Classifying Surface Texture while Simultaneously Estimating Illumination Direction

M. Chantler M. Petrou A. Penirsche M. Schmidt
G. McGunnigle

December 9, 2003

Abstract

We propose a novel classifier that both classifies surface texture and simultaneously estimates the unknown illumination conditions. A new formal model of the dependency of texture features on lighting direction is developed which shows that their mean vectors are trigonometric functions of the illuminations' tilt and slant angles. This is used to develop a probabilistic description of feature behaviour which forms the basis of the new classifier. Given a feature set from an image of an unknown texture captured under unknown illumination conditions the algorithm first estimates the most likely illumination direction for each possible texture class. These estimates are used to calculate the class likelihoods and the classification is made accordingly.

The ability of the classifier to estimate illuminant direction, and to assign the correct class, was tested on 55 real texture samples in two stages. The classifier was able to accurately estimate both the tilt and the slant angles of the light source for the majority of textures and gave a 98% classification rate.

1 Introduction

This paper deals with the classification of surface textures on the basis of their image texture. One characteristic of surface textures is that the appearance of the surface is a function of the illuminant direction as well as of the surface topography, Figure 1, [Cha95] [DN99]. If the image is affected by the direction of lighting, then features drawn from the image will also be affected. The same surface may be classified as belonging to different classes depending on the direction from which it was lit. The effect can be modelled and either accounted for [MC97], or counteracted [CM95], *if* the direction of the illumination is known. However, in many cases this information is not available. The aims of this paper therefore are:

1. to propose a new model of the dependency of texture features on lighting direction, and

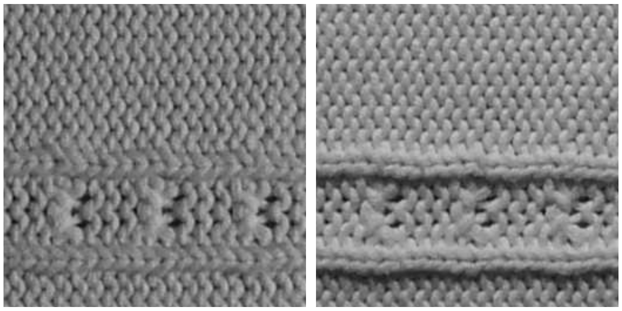


Figure 1: Two images of the same surface texture sample captured using different illuminant tilt angles

2. to use this theory to develop a novel classifier that can classify surface textures and simultaneously estimate the illumination conditions.

Relatively little work has been published on these subjects. Chantler[Cha95] verified and generalised Kube and Pentland's model of the effect of illumination direction on fractal surfaces[KP88]. Dana, Nayar, van Ginneken and Koenderink established the Columbia-Utrecht database of real world surface textures which they used to investigate bidirectional texture functions [DNvGK97]. Later they developed histogram [DN98, vGKD99] and correlation models [DN99] of these textures. Leung and Malik [LM99, LM01] Cula and Dana [OK01] and Varma and Zisserman [MA02b, MA02a] all developed classification schemes using filter banks and 3D 'textons' for the purposes of illumination and viewpoint invariant classification.

The emphasis of this paper however, is different from the empirical approaches discussed above as it develops a new theory of feature behaviour from first principles. It is this theory that is used to develop a novel classifier that simultaneously estimates lighting direction and classifies surface texture.

A deterministic model of the effect of the light source direction on texture features is developed and expressed in probabilistic terms. That is we can state the probability of a particular feature vector occurring – for a given texture illuminated under known lighting conditions. Using Bayes' theorem we can therefore find the most likely lighting direction for each class of texture. To classify, we assume that the test sample belongs to each texture class in turn and estimate the most likely lighting direction given that assumption. By comparing the relative likelihoods of each candidate we can estimate to which class the test sample belongs, and implicitly from which direction it was lit.

We assess the performance of the classifier in two stages. First we use 30 real textures to test its ability to estimate illumination tilt (azimuth) and to perform texture classification given the illumination slant (zenith). Second, we use another database of 25 real textures to assess how well the classifier can perform when both the slant and the tilt angles of the illumination are unknown.

The algorithm was found to be effective for tilt angle estimation. Slant estimation was poorer though the dataset was limited. The classifier was applied to the samples under 24 different lighting directions and achieved a classification rate of 98%.

1.1 Organisation of paper

Section 2 presents our theory of the behaviour of texture features as a function of illumination direction. Section 3 uses this model to develop an illuminant invariant classification scheme. Section 4 presents the results from experiments using two texture image databases.

2 Modelling Feature Vector Behaviour

This section derives an expression for the mean value of a texture feature as a function of the illumination's tilt and slant angles. First however, we define the axis system we use and give a short derivation of the linear illumination model, both of which are necessary for the development of the following theory.

2.1 Axis system

Figure 2 defines the geometry of our setup. The optical axis of the camera is aligned along the z-axis. The surface texture is placed in the x-y plane. The slant (zenith) angle of the illumination is the angle the illumination vector makes with the z-axis. The tilt (azimuth) angle is the angle the illumination vector makes with the x-axis when it is projected onto the x-y plane.

2.2 A Linearised Model of Lambert's Cosine Law

We assume that the surface is Lambertian, has low slope angles, is illuminated by a distant point light source, that there are no significant shadows or inter-reflections, and that the camera projection is orthographic. Ignoring the albedo factor Lambert's cosine rule may be expressed as:

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

where:

$i(x,y)$ is the radiant intensity;
$p(x,y)$ is the partial derivative of the surface height function in x direction;

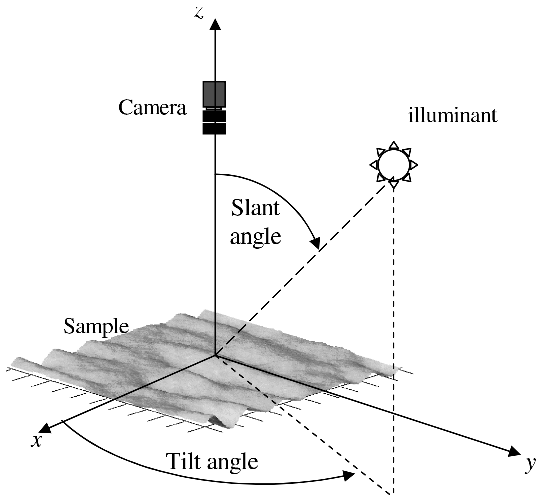


Figure 2: Axis system

$q(x, y)$ is the partial derivative of the surface height function in y direction;
τ is the tilt angle of the illumination; and
σ is the slant angle of the illumination.

For p and $q \ll 1$ we can use a truncated Taylor's series to linearise this equation about $(p = 0, q = 0)$:

$$i(x, y) = -\cos(\tau)\sin(\sigma)p(x, y) - \sin(\tau)\sin(\sigma)q(x, y) + \cos(\sigma)$$

Transforming the above into the frequency domain and discarding the mean term we obtain:

$$\begin{aligned} \mathcal{I}(\omega,\theta) &= [-\cos(\tau)\sin(\sigma)\,i\,\omega\,\cos(\theta) - \sin(\tau)\sin(\sigma)\,i\,\omega\,\sin(\theta)]\mathcal{H}(\omega,\theta) \\ \Longleftrightarrow \mathcal{I}(\omega,\theta) &= -i\,\omega\,\sin(\sigma)\cos(\theta-\tau)\mathcal{H}(\omega,\theta) \qquad (2) \end{aligned}$$

where:

$\mathcal{I}(\omega, \theta)$ is the Fourier transform of the image intensity function;
$\mathcal{H}(\omega, \theta)$ is the Fourier transform of the surface height function; and
(ω, θ) are polar frequency coordinates.

