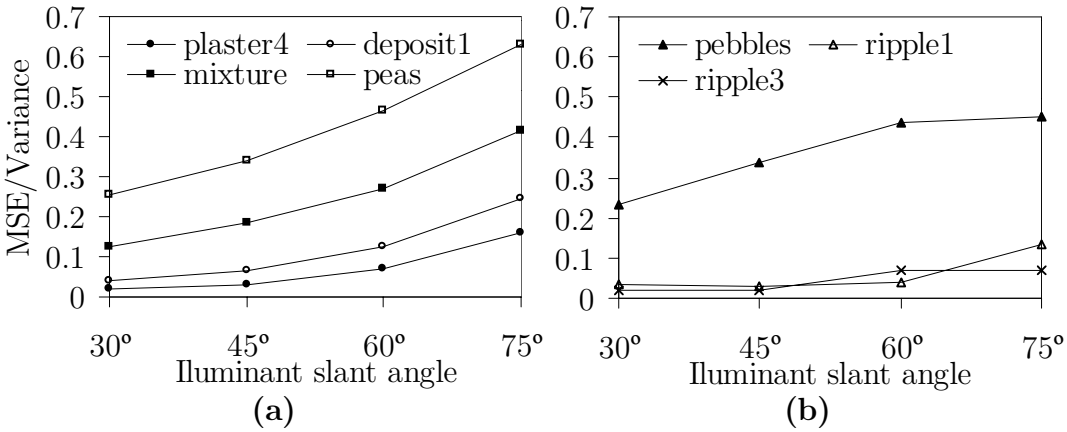


Figure 6.12 Normalised mean square error against illumination slant angle. Two different behaviours are detected: **(a)** the error rises at an exponential rate; **(b)** the error variation is uncertain, yet almost always increasing.

In any case, even though different trends was observed, the imprecision of the simple photometric stereo had a clear tendency to increase for higher slant angles. Finally, Figure 6.13 compares real image textures across a rough, diffuse, plaster surface to rendered images as illuminant slant angles varies, and pictures how renderings turn out to be less realistic for shallower lighting directions. Note that the illustrated example was one of the worst predictions obtained.

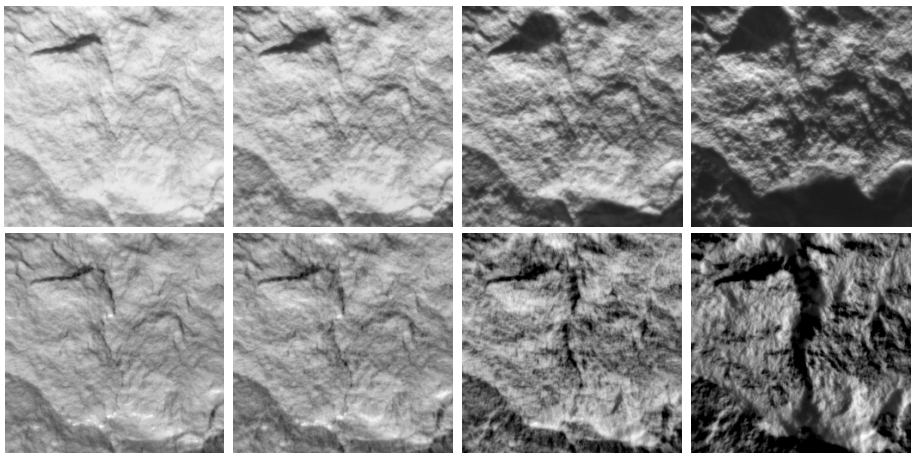


Figure 6.13 Visual comparison between actual and predicted intensity images at four slant angles. Top row: plaster1 illuminated from slant 30°, 45°, 60°, and 75°. Bottom row: simulated plaster1 rendered at identical slant angles.

Experiment 6.6

From the results of Experiment 6.5 it appears that the best input slant angle is 30°. However, the previous experiment only studied prediction tasks performed at the same slant angle used for modelling the surface, which represents a very unlikely condition in a real application. The only intention was to show that even with same input and output slant, a rise in its value has normally negative consequences in image prediction. Hence a more pragmatic, general situation is considered next, where the input and output slant are different. The experiment was carried out in three steps.

First, given a particular texture, images at slants of 30°, 45°, 60°, and 75° were rendered from photometric data at slants of 30°, 45°, 60°, and 75° in all possible combinations (giving 16 input-output slant pairs). For each case the experiment computed the MSE in the prediction; the set of results were gathered in a 4×4 matrix of errors, where the rows represent input slants and the columns represent output slants. The procedure was repeated for the following 17 textures grouped into three types:

- isotropic textures: deposit1, deposit2, deposit3, deposit4, deposit5, deposit6, plaster2, and plaster4.
- directional textures: anaglypta1, anaglypta2, anaglypta3, ripple1, ripple2, and ripple3.
- repeating primitives textures: mixture, peas, and pebbles.

Table 6.2 exemplifies the matrix of errors of a member of the isotropic and directional texture types, deposit5 and ripple2 respectively. A common feature, not only of the representative results shown here but of all the textures evaluated, was that the values of the elements above the diagonal were usually lower than their corresponding symmetrical ones below the diagonal. It suggested that the photometric scheme coped better with a *forward* prediction (input slant < output slant) than a *backward* one (input slant > output slant).

Input slant \ Output slant	**30°**	**45°**	**60°**	**75°**
30°	*55.26*	192.2	246.0	385.1
45°	204.2	*93.66*	160.1	344.8
60°	363.4	209.1	*124.7*	313.3
75°	1145	828.1	527.5	*193.5*

(a)

Input slant \ Output slant	**30°**	**45°**	**60°**	**75°**
30°	*46.53*	83.55	157.1	250.5
45°	52.54	*71.74*	153.3	248.7
60°	348.6	246.4	*70.26*	172.2
75°	881.0	678.3	366.4	*62.25*

(b)

Table 6.2 Matrix of MSE on the prediction of **(a)** deposit5 and **(b)** ripple2. Diagonal elements are in italics to emphasise same input and output slant.

At this point of the analysis it was not simple to conclude which case of input slant angle was more suitable as a standard selection, mainly because the error measure depended on three variables: the texture, the output slant, and the input slant. Then a second level of analysis was developed.

In order to eliminate the dependence on output slant angle the mean square errors in each row were averaged. The results are plotted in Figure 6.14 for each set of textures (in ascending order of $AASR$) and different input slant angle. It was observed that the rendering process from input slant of 45° was typically the most successful case, except for deposit1, deposit3, and deposit4, whose best case was slant of 30° (although with no more than a slight difference). In addition, Figure 6.14 corroborated once more that higher slope angles do not directly imply a worst image prediction, as observed in Experiment 6.2.

Finally, Figure 6.15 illustrates an overall average error where the dependence on the particular texture is also eliminated; each point represents the average MSE (over all output slants and textures) plus-minus the standard deviation of the measures at each input slant angle. It is concluded that the best input slant angle might be found in the range 30°–45°, but the number of samples used was not enough to precisely determine a single optimum input slant angle.

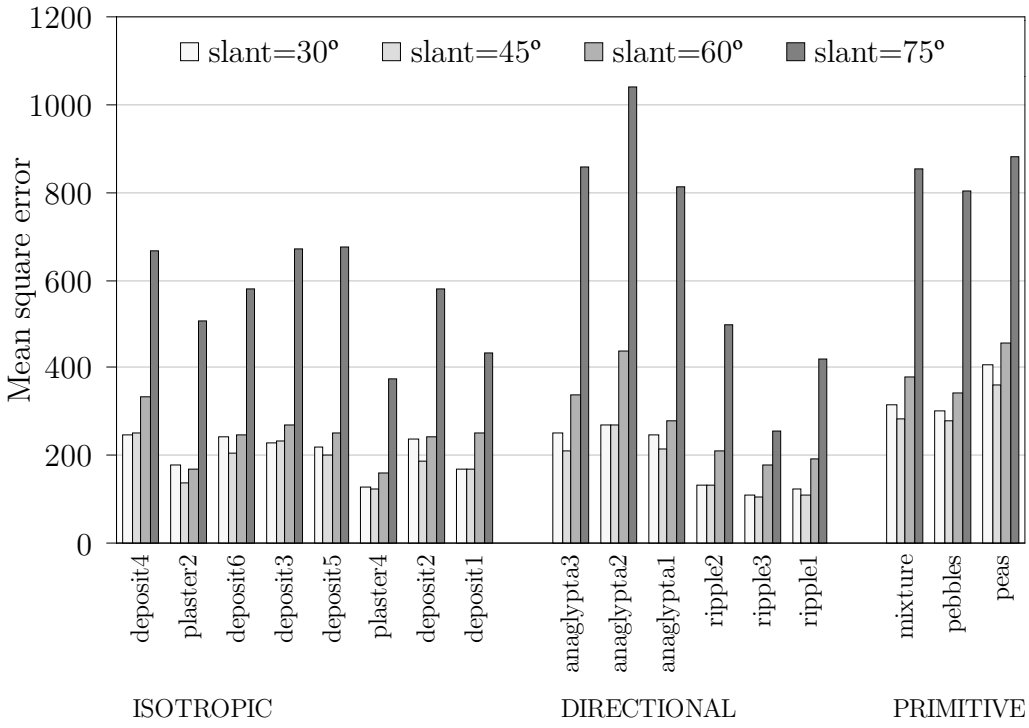


Figure 6.14 Average MSE of 17 test textures for input slant angles of 30°, 45°, 60°, and 75°. The results are separated with respect to three texture types, namely isotropic, directional and repeating primitives; each set being arranged in ascending order of $AASR$.

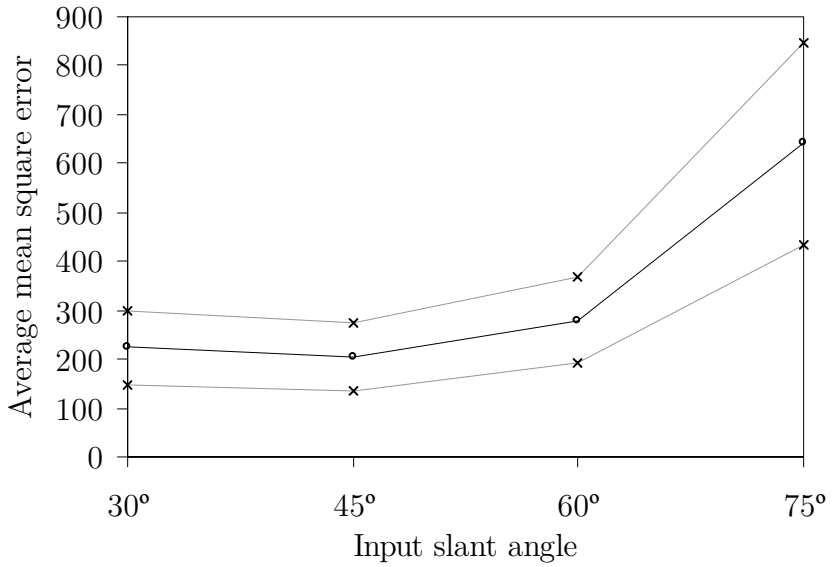


Figure 6.15 Average MSE over all the textures and output slant angles at input slant angles of 30°, 45°, 60°, and 75°. The standard deviation of the measure is also plotted.

6.5 Discussion

Previous sections have evaluated the accuracy of image prediction concerning different aspects. Many results, chiefly in terms of mean square error, have been obtained, and some conclusions have been drawn. To this point, the method, understood as an individual component, has been criticised and to the question whether the rendered images are perfectly correct, the answer is in general no. However, we must keep a sense of perspective about what it is intended. We remind that the training stage prior classification will be performed on the basis of these predicted images. Therefore, the key question is whether the simulated images are accurate enough to enhance the classification performance with respect to a naive classification system[3]. This question will be explicitly answered in next chapter, but first let us justify here the application of such model-based method to texture classification with the following example.

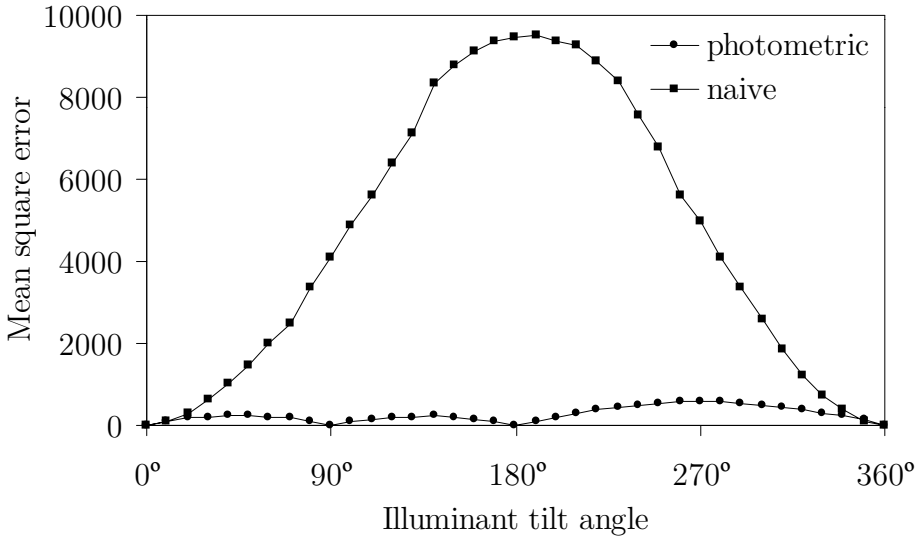


Figure 6.16 Errors in the plaster1 data used to train the classifier for a naive method, which only uses one image at tilt=0°, and photometric stereo, which uses three images at tilt=0°, 90°, and 180°. The horizontal axis represents the tilt angle at which the classification would be performed.

[3] Naive classifier is refer to a classification approach which always uses the same image texture for training regardless of the particular illumination condition.

Figure 6.16 compare the naive and model-based approach in terms of error in the input data used for training the classifier. The graph was obtained as follows. The input and output slant angle were held constant at 45°. For the naive method, a captured image of the texture at tilt of 0° was compared with an image of the texture at tilt angles between 0° and 360° in 10° increments; in so doing, we calculated the MSE of the hypothetic training image that a naive classifier would used, which is always the 0° image instead of the actual image at the particular tilt angle. For the photometric technique, the curve was obtained using the same methodology as Experiment 6.3. The testing was carried out on plaster1, which was one of the worst predicted cases studied in this chapter, and even under the circumstances the improvement is substantial.

Chapter 7

Classification performance

Chapter 6 was concerned with the evaluation of the model-based rendering as an individual component of the entire system. Although the results already encouraged the use of such a method for classification purposes, it is still needed to give a more convincing evidence that this approach is in fact suitable for coping with the problem of lighting dependency. The objective is then to experimentally confirm the ability of photometric classification to the stabilisation of misclassification rate at a reasonable level, ideally close to the best possible classification. Before proceeding, it is noteworthy to mention the difficulty to generalise results at this stage owing to strong interdependences of samples during classification.

7.1 Introduction

Unfortunately, the effect of illuminant variation on classification performance is highly dependent on the application. Changes in lighting conditions cause displacements of class members within the feature space, but these movements are only significant if they cross decision surfaces, the position of which is dependent upon the characteristics of the original training set. Hence the number of classification errors is a function of the feature measure selected, the number of textures, the properties of the textures, and obviously the illuminant tilt angle and slant angle.

The testing and developing of classification tasks representative of all the possible applications is quite out of the question. The results and conclusions regarding improvements introduced by the model-based approach should be considered in context, even though they are believed to give good proof of enhancement in a diversity of situations.

In order to make the results comparable, the focus will be on the approach to the classification problem, keeping the general configuration of the system as similar as possible. Figure 7.1 shows the system set-up; depending on the data available to the training session, three different situations are defined:

best case training is based on real data imaged under the appropriate illumination condition.

naive classifier regardless of the particular lighting conditions, the classifier is always trained with the same image which represents a single illuminant direction. By default, this image is held constant at tilt angle of 0° and slant angle of 45°.

model-based training is based on rendered data at the intended illuminant direction.

By means of this configuration, different classified images are separately obtained, and their quality can be evaluated with respect to each other, as the case may be. Given a particular classification task, misclassification of the model-based approach is presumed to be due to both inadequate image prediction and limitations of the classifier itself, whereas the best case classification may be due solely to the latter. Thus, initially it is logical to believe that the difference between the model-based classification failure and that of the best case may be due exclusively to the inaccuracy of image prediction. The classification error of the naive classifier is expected to suffer a quickly rise as the angle of illuminant is moved away from its original direction used during training, being naturally the worst case out of the three.

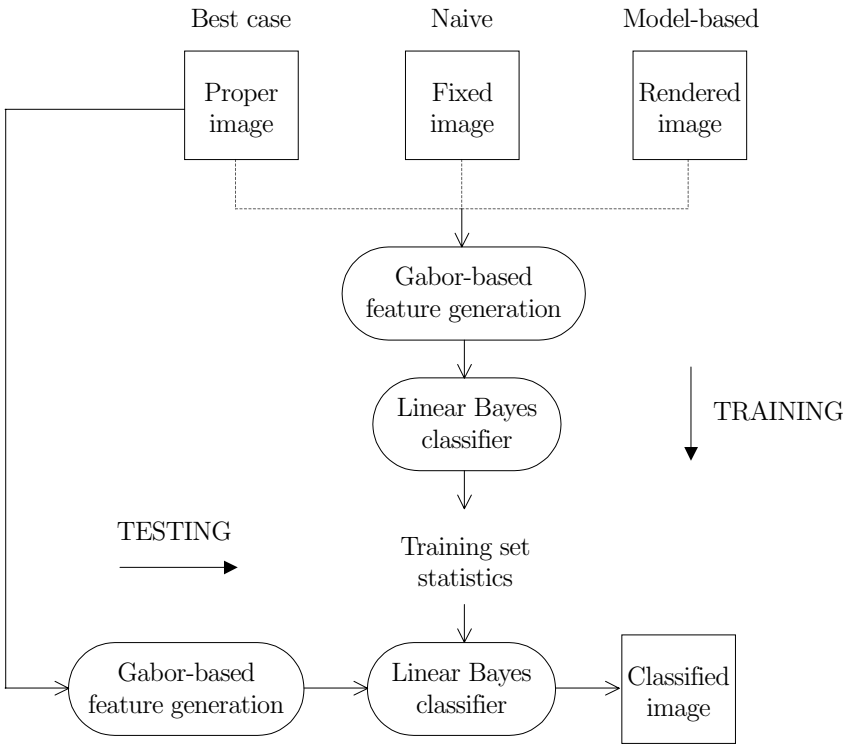


Figure 7.1 Three different approaches to texture classification: best case, naive, and model-based classifiers.

The purpose of the work described in this chapter is not to develop an optimum classifier, but to evaluate the benefits of using a simple photometric stereo technique on a conventional classifier. In previous chapters, effort was not invested in optimising either the feature extraction or the discriminant function— they were merely selected on the basis of popularity in the literature and ease of implementation. Rather the aims of the following sections are:

- to define the scope of the investigation and select adequate test montages,
- to evaluate classification performance under varying lighting conditions,
- to investigate into a possible relationship between inaccuracy of image prediction and misclassification of the model-based approach, and
- to assess the consequences of increasing the number of textures.

7.2 Experimental framework

The work reported here consists of classifying some multi-textured images under varying illuminant tilt and slant angles. The number of textures, configuration of the composite, and borders between textures vary according to the case of study. The variation of illuminant direction is simulated by changing either the tilt or the slant angle between training and classification sessions. The analysis of the two variables at the same time is not pursued.

The proposed procedure is a supervised method, i.e. it is assumed that representative texture samples are given, as done usually in real segmentation or classification problems. In addition, the classifier is trained using the entire montage rather than individually extract statistics from each texture image. Hence the procedure is only testing the ability of the features to describe the textures but not the ability to generalise from the training data.

7.2.1 Selection criteria

The selection of the composite texture images sought to meet the following criteria:

- Overall, the classifier should perform well for the best case approach. In this manner, the initial level of difficulty inherent in a particular situation is maintained as low as possible.

- To the contrary, the naive classifier should exhibit high misclassification rates under varying lighting conditions. Thereby, the effect of illuminant variation on the feature set is guaranteed to be significant in an uncompensated case.

Then, if under these two circumstances the model-based approach was able to enhance the performance of the given classification task, its merits would be really appreciated.

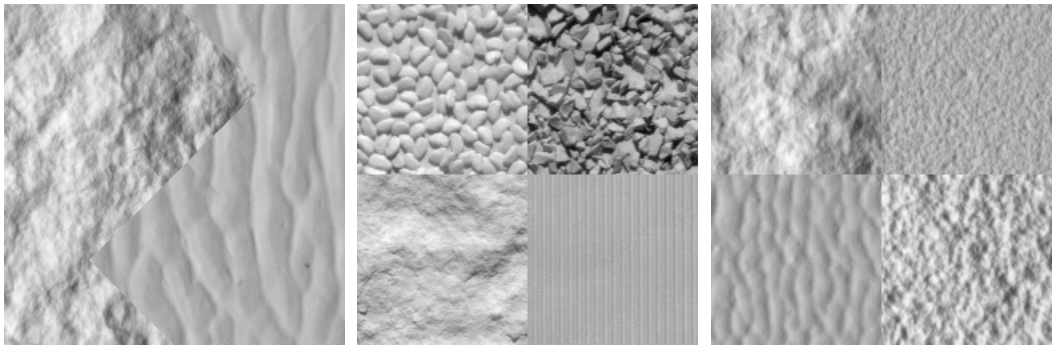
7.2.2 Test images

The same set of textures presented in Section 6.3 are used here to form the composite textures of Figure 7.2. The prediction of most of them has been explicitly assessed in Chapter 6; a sample of each texture can be found in Appendix A. A histogram description of their surface derivative fields can be consulted in Appendix B. None criterion with respect to texture granularity was adopted; consequently, some images are apparently bigger than others when compared with the image dimension.

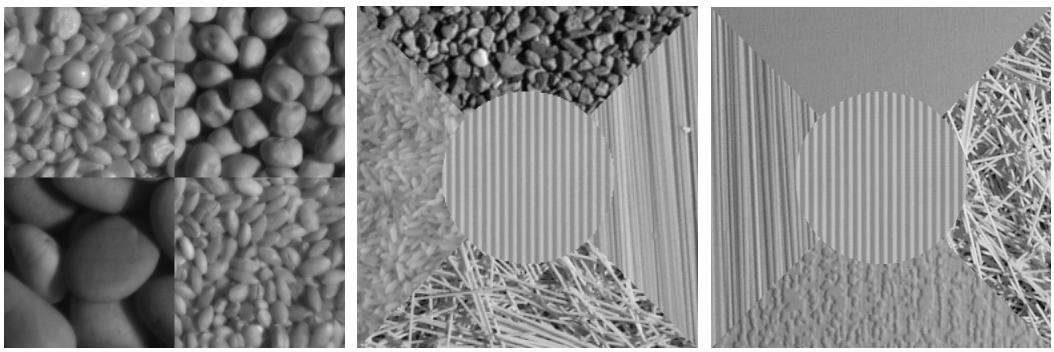
Five types of configurations were selected, each one made up of two, four, five, eight and twelve textures. All the test images are 512×512 monochrome images which are equally divided up to the specified number of regions so that all the textures have the same weight relative to the image. Table 7.1 summarises the characteristics and composition of the montages used in the experiments. Notice that the montages were named following the nomenclature *montage*, plus *number of textures*, and plus *key reference* to Figure 7.2.

Descriptor	**Number textures**	**Figure reference**	**Composition** (in clockwise direction)
montage2A	2	Fig.7.2(a)	plaster3, ripple3
montage4B	4	Fig.7.2(b)	beans, chips, card, rock
montage4C	4	Fig.7.2(c)	plaster2, deposit1, deposit4, ripple2
montage4D	4	Fig.7.2(d)	mixture, peas, barley, pebbles
montage5E	5	Fig.7.2(e)	gravel, spaghetti1, spaghetti2, rice, card
montage5F	5	Fig.7.2(f)	paper, spaghetti1, stucco, spaghetti2, card
montage8G	8	Fig.7.2(g)	deposit3, ripple2, anaglypta2, anaglypta1, lentils, barley, anaglypta3, peas
montage12H	12	Fig.7.2(h)	mixture, ripple1, deposit2, anaglypta1, lentils, barley, peas, pebbles, plaster1, ripple3, sandpaper

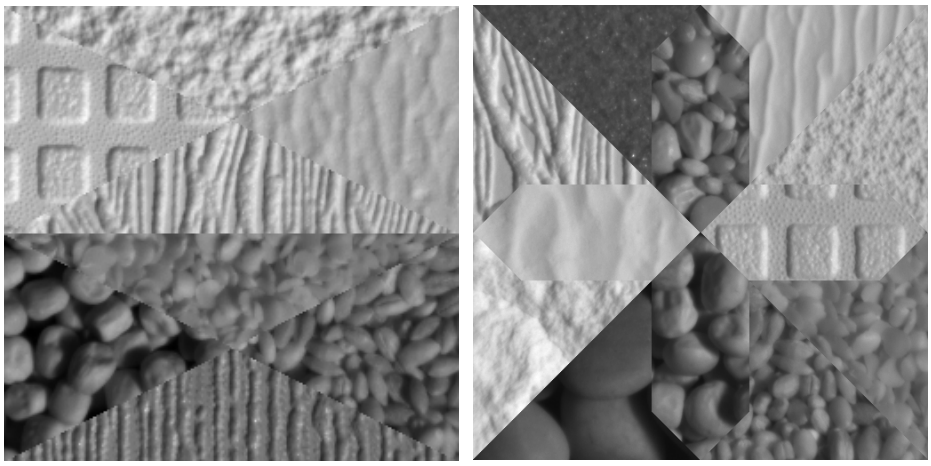
Table 7.1 Summary of montages and characteristics.



(a) (b) (c)



(d) (e) (f)



(g) (h)

Figure 7.2 Composite texture images used in the experiments. Each image will be referred to hereafter as follows: **(a)** montage2A, **(b)** montage4B, **(c)** montage4C, **(d)** montage4D, **(e)** montage5E, **(f)** montage5F, **(g)** montage8G, **(h)** montage12H.

7.3 Accuracy of classification

In this section, the classification performance is thoroughly assessed under varying lighting conditions. It is believed that next experiments give the most convincing proof of enhanced classification. The experimental data of these experiments was obtained using modified versions of shell scripts `shellbest`, `shellnaive`, and `shellmbased`, which are listed in Appendix C.

7.3.1 Variation of tilt angle

The three approaches to texture classification introduced in Figure 7.1 are put to the test with respect to illuminant tilt angle. Its companion, the illuminant slant angle will be held constant throughout the experiment, thus only one variable is analysed at a time. We note that three montages are explicitly considered here, gathered as Experiment 7.1, even though the tilt response was also studied for other montages whose results will be discussed in other sections for convenience.

Experiment 7.1

Composite images montage2A, montages4B, and montage5F were used (see Figure 7.2). All three classifiers were trained with textures captured or rendered with slant angle of 45°. The tilt angle of the test sets was varied first in 10° steps from 0° to 360° (only montage2A) in order to see the overall tilt response, and then in 10° steps from 0° to 180° (for the other two montages), thus concentrating in the first half, where the data set of the model-based approach is contained. The resulting classification error rates are shown in Figure 7.3.

Improvements introduced by the model-based classification are evident in all three classification tasks; not only in few occasions, but at almost all tilt angles, the model-based misclassification rate is lower than that of the naive classifier. It is particularly remarkable the error reduction that takes place approximately between tilts of 50° and 120° —see classification results at tilt of 90° on montage4B in Figure 7.4. Furthermore, the model-based classification

maintained a stable level of error through the tilt axis and certainly follows the line described by the best case, coinciding at tilts of 0°, 90°, and 180°, for which the image prediction was most effective. To a certain extent, the model-based approach made the same errors as the best case, as well as other errors. This led to question whether the differences in the errors are only a consequence of image prediction inaccuracies. Section 7.4 will try to answer this matter.