In this paper it is more convenient to express equation 2 in its power spectrum form:

$$I(\omega, \theta) = \omega^2 \cos^2(\theta - \tau)\sin^2(\sigma)H(\omega, \theta) \qquad (3)$$

where:

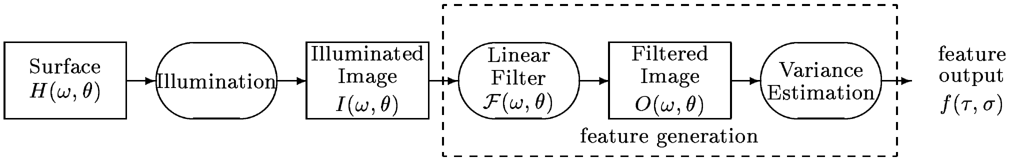


Figure 3: Feature generation

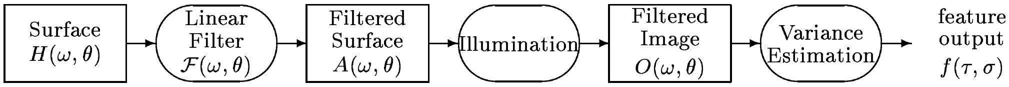


Figure 4: The feature generation model with illumination and linear filter processes interchanged

$I(\omega,\theta)$	is the image power spectrum; and
$H(\omega,\theta)$	is the surface power spectrum.

Equations 2 and 3 are similar to Kube & Pentland's fractal imaging model[KP88] which was adapted and verified experimentally by Chantler and then McGunnigle [Cha95, MC01]. In the context of this paper the most important features of this model are the $\cos(\theta-\tau)$ and $\sin(\sigma)$ factors. In particular the cos term shows that the imaging function acts as a directional filter of the surface height function.

2.3 The output of Linear Texture Filters and their Features

We define a Linear Texture Feature as a linear filter followed by a variance estimator[RH99]. The process formed by applying such a feature to an image is as shown in Fig. 3. Since the model of the illumination process (equation 3) is also linear we may exchange it with the linear filter (Fig. 4). We use $A(\omega,\theta)$ to represent the notional power spectrum of the output of the linear texture filter applied directly to the surface height function.

Thus to determine the mean output of a Linear Texture Filter we simply have to apply Kube & Pentland's model in the form of equation 3 to $A(\omega,\theta)$ and develop an expression for the variance of the subsequent output.

The mean output of a linear texture feature is the variance of the output $o(x,y)$ of its linear filter:

$$f(\tau,\sigma) = \mathcal{VAR}(o(x,y)) \tag{4}$$

If we assume that $o(x,y)$ has a zero mean and that $O(\omega,\theta)$ is its power spectrum expressed in polar co-ordinates then we may express equation 4 as:

$$f(\tau,\sigma) \quad = \quad \int_0^\infty \int_0^{2\pi} \omega\, O(\omega,\theta)\, d\theta\, d\omega \tag{5}$$

Using equation 3 we can express $O(\omega,\theta)$ as follows:

$$\begin{aligned} O(\omega,\theta) &= |\mathcal{F}(\omega,\theta)|^2\omega^2\cos^2(\theta-\tau)\sin^2(\sigma)H(\omega,\theta) \\ \Longleftrightarrow O(\omega,\theta) &= \omega^2\cos^2(\theta-\tau)\sin^2(\sigma)A(\omega,\theta) \qquad (6) \end{aligned}$$

where:

$\mathcal{F}(\omega,\theta)$ is the transfer function of the linear filter; and
$A(\omega,\theta)$ $= H(\omega,\theta)|\mathcal{F}(\omega,\theta)|^2$

Substituting equation 6 into equation 5 we obtain:

$$f(\tau,\sigma) = \int\limits_{-\infty}^{\infty} \omega^3\sin^2(\sigma)\int\limits_{0}^{2\pi}\cos^2(\theta-\tau)A(\omega,\theta)\,d\theta\,d\omega \qquad (7)$$

Using $\cos^2(x) = 1/2\,(1+\cos(2x))$ and $\cos(x-y) = \cos(x)\cos(y)+\sin(x)\sin(y)$ gives:

$$f(\tau,\sigma) = \int\limits_{0}^{\infty}\omega^3\sin^2(\sigma)\int\limits_{0}^{2\pi} 1/2\,[1+\cos(2\theta)\cos(2\tau)+\sin(2\theta)\sin(2\tau)]A(\omega,\theta)\,d\theta\,d\omega \qquad (8)$$

hence:

$$f(\tau,\sigma) = \sin^2(\sigma)(a+b\cos(2\tau)+c\sin(2\tau)) \qquad (9)$$

and:

$$f(\tau,\sigma) = \sin^2(\sigma)(a+d\cos(2\tau+\phi)) \qquad (10)$$

where:

$$\begin{aligned} a &= 1/2\int\limits_{0}^{\infty}\omega^3\int\limits_{0}^{2\pi}A(\omega,\theta)\,d\theta\,d\omega \\ b &= 1/2\int\limits_{0}^{\infty}\omega^3\int\limits_{0}^{2\pi}\cos(2\theta)A(\omega,\theta)\,d\theta\,d\omega \\ c &= 1/2\int\limits_{0}^{\infty}\omega^3\int\limits_{0}^{2\pi}\sin(2\theta)A(\omega,\theta)\,d\theta\,d\omega \end{aligned}$$

$$d \equiv \sqrt{b^2+c^2}, \qquad \sin(\phi)\equiv\frac{c}{d} \qquad \cos(\phi)\equiv\frac{b}{d}$$

The above parameters (a, b etc.) are all functions of the surface height function and the linear texture filter. None of them is a function of illuminant tilt (τ) or slant (σ). Thus equation 10 predicts that the output of a texture feature based on a linear filter is proportional to $\sin^2(\sigma)$ and is also a sinusoidal function of illuminant tilt [1] with a period π radians.

Figure 5 shows the behaviour of four texture features that are typical [2] of the results that we obtained using 30 real textures [CSPM02]. They clearly show that the features' outputs are sinusoidal functions of the illuminant's tilt angle (τ).