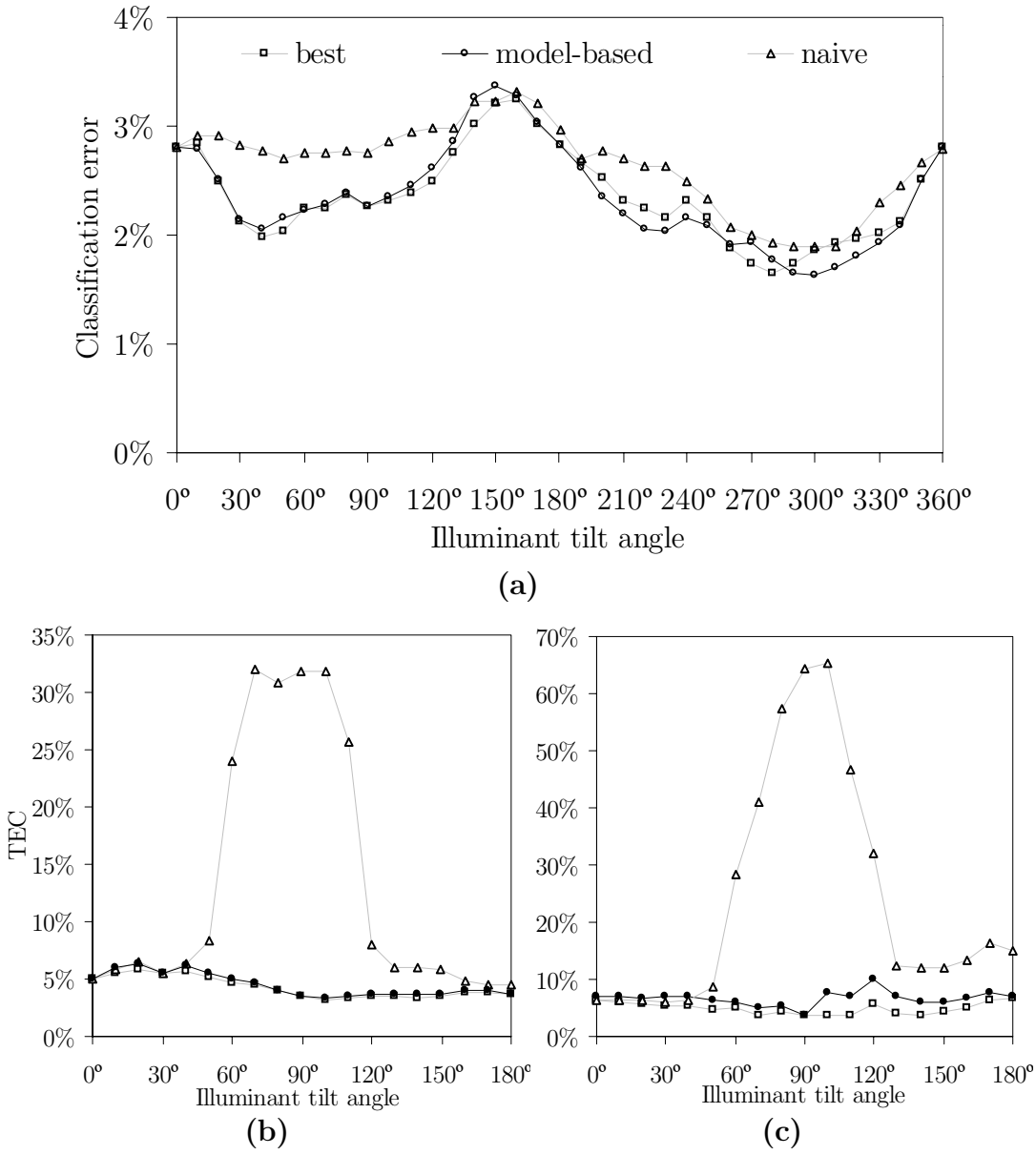
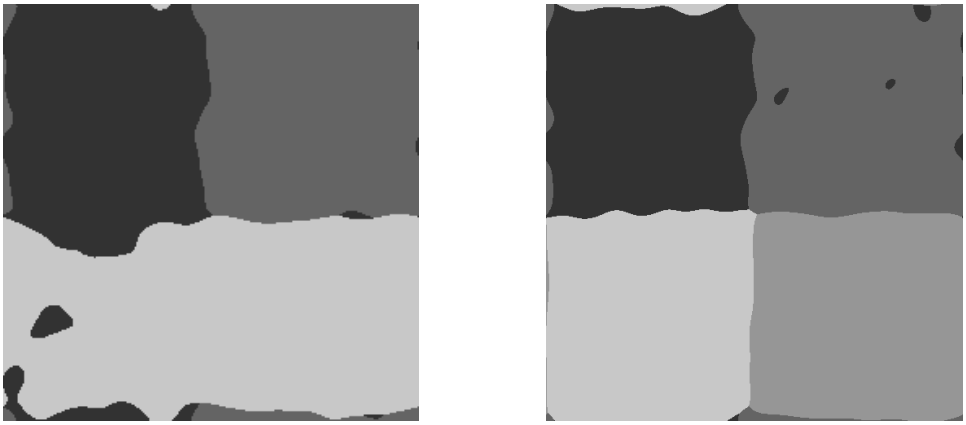


Figure 7.3 The effect of illuminant tilt variation on **(a)** montage2A, **(b)** montage4B, and **(c)** montage5F. The legend is the same for all 3 graphs.



(a) **(b)**

Figure 7.4 Class map at tilt angle of 90° for **(a)** the naive and **(b)** the model-based classifiers (test image: montage4B). While the naive classifier fails in discriminating between textures in the lower part, the model-based method provides a good classified image.

Also perceptible is the fact that in some cases the misclassification rates are even lower than for the so-called 'best case'. The explanation is that the pixels' Gaussian probability assumption of having a specific feature vector (see Section 4.2.3 for details) is actually suboptimal, specially considering the variety of textures presented to the classifier—Appendix B shows that many textures do not approximate Gaussian processes. Hence the TEC optimisation on each classification approach is affected by the data reliability regarding the assumption at each illuminant direction.

Finally, Table 7.2 shows the average misclassification rates for each combination of approach and test image, and the resulting reduction in the number of errors for each classification task. The reduction factor calculates the improvement with respect to the naive classification, and takes values in the range [0,1]. Its definition is given by

$$\text{reduction } (R) = 1 - \frac{ABE}{ANE} \tag{7.1}$$

where ABE is the Average Best case total Error of classification, and ANE is the Average Model-based total Error of Classification.

Classification approach	Test image		
	montage2A	montage4B	montage5F
Best case	2.34%	4.28%	4.93%
Model-based	2.36%	4.46%	6.63%
Naive	2.65%	13.3%	24.0%
Reduction	0.109	0.665	0.724

Table 7.2 Average total error of classification for best case, model-based and naive classifiers (test images: montage2A, montage4B, and montage5F). The reduction factor introduced by the model-based classifier with respect to the naive classifier is also shown.

Sometimes it occurs that a classification task is not significantly affected by changes in illumination, like on montage2A. It happens that these changes generate displacements of class members in the feature space but without extensively invading other class regions. In that situations, the application of the model-based approach, although it does provide improvements, their magnitude (see reduction factor on Table 7.2) is lower, not because of the method itself but because of the nature of the classification task.

The presented results show a considerably improved classification, hitherto robust to changes in the tilt of the illumination vector. In next section, the classification performance will be assessed with respect to the other variable, the illuminant slant angle.

7.3.2 Variation of slant angle

As done in Section 7.3.1, the three classification approaches considered in this chapter are put to the test, but now with respect to variation in the illuminant slant angle, holding the tilt of the illuminant constant throughout the experiment. We remind that—as demonstrated in Section 6.3.3—the application of the simple photometric stereo technique is by definition not recommended beyond the range 30°–75°, and therefore only the slant response inside this range is of interest here.

Experiment 7.2

Two composite images were used in this experiment, namely montage4C, and montage4D. Each classifier was trained at illuminant tilt angle of 0°, either using actual images (for the naive and best case method) or predicted images (for the model-based method). After training, the classifiers were tested at slants of 30°, 45°, 60°, and 75°. The classification results are shown in Figure 7.5. These two plots compare the slant response of the naive, model-based, and best case classifiers under an optimistic situation, since a training image from tilt of 0° is used. Presumably, in a general study of variation in both illuminant angles, the slant-induced TEC would be added to the tilt-induced TEC, reflecting a more realistic figure.

Although being the improvements also noticeable, the results are not as significant as in the tilt response; the most important TEC reduction is found on montage4C at slant of 75° (see corresponding model-based and naive classified images in Figure 7.6). Specially unexpected is the fact that in montage4D, Figure 7.5.b, the error is lower at slant angle of 30° than at 45°, being the latter the training angle.

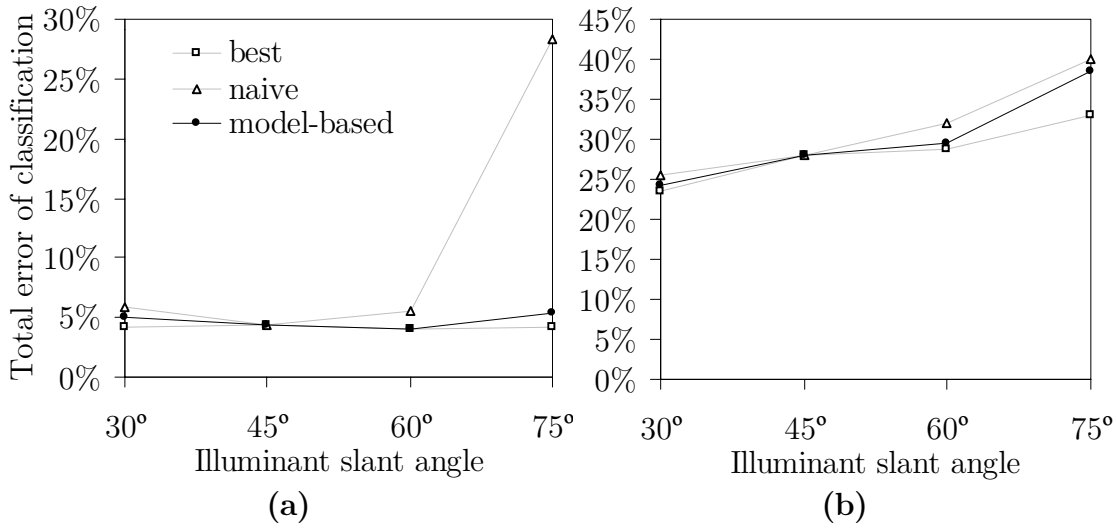
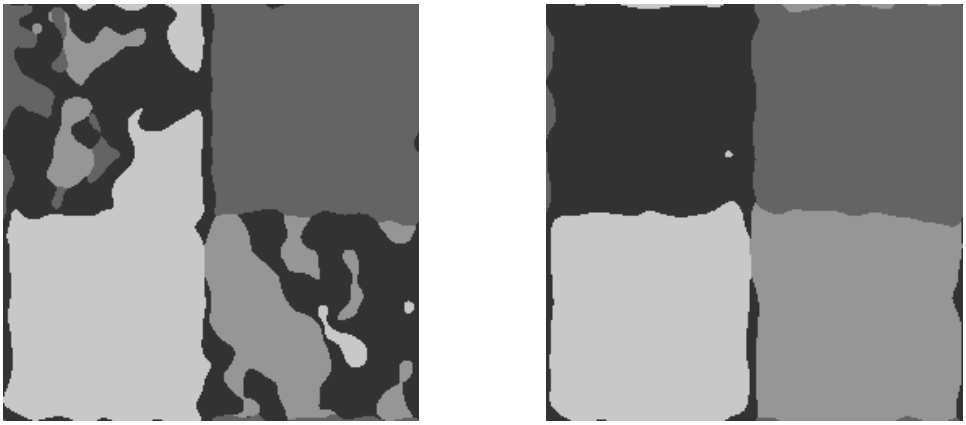


Figure 7.5 The effect of illuminant slant variation on **(a)** montage4C, and **(b)** montage4D. The legend is the same in the two graphs.



(a) **(b)**

Figure 7.6 Classification results at slant angle of 75° for **(a)** the naive and **(b)** the model-based approach (test image: montage4C).

7.4 Dependence on image prediction

It was observed in Experiment 7.1 that the tilt responses of the model-based and best case classification had a similar waveform. Next experiment investigates the implications that these similarities in waveform may have.

Experiment 7.3

This investigation into the differences between misclassifications rates of the model-based and best case classification aim to discern whether they can be utterly ascribed to image prediction inaccuracies. A new error rate, *absolute difference* classification error, was then defined as the simple arithmetic difference (taking the absolute value if negative) between both misclassification rates. Figure 7.7 shows this error rate on montages5E and montages5F. If the proposition was to be true, the curves of the absolute difference error should recall those of the $MSE(\tau)$ presented in Chapter 6. Although some resemblances may be establish, except that both of them reach minimums at tilts of 0°, 90°, and 180°, there is no evident relationship between them.

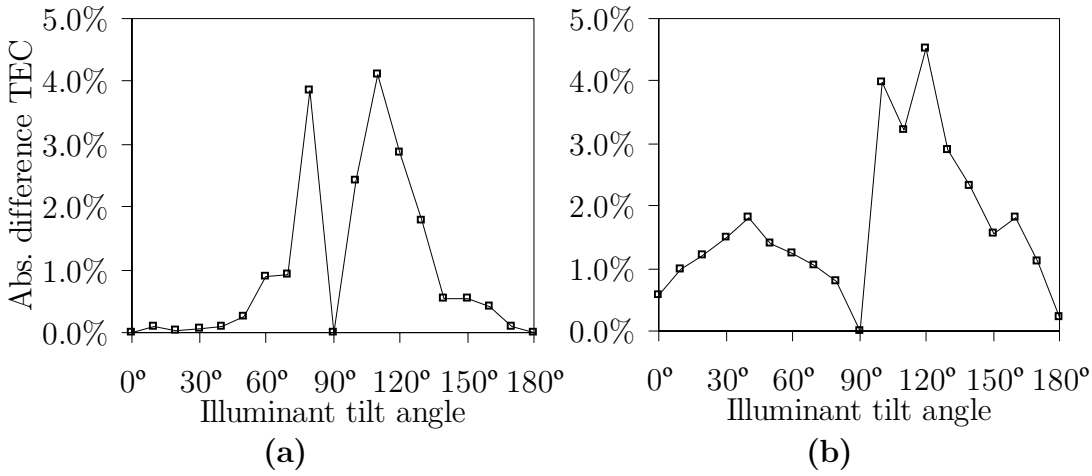


Figure 7.7 Absolute difference misclassification rate between the model-based and best case on **(a)** montage5E and **(b)** montage5F.

In fact, both measures are related by a non-linear relationship which is very complicated to experimentally establish. The main reason is found observing the feature space. Chantler et al [Chantler02] presented both theory and experimental results that show that the means of texture features derived from linear filters are sinusoidal functions of the illumination tilt in a 1D feature space, and in general follow super-elliptical trajectories in multidimensional feature spaces. It happens that the means of textures features obtained from photometric data tracks the elliptical trajectories of the means of the actual texture features but with a certain deviation. If a deviated value cross discrimination hypersurfaces, then they are significant; otherwise they are not reflected in the misclassification computation. It is concluded that the differences illustrated in Figure 7.7 are triggered by imprecisions in the image prediction, but their consequences are not proportionally observed from the misclassification point of view.

7.5 The effect of increasing number of textures

In this section the effect of increasing number of textures within a composite test image is studied. Initially, it is reasonable to suppose that the quality of the model-based classification will be degraded as more different textures are considered in the same scene. It is also expected that not only the model-based approach but also the naive and best case classifiers will perform poorly.

With the intention to ascertain whether the model-based approach maintains a good level of reduction despite increasing number of textures, let us define first two new parameters which characterise together with the reduction factor a specific classification task. Likewise the reduction factor, they are defined in terms of average TEC. The first one, called *efficiency*, represents how much the model-based method approximates to the best case classification, and takes values in the range [0,1], see Eq.(7.2); the second parameter, called *sensitivity*, assumes the best case classification as the starting point and measures how much the classification task is affected by changes in the tilt angle from that point of view, see Eq.(7.3). It can take values in the range $[1,\infty)$; a sensitivity equal to 1 means that the classification task is not susceptible to illuminant variation at all, thus it does not require compensation; whereas an infinite sensitivity represents the hypothetically most extreme classification task where the best case classification contains no errors and the naive classifier completely fails.

$$\text{efficiency } (E) = \frac{ABE}{AME} \tag{7.2}$$

$$\text{sensitivity } (S) = \frac{ANE}{ABE} \tag{7.3}$$

where

ABE is the Average Best case (total) Error of the classification task,

AME is the Average Model-based (total) Error of the classification task, and

ANE is the Average Naive (total) Error of the classification task.

It can be demonstrated that the efficiency and the sensitivity are related with the reduction factor R defined in Section 7.3.1 by the following equation:

$$R = 1 - \frac{1}{E \cdot S} \tag{7.4}$$

Figure 7.8 plots the reduction factor as a function of sensitivity considering different values of efficiency. Two are the main conclusions that can be drawn regarding the below figure:

1. R can be increased by either higher S or higher E; whereas the latter is also dependent on the model-based ability to compensate the tilt-induced classification error, S depends solely on the particular classification task, i.e. on the particular test image in the context of this investigation.

2. The less the efficiency, the higher the sensitivity required. For instance, a minimum value of $S = 2$ is required for a model-based approach, performing with an efficiency $E = 0.50$, in order to start being profitable, i.e. to attain $R > 0$.