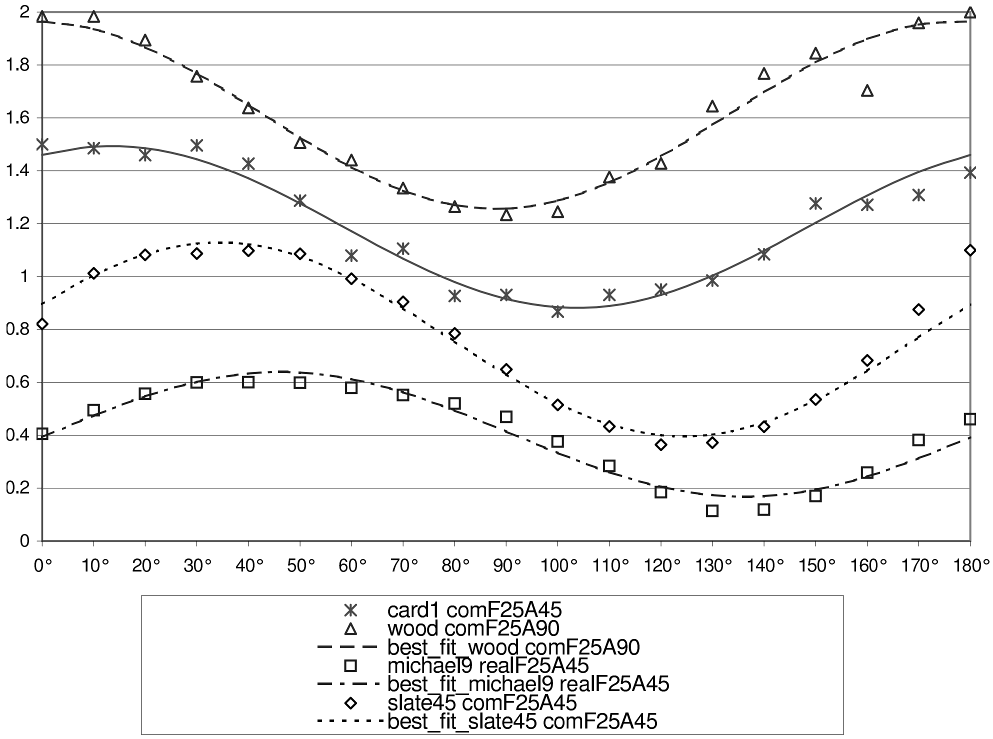


Figure 5: Typical sinusoidal behaviour of texture features. Each plot shows how one output of one feature varies when it is repeatedly applied to the same physical texture sample, but under varying illuminant tilt angles. Discrete points indicate measured output and the curves show the best-fit sinusoids. The vertical axis indicates mean feature output, while the horizontal axis indicates illuminant tilt angle.

[1] In the case of $A(\omega, \theta)$ being isotropic (for instance if both the surface and the filter are isotropic) the response will be independent of τ. However, if an isotropic filter is applied to a directional surface then $A(\omega, \theta)$ will not be isotropic and the tilt response will be a sinusoidal function of tilt.

[2] For display we selected the four feature/texture combinations that gave results that were closest to the median error when they were used to find the best-fit sinusoid functions.

2.4 Behaviour in a Multi-Dimensional Feature Space

As texture classifiers normally exploit the output of several features it is important to investigate the behaviour of texture features in a multi-dimensional decision space.

If two different features are derived from the same surface texture, the results can be plotted in a two-dimensional x, y feature space. Applying our sinusoidal model (equation 10) to each feature we obtain:

$$x = f_1(\tau, \sigma) = \sin^2(\sigma)(a_1 + d_1 \cos(2\tau + \phi_1))$$

$$y = f_2(\tau, \sigma) = \sin^2(\sigma)(a_2 + d_2 \cos(2\tau + \phi_2))$$

Changes of the illuminant slant (σ) therefore simply scale the 2D feature vector plot. However, variation of tilt causes a more complex behaviour. Since the frequency of the two cosines is the same these two equations form two simple harmonic motion components. Therefore the trajectory in 2D feature space as a function of tilt is in general an ellipse.

There are two special cases. If the surface is isotropic and the two filters are identical except for a difference in direction of 90°, the mean value and the oscillation amplitude of the two features are the same and the phase difference becomes 180°. Thus the scatter plot for an isotropic texture and two identical but orthogonal filters is a straight line.

If the surface is isotropic and the two filters are identical except for a difference in direction of 45°, the mean value and the oscillation amplitude of the two features are again the same but the phase difference is now 90°. In this case the scatter plot is a circle.

The line and the circle are the two special cases of all possible curves. In the general case of two or more filters the result is an ellipse or a trajectory on a super-ellipse.

Figure 6 shows the behaviour of two Gabor filters (F25A45com and F25A0com) as a function of illuminant tilt, for six real textures. It clearly shows the elliptical behaviour of the cluster means.

2.5 A Probabilistic Model of Feature Behaviour

In practice a feature's actual behaviour (f_i) will differ slightly from the model's prediction (equation 9). We model the difference as a zero mean, normally distributed random variable with standard deviation s. We can now express the relationship between the feature and lighting direction for a given texture class k in probabilistic terms:

$$p_k(f_i|\tau, \sigma) = \frac{1}{s_i\sqrt{2\pi}} exp[-\frac{[f_i - \sin^2(\sigma)(a_i + b_i \cos(2\tau) + c_i \sin(2\tau))]^2}{2s_i^2}] \quad (11)$$

where $p_k(f_i|\tau, \sigma)$ is the probability of the event of feature i having value f_i occurring, given that the texture k is lit from (τ, σ).

The feature vector, F, is composed of i features. Assuming these are independent the joint distribution is:

$$P_k(F|\tau,\sigma) = \prod_i \frac{1}{s_i\sqrt{2\pi}} exp[-\frac{[f_i - \sin^2(\sigma)(a_i + b_i\cos(2\tau) + c_i\sin(2\tau))]^2}{2s_i^2}] \tag{12}$$

Thus our probabilistic model of the behaviour of an i-dimensional feature vector F requires the estimation of $4i$ parameters (i.e. i sets of a_i, b_i, c_i and s_i) for each texture.

3 Classification

From the 2D scatter diagram (figure 6) it is obvious that linear and higher order classifiers are likely to experience difficulty in dealing with this classification problem. We have therefore chosen to exploit the hyper-elliptical model of feature behaviour described above.

The easiest way to understand the classifier is to consider the 2D case (Figure 6). In this system a test texture's feature vector is represented as a single point on the scatter diagram. The classification task therefore becomes one of finding the point on each class ellipse which is closest (in a probabilistic sense) to the feature vector. The distances to these points, weighted by class variances, provide class likelihoods. The test texture is assigned to the class with the largest likelihood.

The classifier is therefore trained by parameterising the elliptical probabilistic model (equation 12) for each candidate class. Each texture sample must be imaged under different illumination directions and features calculated from these images. We recommend that at least three images should be taken at two or more slants. In this work we use 12 images at two slant angles. The parameter values of the model are calculated to give the best fit to the data. This allows us to predict the likelihood of a particular feature value F, for a given texture class k, lit from a given direction (τ, σ).

Presented with a feature vector, the classifier uses a probabilistic model to identify the most likely lighting direction and texture class. The probability of a texture having been illuminated from (τ, σ) given a particular feature vector can be related to Equation 12 using Bayes' theorem:

$$P_k(\tau,\sigma|F) = \frac{P_k(F|\tau,\sigma)P_k(\tau,\sigma)}{P_k(F)} \tag{13}$$

Now, assuming all lighting directions are, *a prior*, equally likely, $P(\tau,\sigma)$ is constant and because we are only interested in the *relative* probabilities of the values of σ and τ at a given F we may replace $P_k(F)$ with a constant, i.e.

$$P_k(\tau,\sigma|F) = \alpha P_k(F|\tau,\sigma) \tag{14}$$

The most likely direction of the light source, $\hat{\tau}$, $\hat{\sigma}$ for each texture is estimated by *maximising* the likelihood function of that texture.

$$\hat{\tau}, \hat{\sigma} = \underset{\tau, \sigma}{ArgMax} \; P_k(F|\tau, \sigma) \tag{15}$$

To find the maximum we take logs:

$$\ln P_k(F|\tau, \sigma) = \ln \prod_i (\frac{1}{s_i\sqrt{2\pi}}) + \sum_i \frac{[f_i + \sin^2(\sigma)(a_i + b_i \cos(2\tau) + c_i \sin(2\tau))]^2}{2s_i^2} \tag{16}$$

obtain the partial derivatives w.r.t. τ and σ and equate both to zero. The trig. terms are simplified by substituting $x = \sin^2(\sigma)$ and $y = \cos 2\tau$ and the two resulting equations are solved to provide a 12^{th} order polynomial in x. This quite straight forward manipulation results in a long series of expressions that contain many terms. It is therefore not repeated here as the full treatment may be obtained from [Pen02]. The polynomial is solved using a standard Matlab routine. The resulting multiple solutions are tested to obtain the values of $\hat{\tau}$ and $\hat{\sigma}$ that maximise 16.

We now have a series of k competing hypotheses about the class of the sample and the direction it was lit from. Again, we are interested only in relative probabilities. If we assume the classes are, initially, equally likely, the most likely class can be identified by finding the highest class probability, i.e. by evaluating Equation 17.

$$\hat{k} = \underset{k}{ArgMax} \; P_k(F|\hat{\tau_k}, \hat{\sigma_k}) \tag{17}$$

3.1 Summary of Classification Process

In this section we have developed a classifier assuming that that the surface is Lambertian and of constant albedo, that it has low slope angles, that it is illuminated by a distant point light source, that there are no significant shadows or inter-reflections, and that the camera projection is orthographic. The classification process comprises five steps:

Train the classifier: Use the training images to estimate the $4i$ parameters (a_i, b_i, c_i, s_i for each texture class).

Calculate the feature vector: Calculate $F = [f_1, f_2 .... f_i]$ from the image of the test texture imaged under unknown illumination conditions.

Calculate the maximum likelihoods: Use the optimisation procedure described above, and the feature vector F, to find the illumination conditions (τ, σ) that maximises equation 16 for each of the k training textures.

Classify: Assign the unknown texture to the class $\hat{k}$ with the largest of the log-likelihoods P_k found in the previous step.

Output: The selected texture class $\hat{k}$ together with the estimate of the illumination angles for that class $(\hat{\tau_k}, \hat{\sigma_k})$ are returned as the classifier's output. The value of the corresponding probability $P_{\hat{k}}(F|\hat{\tau_k}, \hat{\sigma_k})$ can be returned as the confidence in the classification result.

4 Experiments

For evaluation we used two databases that contained images of surface textures captured under a wide variety of known illumination conditions. The first contained 30 real images obtained using only variation in illuminant tilt. The second, comprising 25 surface textures, was captured using variation in both tilt and slant.

4.1 Robustness to Tilt

This section describes experiments that were performed in order to test the ability of our classification scheme to classify textures of unknown class and unknown illuminant tilt, but of known illuminant slant. Since the slant was known, a slightly simpler version of the classifier was employed - essentially equation 16 was optimised over a single variable (τ) rather than two.

4.1.1 The Image-set

Thirty physical texture samples were used in these experiments. 512x512 8-bit monochrome images were obtained from each sample using illumination tilt angles ranging between 0° and 180° incremented by either 10° or 15° steps. All textures were illuminated at a slant angle of 45°. The illumination source was mounted approximately 1.5m from the target.

This arrangement clearly did not approximate a point source at infinity as a smooth illumination gradient was observed on a flat test surface. We have compensated for this illumination gradient in previous experiments and while we have seen that this makes a significant difference to global height estimation it has little affect on band pass filters such as Gabors. We therefore chose not to compenstate for these effects in the experiments reported here.

The final dataset contains over 600 images. Appendix 1 contains one example image of every texture.

4.1.2 The Feature-set

The classifier's features are estimates of the variance of images produced by filtering the input image with a set of Gabor[JF91] and Laws[Law80]

Table 1: Tilt only experiment: classification errors.

Percentage of Classification errors			
Number of Filters	*six*	*five*	*four*
Detected errors	0.81%	2.02%	3.77%

filters. Gabor filters are Gaussians modulated by complex exponentials—they have a centre frequency ω and orientation ϕ. In our nomenclature they are denoted by $comF\omega A\phi$, where ω is specified in cycles per image and ϕ is in degrees. Laws developed his filters purely empirically from a set of three very simple spatial filters. They normally have a much wider bandwidth than Gabor filters and in our experience often provide a useful complement to Gabors.

The Gabor filters used were comF25A0, comF25A45, comF25A90, comF25A135, comF50A45 while the Laws filters selected were $L5E5$ and $E5L5$. Three combinations of features were used:

set six: four complex Gabor Filters and two Laws Filters.

set five: five complex Gabor Filters.

set four: four complex Gabor Filters.

4.1.3 Tilt Classification Results

Both classification accuracy and the accuracy of illuminant tilt estimation were investigated.

Figures 7 and 8 show the errors that occurred in estimating the illuminant tilt angles. Figure 7 shows the root mean square error in tilt (the mean being calculated over each set of test images obtained from a single texture for a particular classifier). Figure 8 shows a histogram of the all the errors that occurred. Both of these charts show that in the majority of cases the illuminant tilt is estimated to within 5°. Only in a very small number of cases, such as *card1* and *and7*, does the error exceed 10°.

Table 1 shows the texture classification error rates that occurred. Table 2 details the misclassifications for the six and five filter feature sets. For instance it shows that using six filters, the classifier misclassified *slab45* imaged using an illuminant tilt angle of 70°, as *michael6* imaged at a tilt angle of 18°. Examining the images in the appendix explains some of the misclassifications e.g. *twins45, stri45* and *iso45* appear similar. Others look quite different from one another e.g. *radial45* and *michael3*. However, it should be noted that the distinction between these two textures blurs when *michael3* is imaged at 90° of tilt, as this filters out much of the 0° spaghetti texture.

Table 2: Tilt only experiment: misclassification details.

Misclassification					
Input		*six filters*		*four filters*	
texture	tilt	texture	tilt	texture	tilt
stones2	50			chips1	23
stones2	170			michael7	2.2
radial45	170			michael3	90
slab45	70	michael6	18	michael6	22
twins45	90	stri45	93	iso45	90
michael2	170			michael8	180

4.2 Robustness to Tilt and Slant

This section describes the experiments that were performed in order to test the classification scheme's ability to classify images of texture captured under unknown illumination slant and tilt angles.

4.2.1 The Image-set

12-bit 512x512 monochrome images of 25 different samples of surface texture were captured at slant angles of 45° and 60° and tilt angles of 30° increments. Half were used for training and half kept for testing. Example images are shown in appendix 2.

4.2.2 The Feature-set

As the classification task was more challenging, a larger set of 12 Gabor filters was used. These filters were combined into *banks* as indicated in Table 3.

4.2.3 Tilt and slant Classification results

The classifier was again assessed both in terms of its ability to estimate the illumination angles and its ability to perform classification.