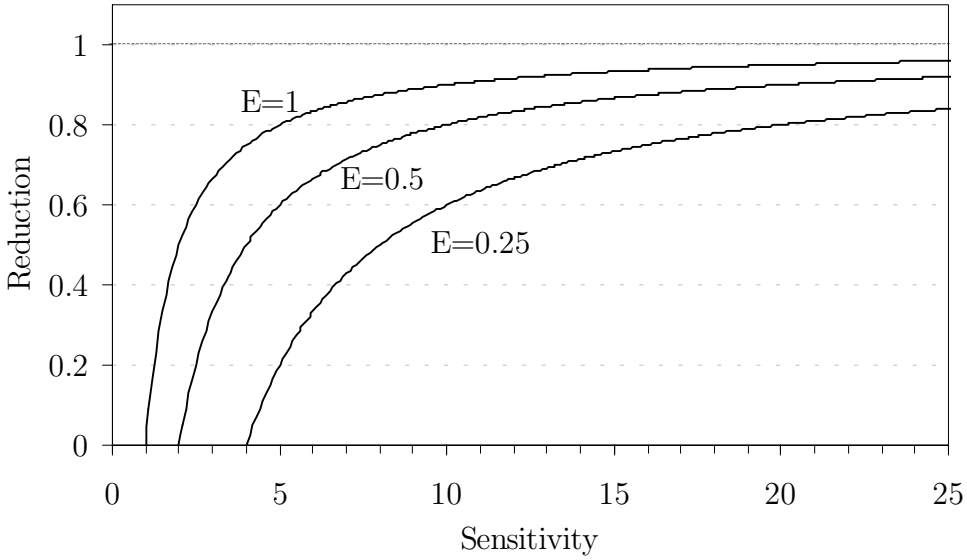


Figure 7.8 Reduction factor against sensitivity for different values of efficiency (E) of a particular classification task. E=0 is not plotted since it would imply a reduction tending to $-\infty$, which means that no improvement can be reached at any rate.

The set of parameters R, E, and S are a useful tool to study the effect of increasing the number of textures on classification. Note, however, the difficulty in analysing such a problem, mainly because of the impossibility of assessing the issue for the same classification task. Thus, it is possible, even though unusual, that a test image containing more textures performs better than other with less textures, because of the textures being easier to discriminate in the former than in the latter.

Experiment 7.4

This experiment reuses the results obtained for montage2A and montage4B, and introduces new data from montage5E, montage8G, and montage12H. Thereby, the effect of increasing the number of textures is assessed for 2-, 4-, 5-, 8-, and 12-textured images. Table 7.3 contained the average TEC for each classification approach on each montage into consideration. *ABE*, *AME*, *ANE* were calculated averaging the *TEC* along the tilt axis, thus indirectly this experiment also analyses the tilt response of new montages apart from the ones already studied in Experiment 7.1.

Average error	**Test image**				
	montage2A	montage4B	montage5E	montage8G	montage12H
ABE	2.34%	4.28%	5.96%	16.6%	24.1%
AME	2.36%	4.46%	6.96%	19.3%	30.0%
ANE	2.65%	13.3%	13.5%	30.9%	37.4%
Reduction	0.109	0.665	0.486	0.377	0.200
Efficiency	0.992	0.960	0.856	0.863	0.805
Sensitivity	1.132	3.114	2.274	1.860	1.553

Table 7.3 Average total error of classification for best case (*ABE*), model-based (*AME*), and naive classifiers (*ANE*) (test images: montage2A, montage4B, montage5E, montage8G, and montage12H). Also shown the associated reduction (R), efficiency (E), and sensitivity (S).

The results showed that the model-based method sustained a notable efficiency despite the rise in the number of textures from 2 to 12, with figures always superior to 0.8. However, the reduction factor R decreased dramatically, on montage2A and montage12H. Not being the efficiency parameter the main reason that such decrease occurred, it had to be caused by a decrement in the sensitivity S. On montage2A the reduction factor is close to 1, which means that the classification task is practically not affected by changes in the illumination. On test images from montage4B to montage12H the rate of increase in ABE was greater than that of the ANE. Certainly, about a tenfold increase in ABE was experienced, whereas only a fivefold increase in ANE. This unfortunate outcome can not be dealt with a better model-based approach, since it is chiefly dependent on the sensitivity, and therefore on the naive and best case performances. One could suggest then that the solution is to develop both better feature extraction system and better discriminant, but this is far beyond the scope of this research.

Chapter 8

Summary, conclusions, and future work

8.1 Summary

The topic of this dissertation is the classification of textured, rough surfaces under varying illuminant. Previous work has shown that variations in illumination direction causes misclassification in systems which are based on unidirectional or multidirectional feature measures such as popular Laws' masks or Gabor filters. This work has presented a simple shape from photometric stereo technique which, integrated into a conventional classification system, helps to decrease the classification errors.

Chapter 2 started with the first element of a conventional texture classification system, the image acquisition process. This process was divided into two parts: characterisation of incident image on the camera, and the so-called data image, which is a degradation of the incident image because of the imaging process. The incident image was modelled as a Lambertian image, based on Kube and Pentland's model of topological texture generalised to non-fractal surfaces [Kube88]. This theory provided a useful tool to the development of the subsequent approach to the problem of lighting dependence. Furthermore, this chapter also introduced some concepts about surface description, and concentrated on a first order estimator, the absolute average slope ratio or simply AASR, and a histogram representation for the

characterisation of surface gradient. The AASR parameter was used in next chapters to describe the roughness of all the textures employed in the experiments. The histograms were used to model the probability density functions of the surface partial derivatives p and q, and it provided a superior level of description, where other surface properties, e.g. directionality, were also identified.

Chapter 3 continued with the other two indispensable components of texture classification: feature generation and discriminant function. The course of discussion began by introducing the fundamentals of texture classification. A supervised statistical method was identified as suitable for our purposes. Afterwards, a survey of different feature measures searched for an approach such that its output had to be sensitive to illumination variation. Popularity in the literature and studies of mammalian visual cortex pointed to the use of a multichannel filtering approach based on a bank of 2D Gabor functions. Afterwards, the discriminant theory was revised and a linear Bayes classifier selected, basically for its simplicity and availability for implementation. In this chapter, no effort was destined to the design of a high-efficient classification system, since the interest was always directed to the reduction of misclassification on existing classification systems.

Chapter 4 provided the theoretical background necessary to develop the new component of the classification system, the recovery stage. It surveyed computer vision methods, commonly known as 'Shape from X'. Among the large number of possibilities, shape from photometric stereo was selected for further investigation, since it was reported as an improved method of traditional single image shape from shading techniques, and no additional hardware was required for its implementation; this method together with a relighting algorithm would produce 2D images of a surface under novel illuminant direction and form the foundations for training the classifier.

Chapter 5 explicitly faced the problematic of the dissertation. All the reviewed theory was linked and used to design a self-sufficient solution: *a*

model-based classification system. Detailed implementation of all the components was given; first, a simple photometric stereo scheme proposed by [McGunnigle98] was adopted to implement the surface recovery stage. This stage produced a model of the surface, consisting of partial derivative fields and albedo texture. When required, the model fed into a Lambertian relighting algorithm could produce an imitation of the perceived image of a texture under arbitrary illuminant tilt angle and slant angle. The filtering feature extraction process was designed using a set of 12 Gabor filters (with 4 equidistant orientations and 3 radial frequencies) based on [Jain91]. The space/spatial-frequency optimality of the Gabor function allow the opportunity to perform a frequency domain-based filtering, thus avoiding the burden of 2D convolutions; instead FFT algorithms were applied.

Chapter 6 evaluated the image prediction process as an individual component. The technique's robustness and reliability were put to the test with respect to variation in both the slant and the tilt of the illumination vector. Throughout the investigation the effect of violating the assumptions made by photometric stereo was intentionally assessed. Particularly, shadowing was shown as the most significant source of errors. Accordingly, an increase in the illuminant slant and the surface roughness were noted as indirect sources of errors, since they made shadowing more severe. Concerning the tilt response, an expression for the power of the error in the prediction, valid for isotropic textures, was presented. Concerning the slant response, the selection of a optimum slant angle for the photometric dataset was theoretically demonstrated to be a trade-off between maximising information and avoiding shadowing. Experimental work was then carried out, and the slant range 30°-45° was found as most suitable for the construction of the surface model.

Chapter 7 assessed the performance of the model-based classification system related to two extremes situations: a naive classifier and a 'best case' classifier. The investigation took into account variation in tilt and slant angles, as well as variation in the number of textures. Concepts of efficiency, reduction, and

sensitivity of a particular classification task were defined in the context of the three considered approaches. The results shown a considerable reduction of classification errors induced by changes in illuminant tilt angle. The investigation into the slant response was, in terms of experimental work, more moderate, but the results were also favourable.

8.2 Conclusions

This dissertation presented the concept of model-based texture classification which incorporated an image prediction subsystem. An increase in surface roughness was demonstrated to have a negative effect on image prediction. When the textured surfaces were made of the same material, a higher roughness typically implied worse prediction; however, an application including a variety of textures broke down the anterior relationship, and many other surface properties had to be considered. For isotropic textures, the power of the prediction error was found to be well modelled by a quadratic function of the illuminant tilt angle, which traced inverted parabolas between tilts of the photometric image set. Analysis of the results for the slant response showed that optimum limits of operation of the photometric stereo scheme may be approximately established between input slant angles of 30° and 45°.

Concerning the classification performance, the model-based classifier provided a considerable reduction in the number of tilt-induced classification errors. An increase in the number of textures to classify did not significantly affect the efficiency of the method, and the final reduction factor was mainly dependent on the classification task being sensitive to changes in the illumination. Despite the simplicity of the simple photometric stereo scheme, experimental simulations showed an enhanced, robust system to changes in lighting conditions, and a potential solution for future applications. In the author's opinion, the results have been encouraging enough to merit further investigation.

8.3 Future work

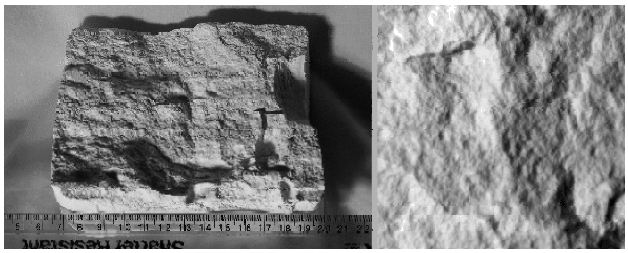
The robustness of the model-based approach to changes in slant angle has been evaluated in the range 30°–75°, and only four equidistant samples have been used. It is recommended a more thorough analysis, ideally from 10° to 80° in 10° increments, in order to ascertain the exact limitations of the proposed solution under varying slant angle. Further studies will also help to determine a definite slant angle of the photometric image set for optimal prediction.

Having observed the potential of the simple photometric stereo scheme, it is advocated the incorporation of a more sophisticated technique. For instance, a general photometric technique due to [Barsky01] is suggested. This approach uses three images, and their corresponding illuminant direction can be defined at any point, without predetermined values of tilt or slant angle. Assessment of the new scheme by comparison with the simple photometric scheme may help to discern whether the increased computational cost is balanced against the introduced benefits.

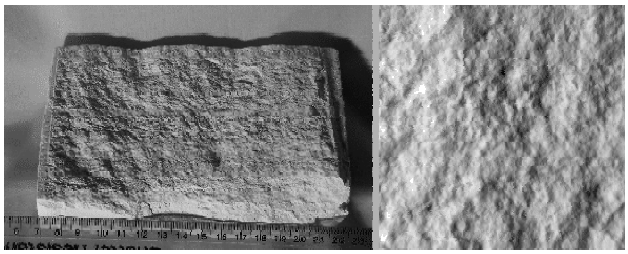
The model-based classification system presented in this dissertation is confined to applications where the illuminant direction can be either manually or automatically controlled. In order to extent the proposal to applications under varying and unknown illumination, investigation into recent photometric techniques that work under these circumstances is advised.

Appendix A: Test textures

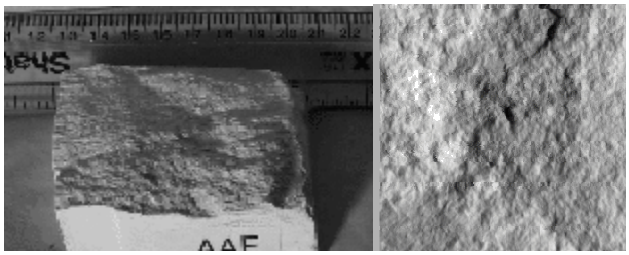
Experiment 6.1



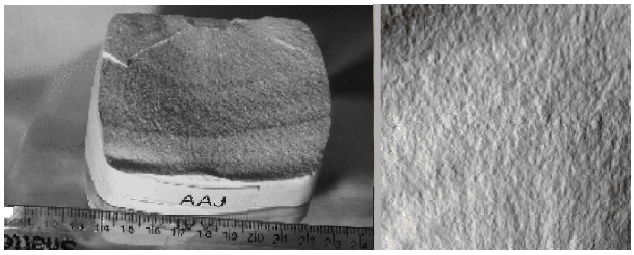
plaster1



plaster2

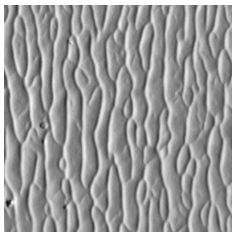


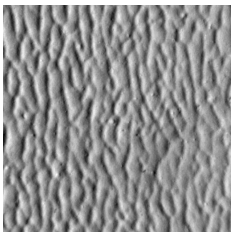
plaster3



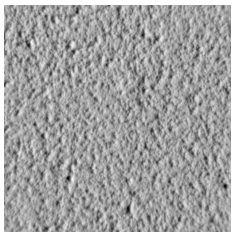
plaster4

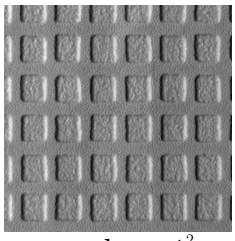
Experiment 6.2[1]

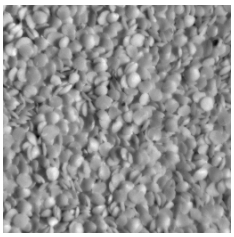

ripple1


ripple2

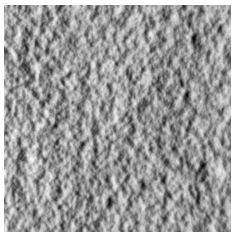

deposit1

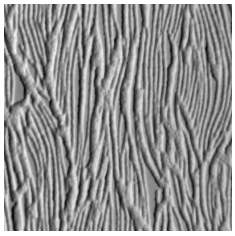

deposit2


anaglypta1[2]

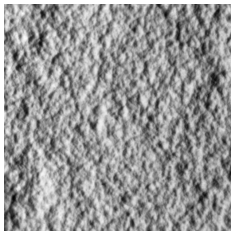

lentils

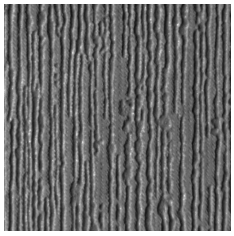

deposit5

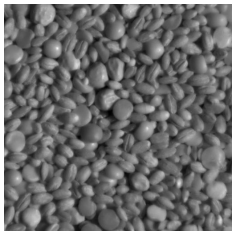

deposit3

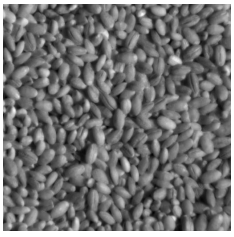

anaglypta2


deposit6


deposit4


anaglypta3


mixture


barley


plaster1

[1] Images are sorted in order of AASR; thus, for example, ripple1 matches the first point in Figure 6.3 around 0.05, while plaster1 corresponds to the last point around 0.18.
[2] Anaglypta™ is a type of thick embossed wallpaper, designed to be painted over.