The accuracy of tilt estimation is shown in Figure 9 (top). 76% of the estimates were within 5° of the correct value, and 82% were within 10°. Only one texture sample was more than 20° in error.

The accuracy of slant estimation is shown in Figure 9 (bottom). There are several points to note regarding this. First, two training slants, separated by 15° were used. 26% of the tests were more than 7.5° in error. Second, estimation from 45° was significantly more accurate than estimation from 60° (52% of samples have less than 2° of error for the 45° case, compared to

filter	**Gabor filter bank**						
	12	10	8	6	4	3	2
comF20A0	X	X	X	X	X	X	X
comF20A45	X	X	X				
comF20A90	X	X	X	X	X	X	X
comF20A135	X	X	X				
comF30A0	X						
comF30A45	X	X	X	X	X	X	
comF30A90	X						
comF30A135	X	X	X	X	X		
comF40A0	X	X	X	X			
comF40A45	X	X					
comF40A90	X	X	X	X			
comF40A135	X	X					

Table 3: Tilt and slant experiment: Gabor filter bank configurations.

only 4% for the 60°case). Third, the image samples that perform poorly for tilt estimation correspond to those that perform badly for slant estimation—these tend to be drawn from the AD* and AF* groups (repeating primitives and fabrics) both of which experience significant shadowing. The last two points suggest that the prime source of inaccuracy is shadowing.

The second, more important criterion for the classifier is classification accuracy. We applied 6 feature sets composed of between 3 and 12 Gabor filters to the dataset, i.e. 25 samples lit from 24 different directions. The overall error rate is shown in Figure 10. The most effective feature vector, composed of 10 features, gave a 98% classification rate. Increasing the number of features gave a small increase in the error rate and also led to problems in obtaining numerical solutions to the polynomial. Reducing the number of features increased the error rate—with the most significant increase occurring for sets of less than 6 features.

5 Conclusions

In this paper we have:

1. presented a completely new and formal model of texture feature behaviour as a function of the lighting vector (equation 10)

2. we have used this new theory to develop a novel classifier that can classify surface textures and *simultaneously* estimate the illumination conditions.

The first point above is the more significant. The model is general to a large class of conventional texture features and it explains, from first principles, why these features are simple trigonometric functions of the illumination conditions.

Hence, given better *a priori* information (i.e. the model) it should be possible to build a variety of improved applications ranging from illuminant estimators through to classifiers and segmentation tools.

We applied the model to the texture classification process and found that, despite the many assumptions that were made during its derivation, it represents the behaviour of the Gabor features that we computed surprisingly well. Admittedly the test set was limited to images taken of 30 textures and contained no really specular surfaces. However, shadowing, local illumination effects, and albedo variation are clearly evident in many of our images. We therefore feel that, with the exception of highly specular surfaces, the model has proven to be robust to violation of many of the initial assumptions.

This has allowed us to develop a reliable classifier that simultaneously estimates the direction of the illumination while performing the classification task. Tests with 25 real textures have shown that the system is capable of reliably classifying a range of surface texture while accurately resolving the illumination's tilt angle, and to a lesser extent its slant angle.

References

[Cha95] M.J. Chantler. Why illuminant direction is fundamental to texture analysis. *IEE Proc. Vision, Image and Signal Processing*, 142(4):199–206, August 1995.

[CM95] M.J. Chantler and G. McGunnigle. Compensation of illuminant tilt variation for texture classification. In *Proceedings of 5th International Conference on image processing and its applications*, pages 767–761, 1995.

[CSPM02] M.J. Chantler, M. Schmidt, M. Petrou, and G. McGunnigle. The effect of illuminant rotation on texture filters: Lissajous's ellipses. In *ECCV2002, European Conference on Computer Vision*, volume III, pages 289–303, 2002.

[DN98] K.J. Dana and S.K. Nayar. Histogram model for 3d textures. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 618–624, 1998.

[DN99] K.J. Dana and S.K. Nayar. Correlation model for 3d texture. In *Proceedings of ICCV99: IEEE International Conference on Computer Vision*, pages 1061–1067, 1999.

[DNvGK97] K.J. Dana, S.K. Nayar, B. van Ginneken, and J.J. Koenderink. Reflectance and texture of real-world surfaces. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 151–157, 1997.

[JF91] A.K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24(12):1167–1186, December 1991.

[KP88] P.R. Kube and A.P. Pentland. On the imaging of fractal surfaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(5):704–707, September 1988.

[Law80] K.I. Laws. *Textured Image Segmentation.* PhD thesis, Electrical Engineering, University of Southern California, 1980.

[LM99] T. Leung and J. Malik. Recognizing surfaces using three-dimensional textons. In *Proceedings of ICCV99: IEEE International Conference on Computer Vision*, pages 1010–1017, 1999.

[LM01] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, June 2001.

[MA02a] Varma M. and Zisserman A. Classifying images of materials: Achieving viewpoint and illumination independence. In *ECCV2002, European Conference on Computer Vision*, pages 255–271, 2002.

[MA02b] Varma M. and Zisserman AP. Classifying materials from images: to cluster or not to cluster? In *Texture2002: The 2nd international workshop on texture analysis and synthesis, 1 June 2002, Copenhagen*, pages 139–144, 2002.

[MC97] G. McGunnigle and M.J. Chantler. A model-based technique for the classification of textured surfaces with illuminant direction invariance. In *Proceedings of BMVC97: British Machine Vision Conference*, volume 2, pages 470–479, 1997.

[MC01] G. McGunnigle and M.J. Chantler. Evaluating kube and pentland's fractal imaging model. *IEEE Trans. on Image Processing*, 10(4):534–542, April 2001.

[OK01] Cula O.G. and Dana K.J. Recognition methods for 3d textured surfaces. In *Proceedings of SPIE, San Jose*, January 2001.

[Pen02] A. Penirschke. Illumination invariant classification of 3d surface textures. *Research Memorandum*, (RM/02/4), 2002.

[RH99] T. Randen and J.H. Husoy. Filtering for texture classification: A comparative study. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(4):291–310, April 1999.

[vGKD99] B. van Ginneken, J.J. Koenderink, and K.J. Dana. Texture histograms as a function of irradiation and viewing direction. *International Journal of Computer Vision*, 31(2/3):169–184, April 1999.