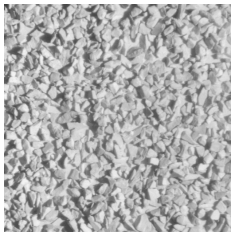
Experiment 6.3

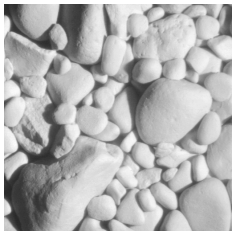
Isotropic Case

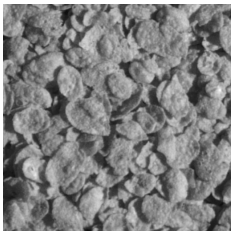

rock

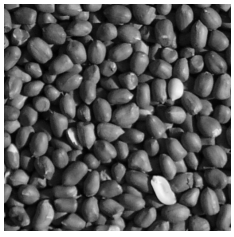

rice

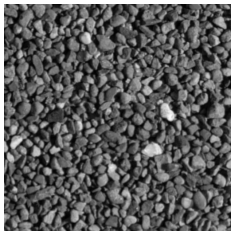

beans


chips

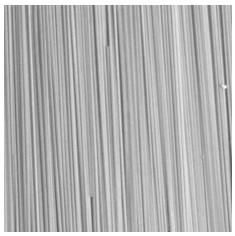

stones

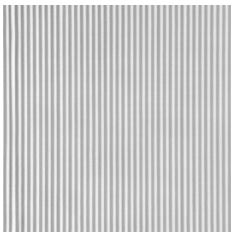

cornflakes

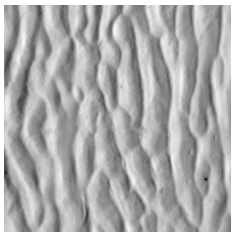

peanuts

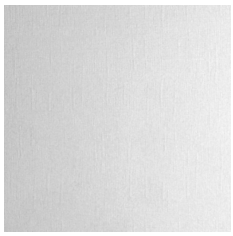

gravel

Anisotropic Case

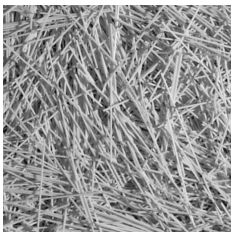

spaghetti1


card

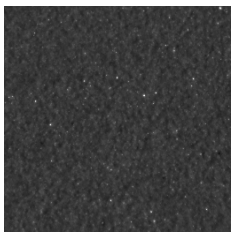

ripple3


paper


stucco

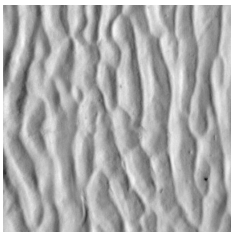

spaghetti2

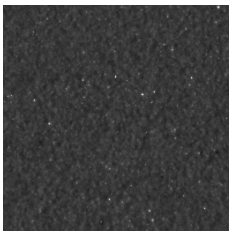
Specular case


sandpaper

Experiment 6.4


plaster2

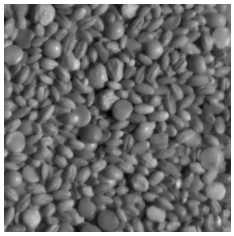

ripple3

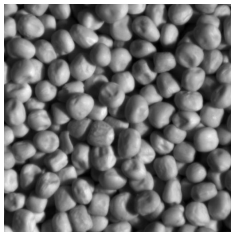

sandpaper

Experiment 6.5

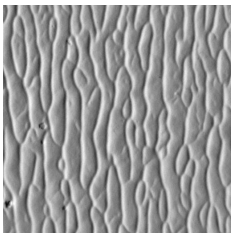

plaster4

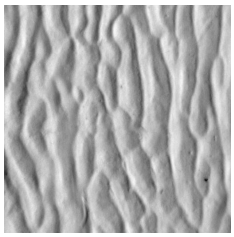

deposit1


mixture


peas


pebbles


ripple1


ripple3

Experiment 6.6

This experiment does not introduce any new texture. A sample image of the textured surfaces used can be consulted in previous sections of this appendix.

- isotropic textures: deposit1, deposit2, deposit3, deposit4, deposit5, deposit6, plaster2, and plaster4.
- directional textures: anaglypta1, anaglypta2, anaglypta3, ripple1, ripple2, and ripple3.
- repeating primitives textures: mixture, peas, and pebbles.

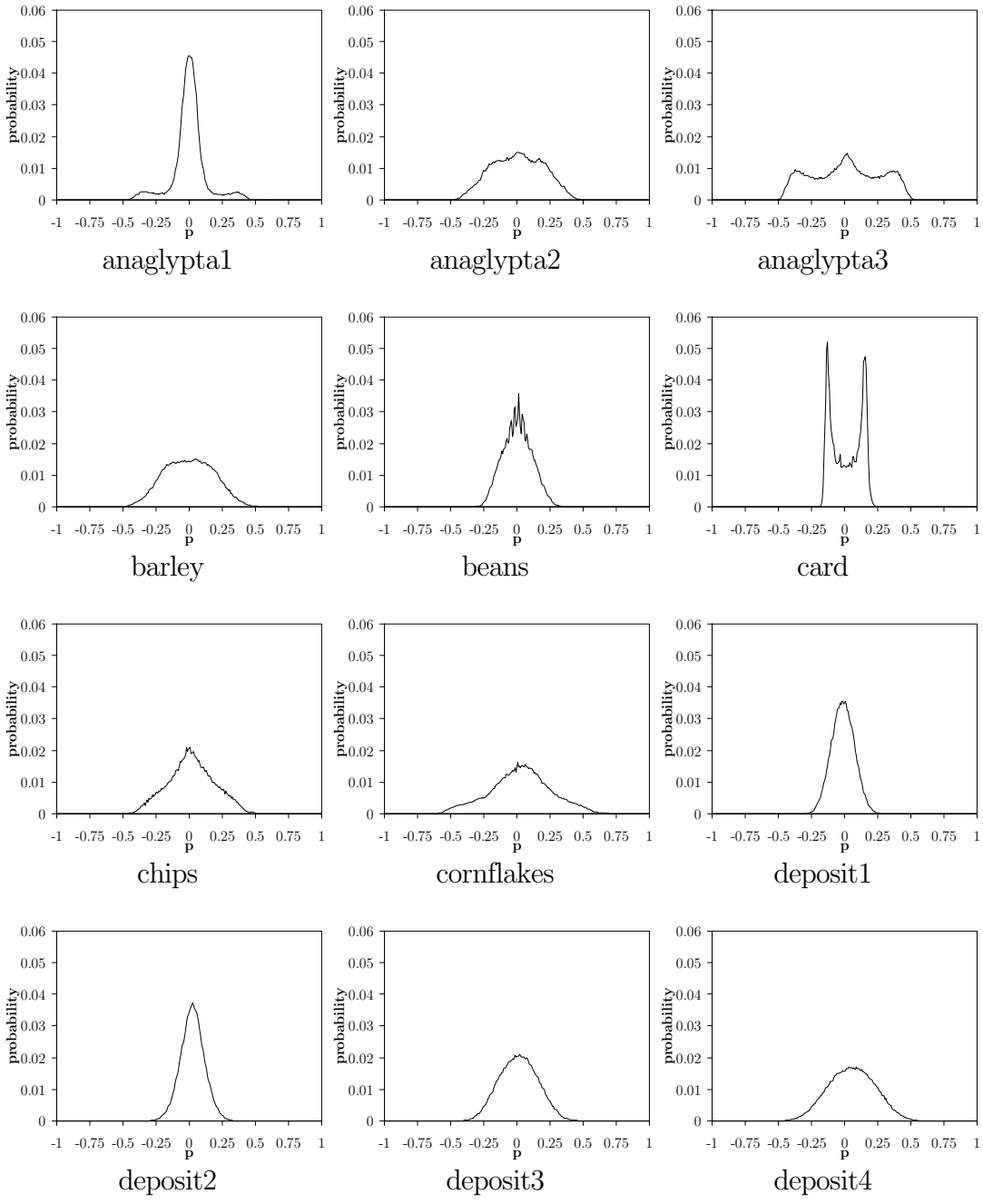
Appendix B: Histogram description of surface gradient

This appendix contains the estimated probability density functions (PDF) for the surface partial derivatives p and q of all 35 textures used in this investigation. The 512×512 p-map and q-map of the surface textures were estimated using the simple photometric stereo scheme developed in Section 5.3.1—Eq.(5.8) and Eq.(5.9) respectively. The histograms shown in Section B.1 and B.2 are 256-points discrete approximations of the actual PDFs in the range $[-1,1]$. These histograms represent a statistical model to surface description, and because there is a linear relationship between surface gradient and surface height, the characteristics observed here such as Gaussian distributions are also valid to describe the surface height map.

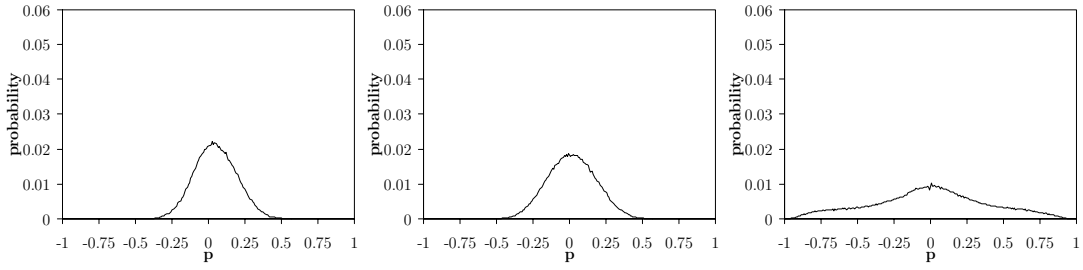
Table B.1, at the end of the appendix, lists the AASR values for all the textures, as well as the mean values of the surface slopes. Observe that both p and q mean always tends to zero — not strictly true for peanuts and gravel q map — which validates the assumption made in Section 1.2 that the surfaces considered in this dissertation are globally flat.

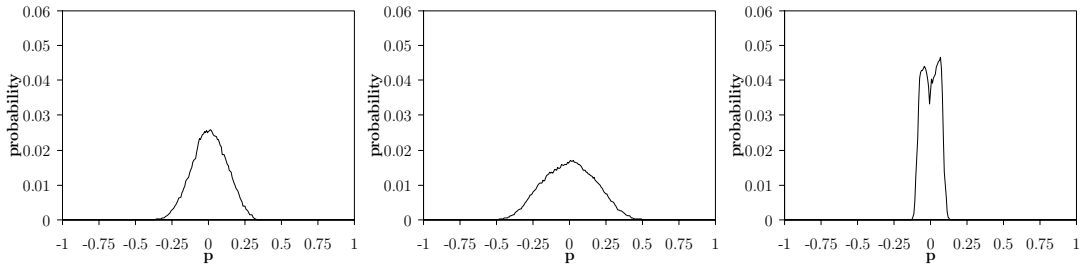
B.1 Estimated p-map probability density functions[1]

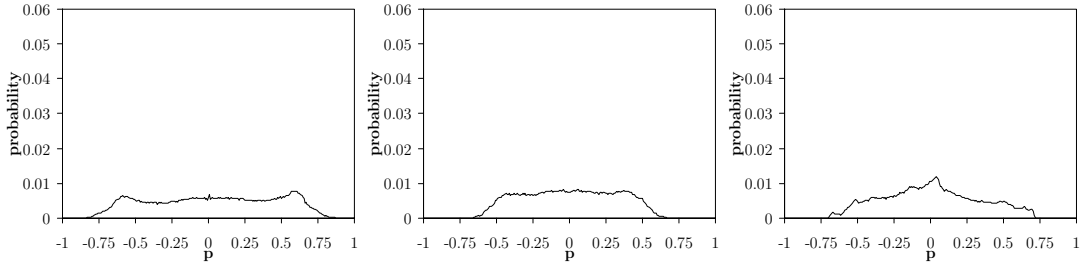
In alphabetical order:

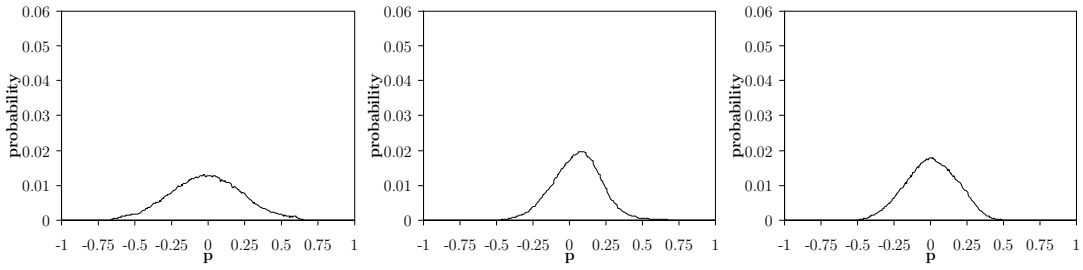


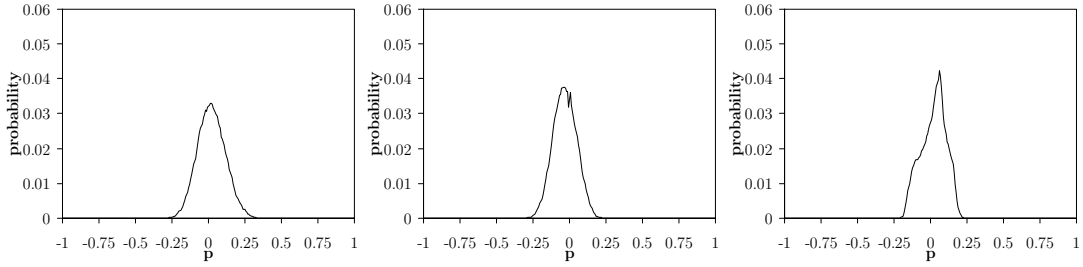
[1] Texture sandpaper has a maximum probability density of 0.0926. This values is out of plot, but it was preferred to maintain the same scale of y-axis so as to allow comparison between histograms.