List of Figures

List of Tables

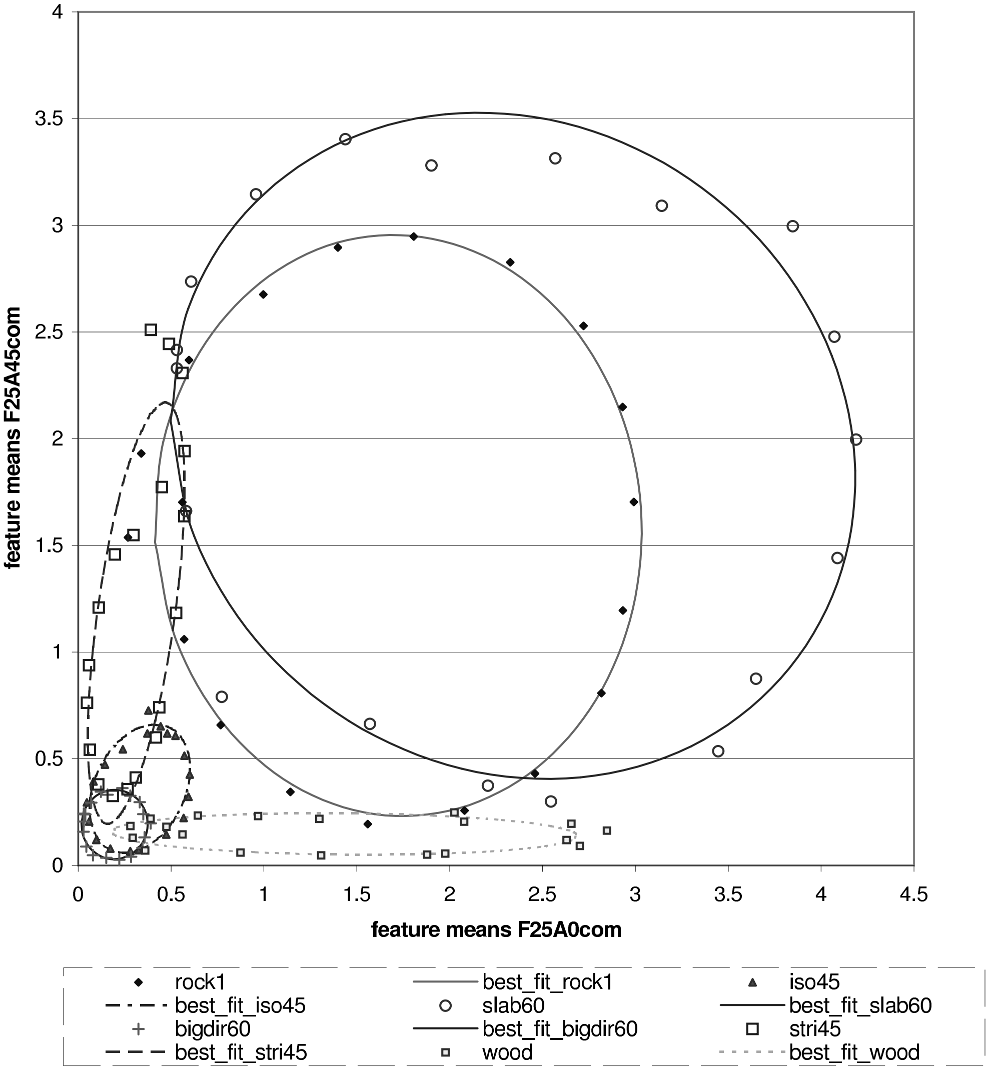


Figure 6: The behaviour of six textures in the comF25A0/comF25A45 feature space together with the best fit ellipses. Each point on an ellipse corresponds to a different value of illuminant tilt. All points on the same ellipse correspond to the same surface.

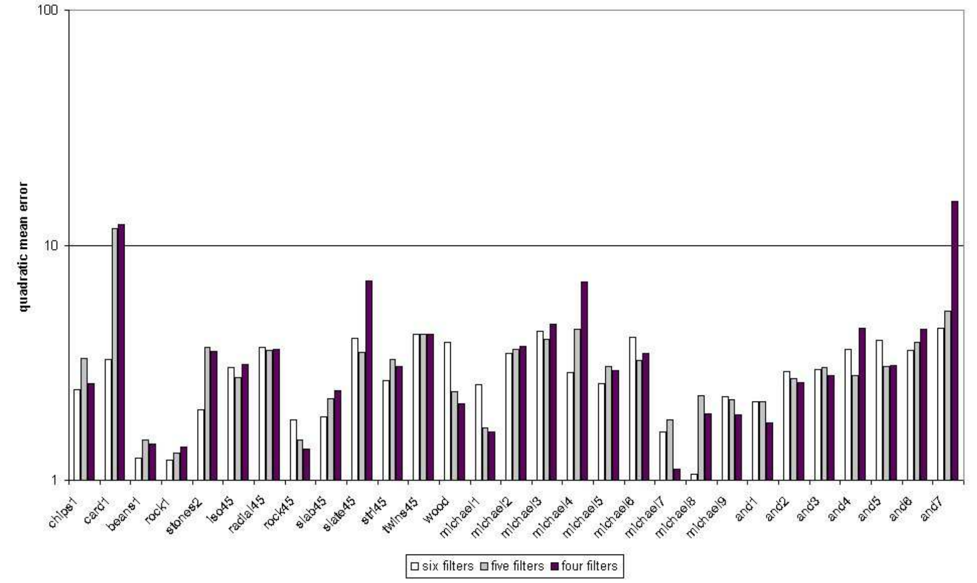


Figure 7: Tilt only experiment: root mean square tilt estimation errors (in degrees) for the use of four, five and six feature measures for each surface texture.

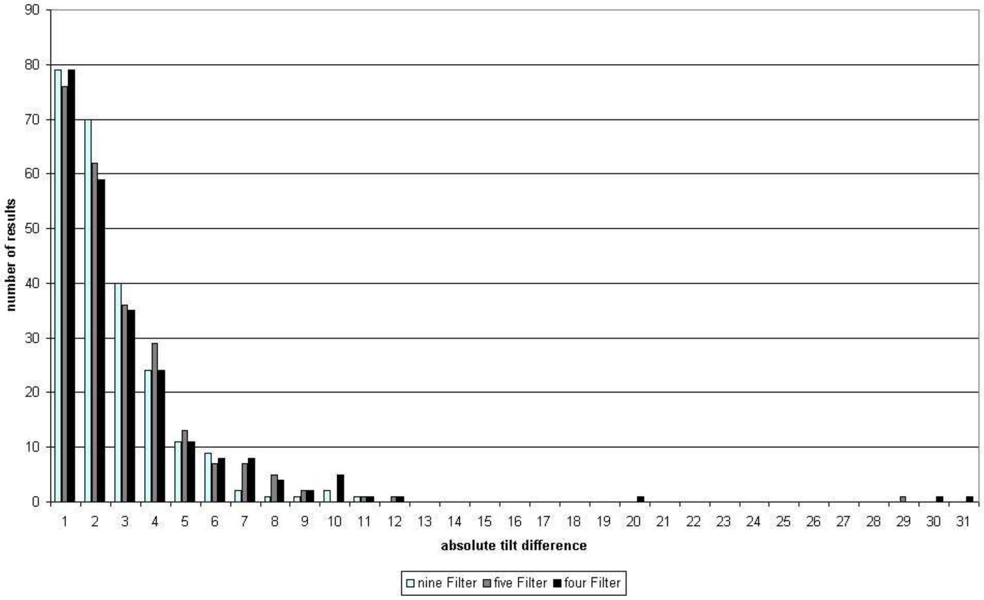
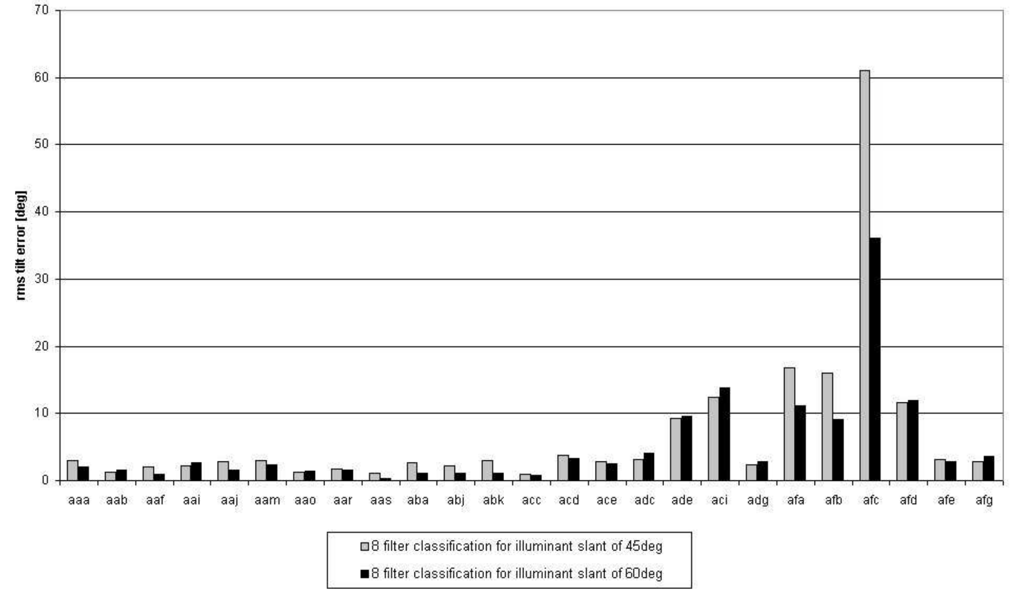