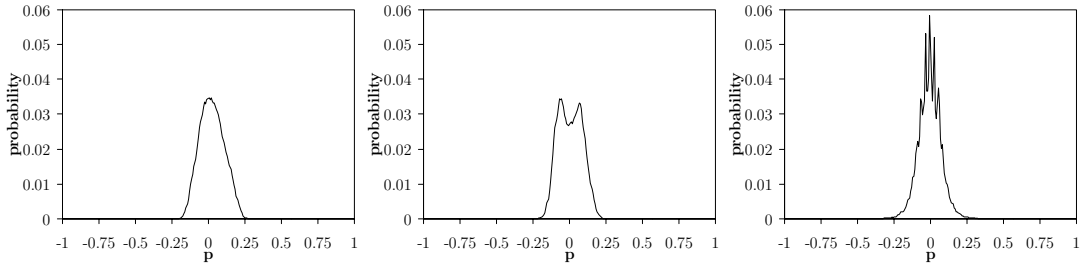

probability
p
deposit5
deposit6
gravel


probability
p
lentils
mixture
paper


probability
p
peanuts
peas
pebbles


probability
p
plaster1
plaster2
plaster3

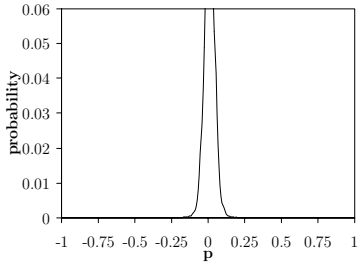

probability
p
plaster4
rice
ripple1



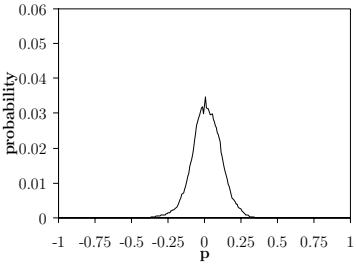
ripple2

ripple3

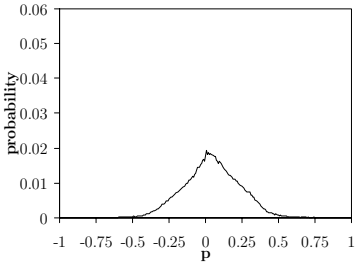
rock



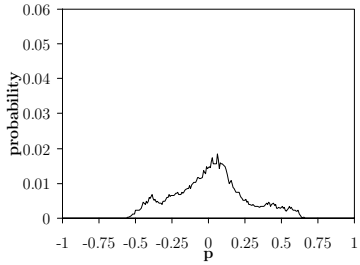
sandpaper



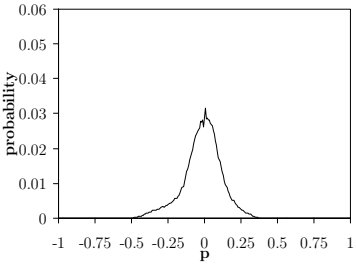
spaghetti1



spaghetti2



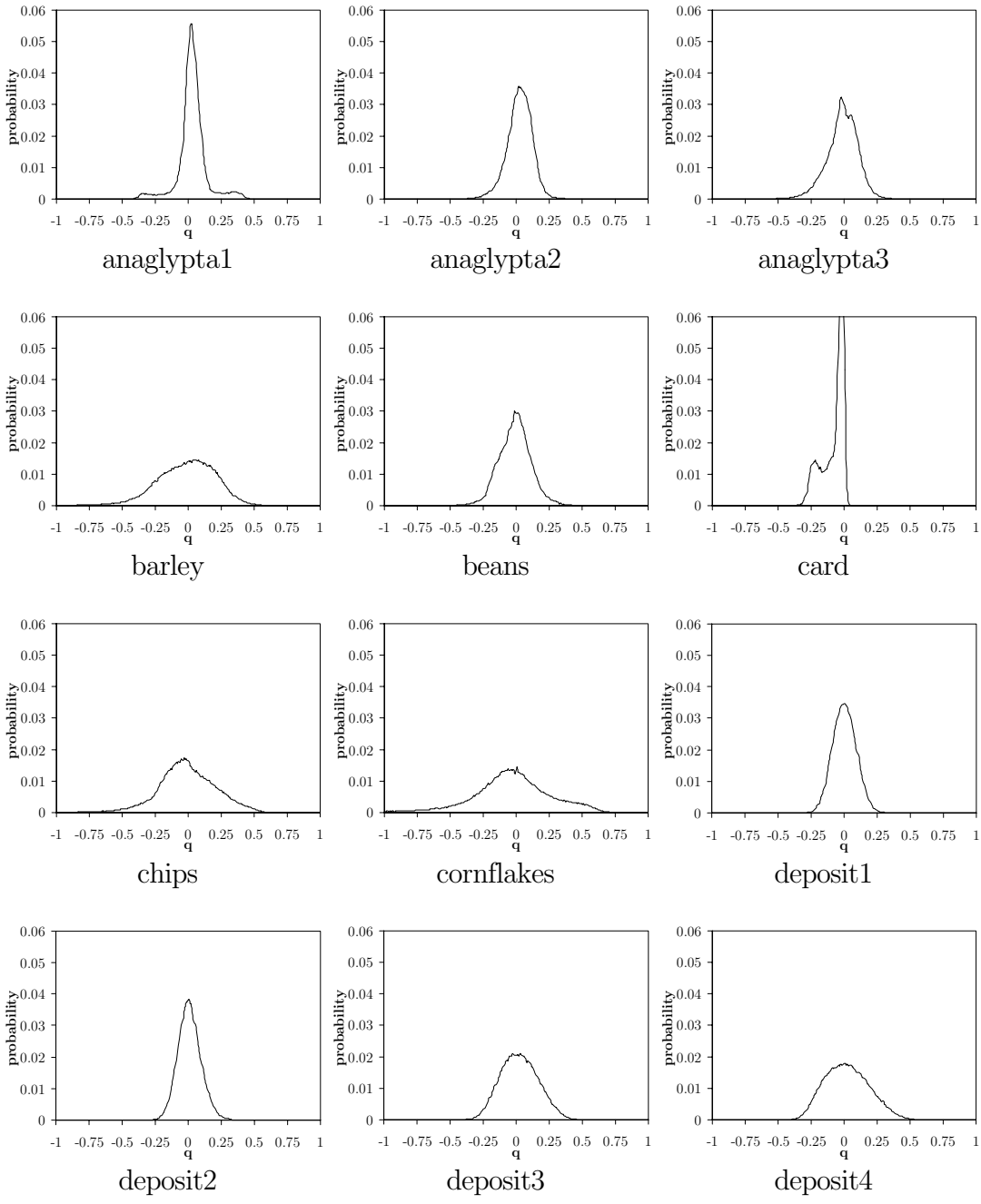
stones



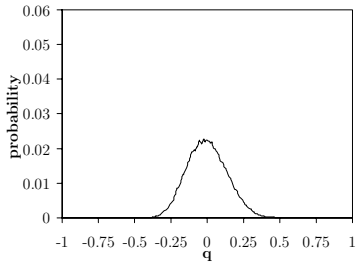
stucco

B.2 Estimated q-map probability density functions[2]

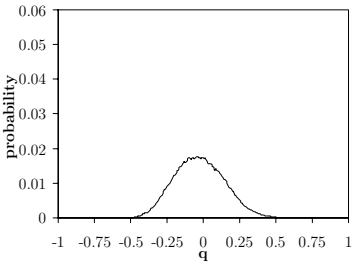
In alphabetical order:



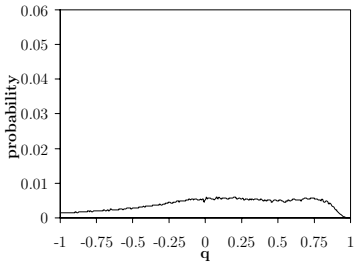
[2] Textures card, ripple1, ripple2, and ripple3 have a maximum probability density of 0.102, 0.123, 0.0798, and 0.101 correspondingly. These values are out of plot, but it was preferred to maintain the same scale of y-axis so as to allow comparison between histograms.



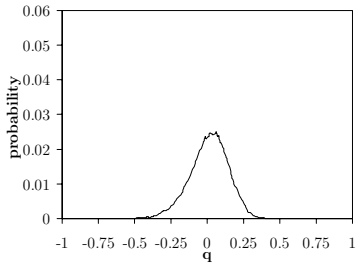
deposit5



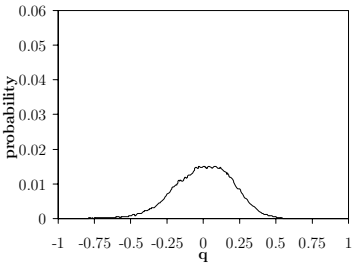
deposit6



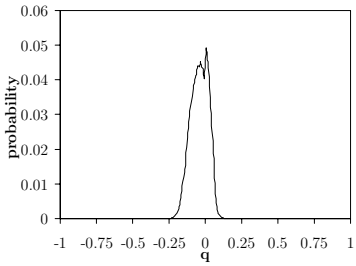
gravel



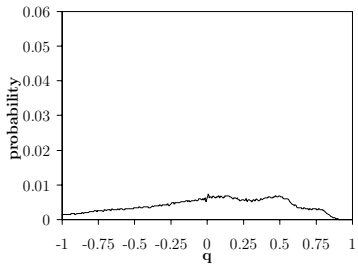
lentils



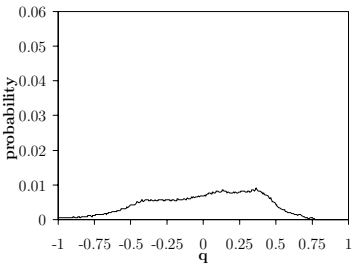
mixture



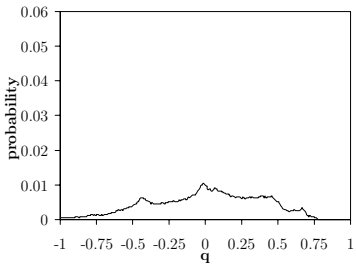
paper



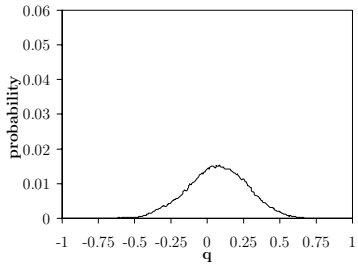
peanuts



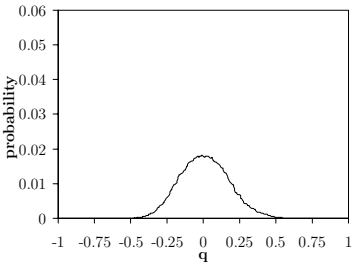
peas



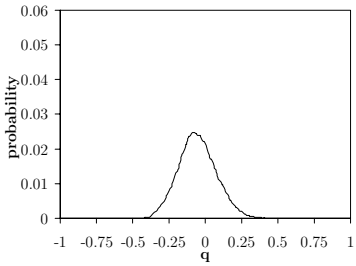
pebbles



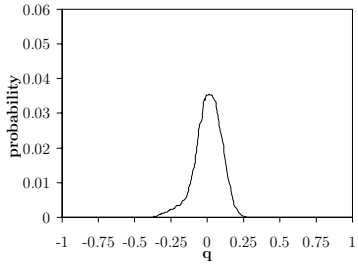
plaster1



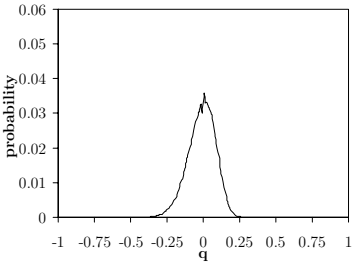
plaster2



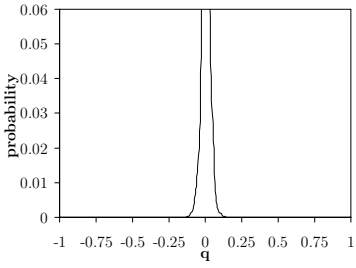
plaster3



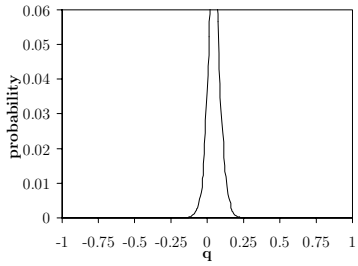
plaster4



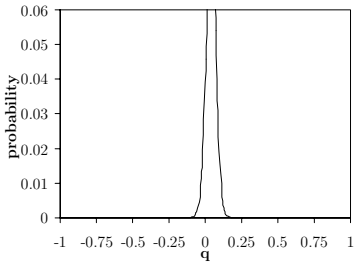
rice



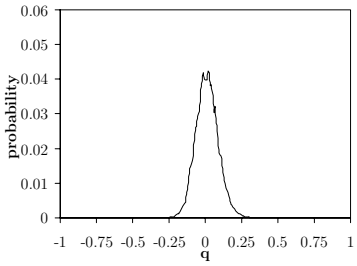
ripple1



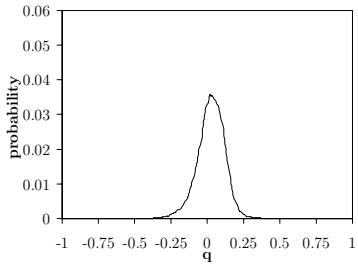
ripple2



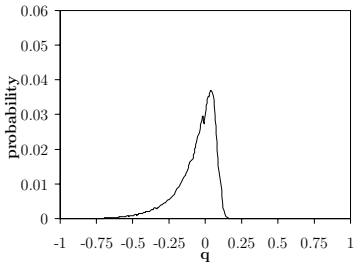
ripple3



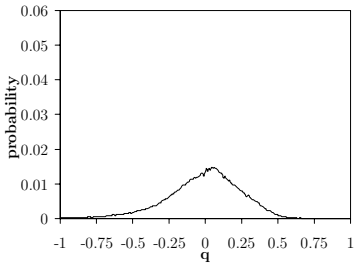
rock



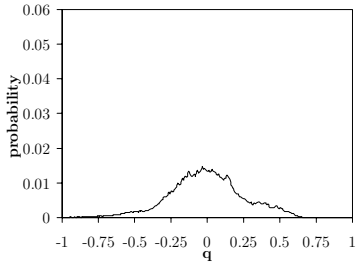
sandpaper



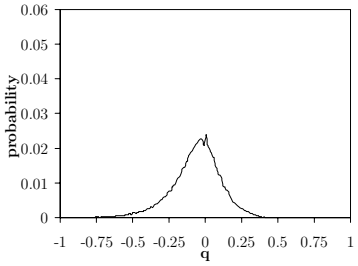
spaghetti1



spaghetti2



stones



stucco

B.3 First order statistics

Texture	AASR	P mean	Q mean
anaglypta1	0.07835	0.00056	0.03273
anaglypta2	0.11805	0.00637	0.02657
anaglypta3	0.15151	0.00950	–0.01386
barley	0.16267	0.00069	–0.00137
beans	0.09100	0.00462	–0.00898
card	0.09192	0.01457	–0.07522
chips	0.15430	0.01627	–0.00983
cornflakes	0.21397	0.02738	–0.06022
deposit1	0.07043	–0.01271	0.00327
deposit2	0.07148	0.02452	0.00967
deposit3	0.11586	0.01306	0.02450
deposit4	0.14120	0.05402	0.02321
deposit5	0.11502	0.04863	0.00016
deposit6	0.13599	0.01587	–0.02694
gravel	0.48159	–0.01076	–0.19207
lentils	0.10060	0.00788	0.01183
mixture	0.15331	–0.00089	–0.00112
paper	0.05258	0.00135	–0.03937
peanuts	0.47179	0.03619	–0.20904
peas	0.30300	0.00478	–0.04198
pebbles	0.30423	0.00536	–0.04619
plaster1	0.18494	–0.01185	0.06890
plaster2	0.13719	0.05316	0.00946
plaster3	0.12644	0.00593	–0.05353
plaster4	0.07573	0.02089	0.00640
rice	0.07278	–0.03170	–0.01116
ripple1	0.04840	0.02137	0.00440
ripple2	0.06193	0.02428	0.04736
ripple3	0.05690	0.00557	0.04082
rock	0.06062	–0.05202	0.01281
sandpaper	0.03786	0.00936	0.00097
spaghetti1	0.09029	0.01474	–0.05943
spaghetti2	0.17848	0.04192	–0.01312
stones	0.19778	0.01677	–0.00877
stucco	0.11598	–0.01392	–0.06286

Table B.1 Estimated AASR parameter, and mean value of the partial derivatives fields p and q for all the textures used in this dissertation (listed alphabetically).

Appendix C: Shell scripts

This appendix provides prototypical Unix shell scripts which, with little or no modification, may be used to reproduce all the experimental work contained in this dissertation. Implementation of all representative scripts is fully detailed, concerning the evaluation of image prediction accuracy and classification performance. Required C/C++ programs (source code and executable versions) can be found in `/u1/cs4/ceeag2/src` and `/u1/cs4/ceeag2/bin`.

C.1 Image prediction

This produces a photometric model of a given texture, uses the model to render images at tilt angles between 0° and 360° in specified increments, and calculates the MSE of the predictions by comparison with the actual image of the texture under the current tilt angle. The possibility of introducing different input and output slant angles is contemplated. The raw data is gathered into a `.xls` file for further processing and analysis. It can be used to reproduce Experiment 6.1, 6.2, 6.3, 6.4, 6.5, and 6.6.

```
# ====================================================================
# SHELL SCRIPT TO CALCULATE PHOTOMETRIC STEREO ERROR
# USAGE: pserror texture_name input_slant output_slant tilt_increm
# ====================================================================

# Definition of paths
set -x
programs=~/bin
data=/net/delos.macs.hw.ac.uk/spare/dbase_cooked/Investigation0
```

```
# ===== SIMPLE PHOTOMETRIC STEREO =====
# Select photometric image set
# tilt at 0,90,180 deg, slant at input_slant
surf0=$data/$1/1.$1.0.$2.0
surf90=$data/$1/1.$1.0.$2.90
surf180=$data/$1/1.$1.0.$2.180

# Estimate derivative fields and albedo using photometric stereo
$programs/photometric2 -s$2 $surf0 $surf90 $surf180 $1_p.tmp
  $1_q.tmp $1_alb.tmp

# ===== RELIGHTING ALGORITHM =====
# Initialisation of tilt angle
tilt=0

# Looping condition on tilt angle
while [ ${tilt} -le 359 ]
do

# Lambertian rendering at current tilt angle and output_slant
$programs/lamb -s$3 -t$tilt -o2 $1_p.tmp $1_q.tmp $1-s$3t$tilt.tmp2

# Compensating a constant of 240 introduced by Lambertian program
$programs/divim1 -v240 $1-s$3t$tilt.tmp2 $1-s$3t$tilt.tmp2
  $1-s$3t$tilt.tmp3

# Adding albedo
$programs/multimago $1-s$3t$tilt.tmp3 $1_alb.tmp $1-s$3t$tilt.o

# Remove temporal images
rm *.tmp2
rm *.tmp3

# ===== CALCULATE MEAN SQUARE ERROR (MSE) =====
# Selecting actual image
real=$data/$1/1.$1.0.$3.$tilt

# Writing related info into error file
echo
echo -n "tilt=" >> err-$1-s$2$3.xls
echo $tilt >> err-$1-s$2$3.xls

# Computing mean square error and writing into error file
$programs/mserror $real $1-s$3t$tilt.o >> err-$1-s$2$3.xls

# Remove rendered image (optional)
rm *.o

# Increment tilt
tilt=`expr ${tilt} + $4`
```

```
# Writing current tilt into error file
echo ${tilt}
done

# Remove both partial derivatives and albedo images
rm *.tmp
```

A more compact procedure to obtain data from multiple slant angles, like in the case of Experiment 6.6, is to construct an auxiliary shell script that methodically repeats the experiment by executing `pserror` under different combinations of input-output slant angles. For instance, consider the following implementation called `mseall`.

```
# ================================================================
# SHELL SCRIPT TO REPEAT PSERROR FOR MULTIPLE
# COMBINATIONS OF INPUT-OUTPUT SLANT ANGLES
# USAGE: sh mseall texture_name tilt_increment
# ================================================================

for input_slant in 30 45 60 75
do
   for output_slant in 30 45 60 75
   do
      sh pserror $1 $input_slant $output_slant $2
   done
done
```

C.2 Classification performance

This section list the shell scripts used to simulate the best case, naive and model-based classification on montage2A. In total, three files are generated, `efile-best.xls`, `efile-naive.xls`, and `efile-mbased.xls`, which contain the TEC at illuminant tilt angles from 0° to 360° in 10° increments for each approach respectively. Each shell script consists of three parts: (a) montage construction, (b) feature extraction, and (c) training/testing. Implementation of parts (a) and (c) depends on the approach, and they are fully detailed in each subsections. As the feature extraction is invariant it is only listed for the best case classification. Experiment 7.1, 7.2, 7.3, and 7.4 can be reproduced using these shell scripts.

Best case classification

```
# ====================================================================
# SHELL SCRIPT TO PERFORM BEST CASE CLASSIFICATION
# USAGE: shellbest
# ====================================================================

set -x
programs=/u1/cs4/ceeag2/bin
data=/net/delos.macs.hw.ac.uk/spare/dbase_cooked/Investigation0
outdir=~mjc/public/images

tilt=0
while [ ${tilt} -le 360 ]
do

# ===== PART A: MONTAGE CONSTRUCTION =====
t1=aab
t2=aba
surf1=$data/$t1/0.$t1.0.45.$tilt
surf2=$data/$t2/0.$t2.0.45.$tilt
classmap=~/images/classmaps/classmap2b.o
$programs/montage -C "$classmap" -c50 "$surf1" -c100 "$surf2" montage

# ===== PART B: FEATURE EXTRACTION =====
# Transform montage in to the frequency domain
$programs/fft3 montage $outdir/real $outdir/imag
$programs/swapshop $outdir/real $outdir/real $outdir/real
$programs/swapshop $outdir/imag $outdir/imag $outdir/imag

# Create even and odd Gabor filters (3 radial freq, 4 orientations)
$programs/gaborset1 -f25 -a0 -p1 -n512 $outdir/even1.gab
$programs/gaborset1 -f25 -a0 -p-1 -n512 $outdir/odd1.gab
$programs/gaborset1 -f25 -a45 -p1 -n512 $outdir/even2.gab
$programs/gaborset1 -f25 -a45 -p-1 -n512 $outdir/odd2.gab
$programs/gaborset1 -f25 -a90 -p1 -n512 $outdir/even3.gab
$programs/gaborset1 -f25 -a90 -p-1 -n512 $outdir/odd3.gab
$programs/gaborset1 -f25 -a135 -p1 -n512 $outdir/even4.gab
$programs/gaborset1 -f25 -a135 -p-1 -n512 $outdir/odd4.gab

$programs/gaborset2 -f50 -a0 -p1 -n512 $outdir/even5.gab
$programs/gaborset2 -f50 -a0 -p-1 -n512 $outdir/odd5.gab
$programs/gaborset2 -f50 -a45 -p1 -n512 $outdir/even6.gab
$programs/gaborset2 -f50 -a45 -p-1 -n512 $outdir/odd6.gab
$programs/gaborset2 -f50 -a90 -p1 -n512 $outdir/even7.gab
$programs/gaborset2 -f50 -a90 -p-1 -n512 $outdir/odd7.gab
$programs/gaborset2 -f50 -a135 -p1 -n512 $outdir/even8.gab
$programs/gaborset2 -f50 -a135 -p-1 -n512 $outdir/odd8.gab

$programs/gaborset3 -f100 -a0 -p1 -n512 $outdir/even9.gab
$programs/gaborset3 -f100 -a0 -p-1 -n512 $outdir/odd9.gab
$programs/gaborset3 -f100 -a45 -p1 -n512 $outdir/even10.gab
```

```
$programs/gaborset3 -f100 -a45 -p-1 -n512 $outdir/odd10.gab
$programs/gaborset3 -f100 -a90 -p1 -n512 $outdir/even11.gab
$programs/gaborset3 -f100 -a90 -p-1 -n512 $outdir/odd11.gab
$programs/gaborset3 -f100 -a135 -p1 -n512 $outdir/even12.gab
$programs/gaborset3 -f100 -a135 -p-1 -n512 $outdir/odd12.gab

# Perform Gabor filtering in the frequency domain
for case in 1 2 3 4 5 6 7 8 9 10 11 12
do
$programs/multimago $outdir/real $outdir/even$case.gab
  $outdir/real-even$case.tmp
$programs/multimago $outdir/real $outdir/odd$case.gab
  $outdir/real-odd$case.tmp
$programs/multimago $outdir/imag $outdir/even$case.gab
  $outdir/imag-even$case.tmp
$programs/multimago $outdir/imag $outdir/odd$case.gab
  $outdir/imag-odd$case.tmp
done

rm $outdir/*.gab
rm $outdir/real
rm $outdir/imag

for case in 1 2 3 4 5 6 7 8 9 10 11 12
do
$programs/swapshop $outdir/real-even$case.tmp $outdir/real-
  even$case.tmp $outdir/real-even$case.tmp
$programs/swapshop $outdir/real-odd$case.tmp $outdir/real-
  odd$case.tmp $outdir/real-odd$case.tmp
$programs/swapshop $outdir/imag-even$case.tmp $outdir/imag-
  even$case.tmp $outdir/imag-even$case.tmp
$programs/swapshop $outdir/imag-odd$case.tmp $outdir/imag-
  odd$case.tmp $outdir/imag-odd$case.tmp
done

# Apply IFFT
for case in 1 2 3 4 5 6 7 8 9 10 11 12
do
$programs/ffti $outdir/real-even$case.tmp $outdir/imag-even$case.tmp
  $outdir/real$case
$programs/multim1 -v-1 $outdir/real-odd$case.tmp $outdir/real-
  odd$case.tmp $outdir/real-odd$case.tmp
$programs/ffti $outdir/imag-odd$case.tmp $outdir/real-odd$case.tmp
  $outdir/imag$case
done

rm $outdir/*.tmp

# Convert real and image parts into magnitude
for case in 1 2 3 4 5 6 7 8 9 10 11 12
do
$programs/magn $outdir/real$case $outdir/imag$case
```

```
$outdir/magnitude$case $outdir/phase$case
rm $outdir/real$case
rm $outdir/imag$case
rm $outdir/phase$case
done

# Create Gaussian (smoothing) LPFs for each set of Gabor filters
$programs/gablowp3 -f0 -a0 -p1 -n512 $outdir/gaborlowpass1
$programs/gablowp5 -f0 -a0 -p1 -n512 $outdir/gaborlowpass2
$programs/gablowp8 -f0 -a0 -p1 -n512 $outdir/gaborlowpass3

# Apply LPFs
for case in 1 2 3 4
do
$programs/fft3 $outdir/magnitude$case $outdir/real$case
  $outdir/imag$case
$programs/swapshop $outdir/real$case $outdir/real$case
  $outdir/real$case
$programs/swapshop $outdir/imag$case $outdir/imag$case
  $outdir/imag$case

$programs/multimago $outdir/gaborlowpass1 $outdir/real$case
  $outdir/realfiltered$case
$programs/multimago $outdir/gaborlowpass1 $outdir/imag$case
  $outdir/imagfiltered$case
$programs/swapshop $outdir/realfiltered$case
  $outdir/realfiltered$case $outdir/realfiltered$case
$programs/swapshop $outdir/imagfiltered$case
  $outdir/imagfiltered$case $outdir/imagfiltered$case
rm $outdir/real$case
rm $outdir/imag$case
rm $outdir/magnitude$case
done

for case in 5 6 7 8
do
$programs/fft3 $outdir/magnitude$case $outdir/real$case
  $outdir/imag$case
$programs/swapshop $outdir/real$case $outdir/real$case
  $outdir/real$case
$programs/swapshop $outdir/imag$case $outdir/imag$case
  $outdir/imag$case

$programs/multimago $outdir/gaborlowpass2 $outdir/real$case
  $outdir/realfiltered$case
$programs/multimago $outdir/gaborlowpass2 $outdir/imag$case
  $outdir/imagfiltered$case
$programs/swapshop $outdir/realfiltered$case
  $outdir/realfiltered$case $outdir/realfiltered$case
$programs/swapshop $outdir/imagfiltered$case
  $outdir/imagfiltered$case $outdir/imagfiltered$case
rm $outdir/real$case
```

```
rm $outdir/imag$case
rm $outdir/magnitude$case
done

for case in 9 10 11 12
do
$programs/fft3 $outdir/magnitude$case $outdir/real$case
  $outdir/imag$case
$programs/swapshop $outdir/real$case $outdir/real$case
  $outdir/real$case
$programs/swapshop $outdir/imag$case $outdir/imag$case
  $outdir/imag$case

$programs/multimago $outdir/gaborlowpass3 $outdir/real$case
  $outdir/realfiltered$case
$programs/multimago $outdir/gaborlowpass3 $outdir/imag$case
  $outdir/imagfiltered$case
$programs/swapshop $outdir/realfiltered$case
  $outdir/realfiltered$case $outdir/realfiltered$case
$programs/swapshop $outdir/imagfiltered$case
  $outdir/imagfiltered$case $outdir/imagfiltered$case
rm $outdir/real$case
rm $outdir/imag$case
rm $outdir/magnitude$case
done

rm $outdir/gaborlowpass*

# Apply IFFT
for case in 1 2 3 4 5 6 7 8 9 10 11 12
do
$programs/ffti $outdir/realfiltered$case $outdir/imagfiltered$case
  if$case
rm $outdir/realfiltered$case
rm $outdir/imagfiltered$case
done

# ===== PART C: TRAINING/TESTING THE CLASSIFIER =====
echo "if1 \n if2 \n if3 \n if4 \n if5 \n if6 \n if7 \n if8 \n if9 \n
  if10 \n if11 \n if12" > FLIST
$programs/covar12 "$classmap" cov_stats 2 12 50 < FLIST
$programs/discrim12 -o50 -c2 -f12 -C cov_stats -D discriminant_file
  -F FLIST -O segout

# Calculate and write error into a file
echo
echo -n "tilt=" >> efile-best.xls
echo $tilt >> efile-best.xls
$programs/errcnt -c2 "$classmap" segout >> efile-best.xls
rm discriminant_file
rm if*
done
```