Figure 8: Tilt only experiment: histogram of tilt estimation errors (in degrees) for the three filter banks



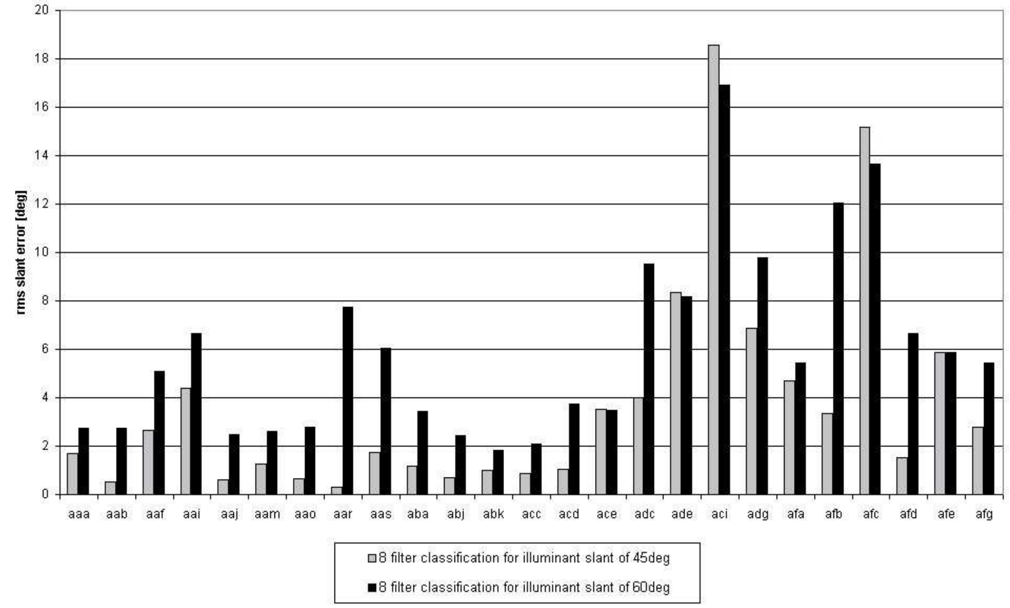


Figure 9: Tilt and slant experiment: root mean square tilt error (top) and rms slant error (bottom).

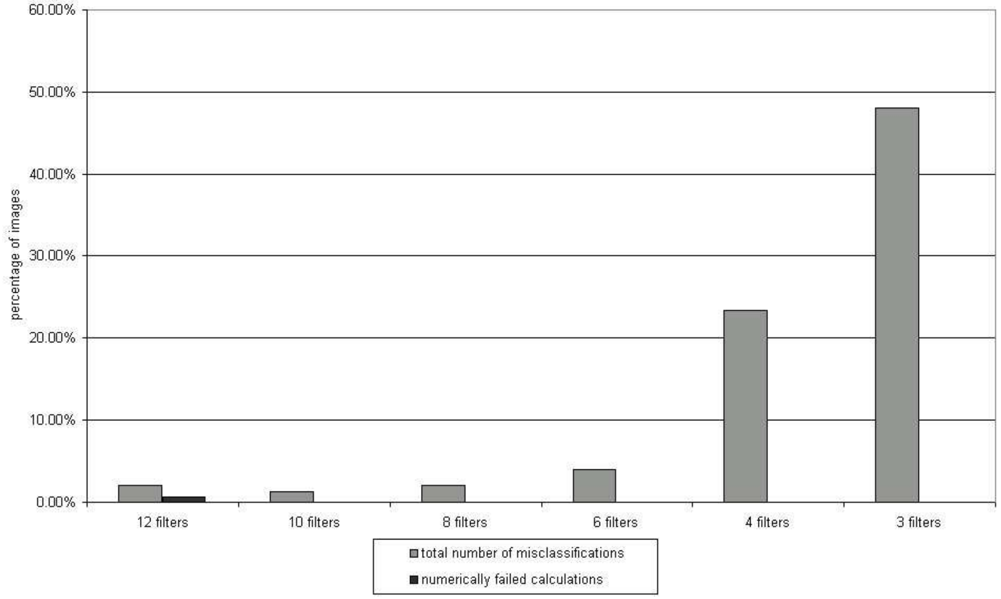
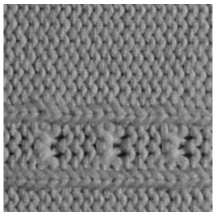
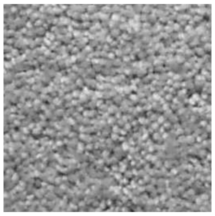


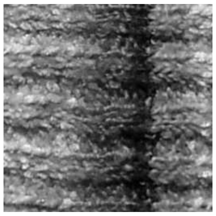
Figure 10: Tilt and slant experiment: classification errors

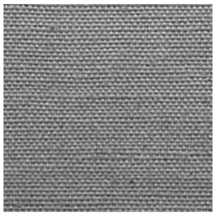
Appendix 1: Tilt only Database

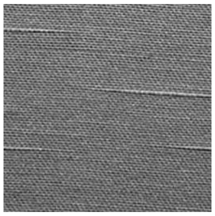
One example image of each texture is shown below:

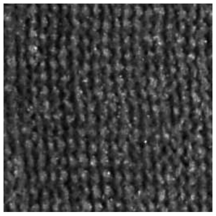

and1(zoomed)

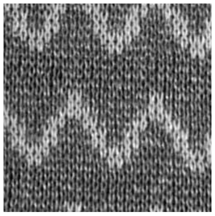

and2(zoomed)


and3(zoomed)


and4(zoomed)

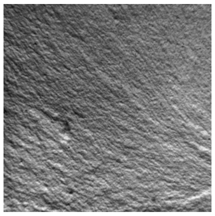

and5(zoomed)


and6(zoomed)