Naive classification

```
# ====================================================================
# SHELL SCRIPT TO PERFORM NAIVE CLASSIFICATION
# USAGE: shellnaive
# ====================================================================

set -x
programs=/u1/cs4/ceeag2/bin
data=/net/delos.macs.hw.ac.uk/spare/dbase_cooked/Investigation0
outdir=~mjc/public/images

tilt=0

# ===== PART A: MONTAGE CONSTRUCTION =====
# ===== PART B: FEATURE EXTRACTION =====
# Identical to best case implementation

# ===== PART C: TRAINING/TESTING THE CLASSIFIER =====
echo "if1 \n if2 \n if3 \n if4 \n if5 \n if6 \n if7 \n if8 \n if9 \n
  if10 \n if11 \n if12" > FLIST

# Training
$programs/covar12 "$classmap" cov_stats 2 12 50 < FLIST
# Generates the discriminant_file at tilt=0
$programs/discrim12 -o50 -c2 -f12 -C cov_stats -D discriminant_at0
  -F FLIST

tilt=10
while [ ${tilt} -le 360 ]
do

# ===== PART A: MONTAGE CONSTRUCTION =====
# ===== PART B: FEATURE EXTRACTION =====
# Identical to best case implementation

# ===== PART C: TRAINING/TESTING THE CLASSIFIER =====
# Testing (uses discriminant at tilt=0 regardless of the actual tilt)
$programs/discrim12 -o50 -c2 -f12 -d discriminant_at0 -F FLIST
  -O segout

# Calculate and write error into a file
echo
echo -n "tilt=" >> efile-naive.xls
echo $tilt >> efile-naive.xls
$programs/errcnt -c2 "$classmap" segout >> efile-naive.xls
rm montage
rm segout
rm if*
done

rm discriminant_at0
```

Model-based classification

This approach was implemented using three shell scripts, namely `recovery`, `training`, and `testing`, corresponding to the three stages defined in Chapter 5. In order to efficiently reproduce the experiment at different tilt angles, the three shell scripts were combined in a new shell script called `shellmbased`, which is listed at the end of the section.

```
# ====================================================================
# SHELL SCRIPT TO PERFORM THE RECOVERY STAGE
# USAGE: recovery texture_name input_slant
# ====================================================================

set -x
programs=~/bin
data=/net/delos.macs.hw.ac.uk/spare/dbase_cooked/Investigation0

# ===== SIMPLE PHOTOMETRIC STEREO =====
# Select photometric image set
# tilt at 0,90,180 deg, slant at input_slant
surf0=$data/$1/1.$1.0.$2.0
surf90=$data/$1/1.$1.0.$2.90
surf180=$data/$1/1.$1.0.$2.180

# Estimate derivative fields and albedo using photometric stereo
$programs/photometric2 -s$2 $surf0 $surf90 $surf180 $1_p.tmp
  $1_q.tmp $1_alb.tmp


# ====================================================================
# SHELL SCRIPT TO PERFORM THE TRAINING STAGE
# USAGE: training output_slant tilt_angle tex_name1 tex_name2
# ====================================================================

set -x
programs=~/bin
outdir=~mjc/public/images

# ===== RELIGHTING ALGORITHM =====
# Lambertian rendering at current tilt angle and output_slant
$programs/lamb -s$1 -t$2 -o2 $3_p.tmp $3_q.tmp $3-$1-$2.tmp2
$programs/lamb -s$1 -t$2 -o2 $4_p.tmp $4_q.tmp $4-$1-$2.tmp2

# Compensating a constant of 240 introduced by Lambertian program
$programs/divim1 -v240 $3-$1-$2.tmp2 $3-$1-$2.tmp2 $3-$1-$2.tmp3
$programs/divim1 -v240 $4-$1-$2.tmp2 $4-$1-$2.tmp2 $4-$1-$2.tmp3

rm *.tmp2
```

```
#Adding albedo
$programs/multimago $3-$1-$2.tmp3 $3_alb.tmp $3-$1-$2.tmp4
$programs/multimago $4-$1-$2.tmp3 $4_alb.tmp $4-$1-$2.tmp4

rm *.tmp3

# ===== PART A: MONTAGE CONSTRUCTION =====
classmap=~/images/classmaps/classmap2b.o
$programs/montage -C "$classmap" -c50 $3-$1-$2.tmp4 -c100
  $4-$1-$2.tmp4 montage

rm *.tmp4

# ===== PART B: FEATURE EXTRACTION =====
# Identical to best case implementation

# ===== PART C: TRAINING THE CLASSIFIER =====
$programs/covar12 "$classmap" cov_stats 2 12 50 < FLIST
$programs/discrim12 -o50 -c2 -f12 -C cov_stats -D discriminant_train
  -F FLIST

rm cov_stats
rm if*
rm montage


# ====================================================================
# SHELL SCRIPT TO PERFORM THE TESTING STAGE
# USAGE: testing output_slant tilt_angle tex_name1 tex_name2
# ====================================================================

set -x
programs=/u1/cs4/ceeag2/bin
data=/net/delos.macs.hw.ac.uk/spare/dbase_cooked/Investigation0
outdir=~mjc/public/images

surf1=$data/$3/0.$3.0.$1.$2
surf2=$data/$4/0.$4.0.$1.$2

# ===== PART A: MONTAGE CONSTRUCTION =====
# ===== PART B: FEATURE EXTRACTION =====
# Identical to best case implementation


# ===== PART C: TESTING THE CLASSIFIER =====
# Using discriminant from training (based on photometric data)
$programs/discrim12 -o50 -c2 -f12 -d discriminant_train -F FLIST
  -O segout

# Calculate and write error into a file
echo
echo -n "tilt=" >> efile-mbased.xls
```

```
echo $1 >> efile-mbased.xls
$programs/errcnt -c2 "$classmap" segout >> efile-mbased.xls

rm if*
rm segout
rm discriminant_file
rm montage


# ====================================================================
# SHELL SCRIPT TO PERFORM MODEL-BASED CLASSIFICATION USING
# RECOVERING + TRAINING + TESTING (LISTED ABOVE)
# USAGE: shellmbased input_slant output_slant tex_name1 tex_name2
# ====================================================================

set -x
tilt=0

sh recovery $3 $1
sh recovery $4 $1

while [ ${tilt} -le 360 ]
do

sh training $2 ${tilt} $3 $4
sh testing $2 ${tilt} $3 $4

tilt=`expr ${tilt} + 10`

done
```

References

Aach95 T. Aach, A. Kraup & R. Mester, "On texture analysis: Local energy versus quadrature filters", Signal Processing, vol.45, pp.173–181, 1995.

Ashley95 J. Ashley, M. Flickner, J. Hafner, D. Lee, W. Niblack & D. Petkovic, "The query by image content (qbic) system", Proc. of the ACM SIGMOD International Conference on Management Data, San Jose, CA, USA, 1995

Barsky01 S. Barsky & M. Petrou, "Colour photometric stereo: simultaneous reconstruction of local gradient and colour of rough textured surfaces", Proceedings Eighth IEEE International Conference on Computer Vision, ICCV 2001, vol.2, pp.600–605, Los Alamitos, CA, USA, 2001.

Bennett89 J.M. Bennett & L. Mattson, "Introduction to surface roughness and scattering", Optical Society of America, Washington D.C., pp.39–44, 1989.

Bovik90 A.C. Bovik, M. Clark & W.S. Geisler, "Multichannel texture analysis using localized spatial filters", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.12, no.1, pp.55–73, 1990.

Chantler02 M. Chantler, M. Schmidt, M. Petrou & G. McGunnigle, "The effect of illuminant rotation on texture filters: Lissajous's ellipses", European Conference on Computer Vision, Copenhagen, 2002

Chantler94 M.J. Chantler, "The effect of illuminant direction on texture classification", PhD. thesis, Dept. of Computing and Electrical Engineering, Heriot-Watt University, 1994.

Clark92 J.J. Clark, "Active photometric stereo", IEEE Computer Vision and Pattern Recognition, pp.29–34, 1992.

Clarke92 S.J. Clarke, "The analysis and synthesis of texture in sidescan sonar data", PhD. thesis, Dept. of Electrical and Electronic Engineering, Heriot-Watt University, 1992.

Coleman82 E.N. Coleman & R. Jain, "Obtaining 3-dimensional shape of textured and specular surfaces using four-source photometry", Computer Vision, Graphics, and Image Processing, vol.18, no.4, pp.309–328, 1982.

Couch93 L.W. Couch, "Digital and analog communication systems", Fourth edition, Macmillan, 1993.

Daugman85 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: Optics and Image Science, vol.2, no.7, pp.1160–1169, 1985.

Devijver86 P.A. Devijver & J. Kittler, "Pattern recognition theory and applications", Springer Verlag, Berlin-New York-Tokyo, 1986.

Dunn95 D. Dunn & W.E. Higgins, "Optimal Gabor filters for texture segmentation", IEEE Transactions on Image Processing, vol.4, no.7, pp.947–964, 1995.

Fukuda99 S. Fukuda & H. Hirosawa, "A wavelet-based texture feature set applied to classification of multifrequency polarimetric sar images", IEEE Trans. on Geoscience and Remote Sensing, vol.37, no.5, pp.2282–2286, 1999

Fukunaga90 K. Fukunaga, "Introduction to statistical pattern recognition", Academic Press, Boston, 2nd edition, 1990

Gabor46 D. Gabor, "Theory of communication", J. Inst. Electr. Eng., vol.93, pp.429–457, 1946

Healey88 G.E. Healey & T.O. Binford, "Local shape from specularity", Computer Vision, Graphics, and Image Processing, vol.42, pp.62–86, 1988.

Healey94 G.E. Healey & R. Kondepudy, "Radiometric CCD camera calibration and noise estimation", IEEE Transactions of Pattern Analysis and Machine Intelligence, vol.16, no.3, pp.267–276, 1994.

Horn75 B.K.P. Horn, "Shape from shading", McGraw Hill, New York, 1975.

Ikeuchi81 K. Iteuchi, "Determining surface orientations of specular surfaces by using photometric stereo method", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.3, no.6, pp.661–669, 1981.

Ikeuchi90 K. Ikeuchi, S.K. Nayar & T. Kanade, "Determining shape and reflectance of hybrid surfaces by photometric sampling", IEEE Trans. on Robotics and Automation, vol.6, pp.418–430, 1990.

Jain91 A.K. Jain & K. Farrokhnia, "Unsupervised texture segmentation using Gabor filters", Pattern Recognition, vol.24, no.12, pp.1167–1186, 1991.

James85 M. James, "Classification algorithms", Collins, 1985.

Kieran95 Kieran, "Implementation and design of discrete Gabor filter for sonar texture classification", PhD Thesis, Dept. of Computing and Electrical Engineering", Heriot-Watt University, 1995.

Kube88 P. Kube & A. Pentland, "On the imaging of fractal surfaces", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.10, no.5, pp.704–707, 1998.

Laws79 K.I. Laws, "Texture energy measures", Proceedings of Image Understanding Workshop, pp.47–51, 1979.

Lee92 K.M. Lee & C.C.J. Kuo, "Shape reconstruction from photometric stereo", Computer Vision and Pattern Recognition, pp.479–484, 1992.

Leung1999 T. Leung & J. Malik, "Recognizing surfaces using three-dimensional textons", IEEE International Conference on Computer Vision, pp.1010–1017, 1999.

Leung2001 T. Leung & J.Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textures", International Journal of Computer Vision, vol.43, no.1, pp.29–44, 2001.

Linnet91 L.M.Linnett, "Multi-texture image segmentation", PhD. thesis, Dept. of Computing and Electrical Engineering, Heriot-Watt University, 1991.

Malik90 J. Malik & P. Perona, "Preattentive texture discrimination with early vision mechanisms", Journal of the Optical Society of America A: Optics and Image Science, vol.7, pp.923–932, 1990

McGunnigle98 G. McGunnigle, "The classification of textured surfaces under varying illuminant direction", PhD. thesis, Dept. of Computing and Electrical Engineering, Heriot-Watt University, 1998.

Nayar90 S.K. Nayar, K. Ikeuchi & T. Kanade, "Shape from interreflections", Thrid Conference on Computer Vision, pp.1–11, 1990.

Nayar94 S.K. Nayar, Y. Nakagawa, "Shape from focus", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.16, no.8, pp.824–831, 1994.

Nicodemus77 F.E. Nicodemus, J.C. Richmond & J.J. Hsai, "Geometrical considerations and nomenclature for reflectance", U.S. Dept. of Commerce, National Bureau of Standards, 1977.

Pavel89 M. Pavel, "Fundamentals of pattern recognition", M. Dekker, New York, 1989

Pentland84 A. Pentland, "Local shading analysis", IEEE Transactions on PAMI, vol.6, pp.170–187, 1984

Pentland96 A. Pentland, R.W. Picard & S. Sclaroff, "Photobook: Content-based manipulation of image databases", Int. Journal of Computer Vision, vol.18, no.3, pp.233–254, 1996

Perry89 A. Perry & D.G. Lowe, "Segmentation of textured images", Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognition, pp.326–332, 1989

Randen98 T. Randen, B. Reymond, H.I. Sjulstad & Snneland, "New seismic attributes for automated stratigraphic facies boundary detection, SEG-98, New Orleans, LA, 1998

Randen99 T. Randen & J.H. Husoy, "Filtering for texture classification: A comparative study", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.21, no.4, pp.291–310, 1999.

Sanderson88 A.C. Sanderson, L.Weiss & S.K. Nayar, "Structured highlight inspection of specular surfaces", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.10, pp.44–55, 1988.

Shafer85 S.A. Shafer, "Using color to separate reflection components", Color Res. App., vol.10, no.4, pp.210–218, 1985.

Solomon92 F. Solomon & K. Ikeuchi, "Exctracting the shape and roughness of specular lobe objects using four light photometric stereo", Computer Vision and Pattern Recognition, pp.466–471, 1992.

Tagare90 H.D. Tagare & R.J.P. de Figueiredo, "Simultaneous estimation of shape and reflectance maps from photometric stereo", IEEE International Conference on Computer Vision, pp.340–343, 1990.

Tan90 T.N. Tan & A.G. Constantinides, "Texture analysis based on a human visual model", IEEE International Conference on Acoustics, Speech, and Signal Processing, pp.2091–2110, 1990

Tou74 J.T. Tou & R.C. Gonzalez, "Pattern recognition principles", Addison-Wesley, 1974.

Tuceryan93 M. Tuceryan & A.K. Jain, "Texture analysis", Handbook Pattern Recognition and Computer Vision, pp.235–276, Singapore: World Scientific, 1993

Unser90 M. Unser & M. Eden, "Nonlinear operators for improving texture segmentation based on features extracted by spatial filtering", IEEE Trans. System, Man, Cybernetics, vol.20, pp.804–815, 1990

VanGool85 L. Van Gool, P. Dewaele & A.Oosterlinck, "Texture analysis anno 1983", Computer Vision, Graphics and Image Processing, vol.29, pp.336-357, 1985.

Wang96 C.C. Wang & C.C. Chen, "Low-cost subpixel detection method for phase drift of camera/grabber signal conversion", Opt. Eng., vol.35, no.12, pp.3385–3391, 1996.

Wechsler80 H. Wechsler, "Texture analysis – a survey", Signal Processing 2, pp.271–282, 1980.

Weldon96 T.P. Weldon & W.E. Higgins, "Design of multiple Gabor filters for texture segmentation", IEEE International Conference on Acoustics, Speech, and Signal Processing, pp.2243–2246, 1996.

Woodham80 R.J. Woodham, "Photometric method for determining surface orientation from multiple images", Optical Engineering, vol.19, no.1, pp.139–144, 1980.

Wu99 D. Wu & J. Linders, "A new texture approach to discrimination of forest cleancut, canopy and burned area using airborne C-band SAR", IEEE Trans. on Geoscience and Remote Sensing, vol.37, no.1, pp.555–563, 1999