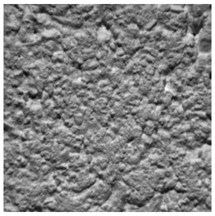

and7(zoomed)

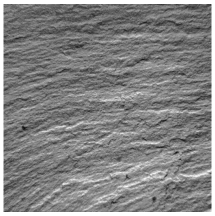
bigdir45

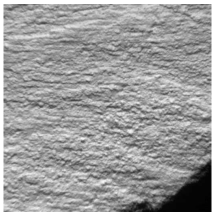
iso45

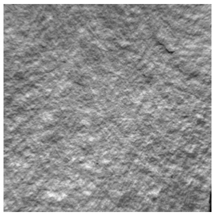

radial45

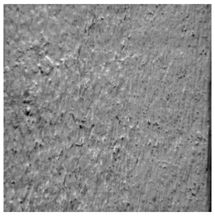
rock45

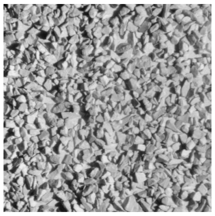

slab45

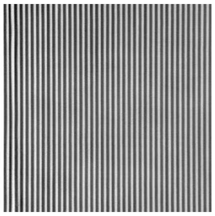

slate45

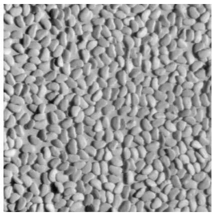

stri45


twins45

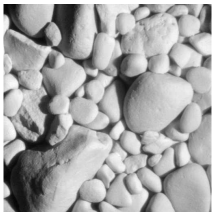

wood

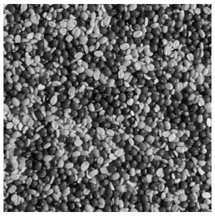

chips1

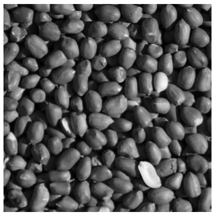

card1

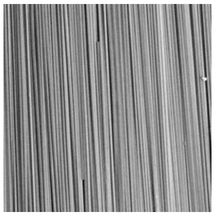

beans1

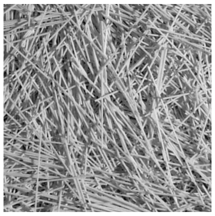
rock1

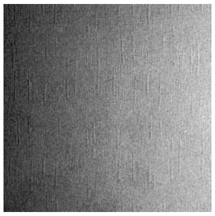

stones2


michael1

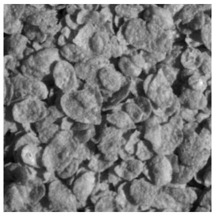

michael2

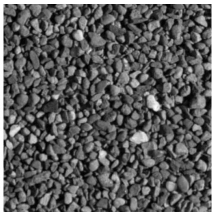

michael3

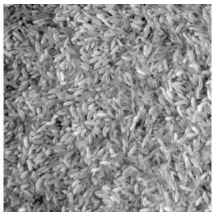

michael 4


michael5

michael6

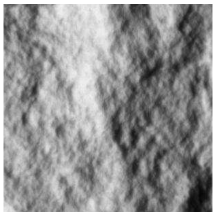

michael7

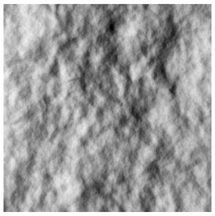

michael8

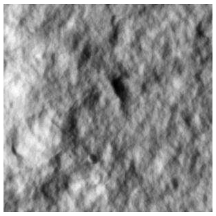

michael9

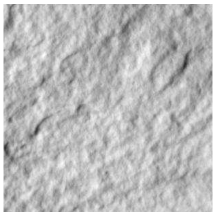
Appendix 2: Tilt and Slant Database

One example image of each texture is shown below:

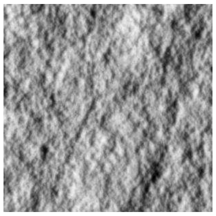

aaa

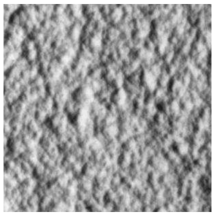

aab

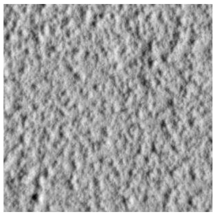

aaf

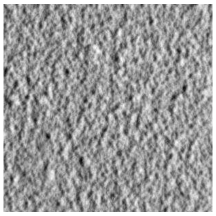

aai

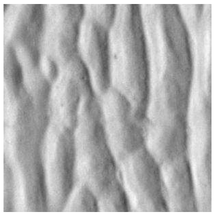
aaj

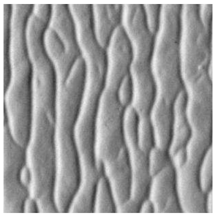

aam

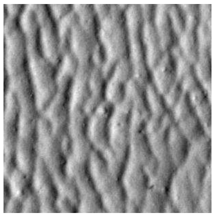

aao

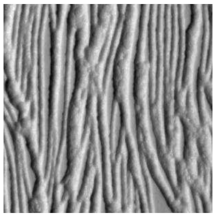

aar

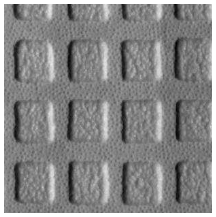

aas

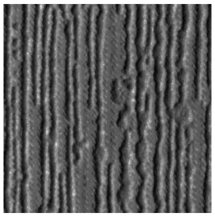

aba

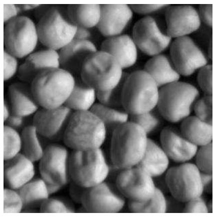

abj

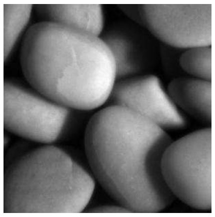

abk

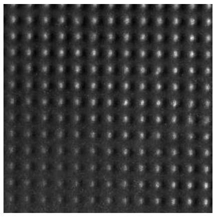

acc

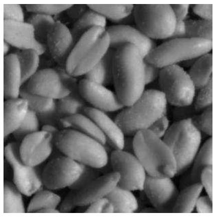

acd

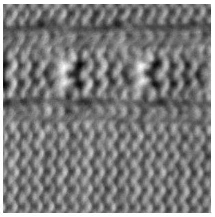

ace

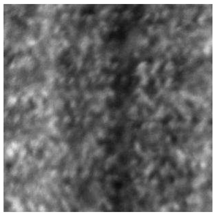

adc

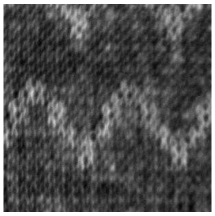

ade

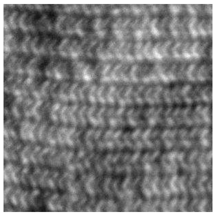

aci

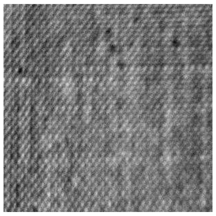

adg

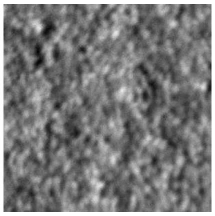

afa


afb


afc


afd


afe


afg