Shape from Texture of Developable Surfaces via Fourier Analysis

Fabio Galasso and Joan Lasenby

University of Cambridge, Cambridge, UK

Abstract. Shape from texture has received much attention in the past few decades. We propose a computationally efficient method to extract the 3D shape of developable surfaces from the spectral variations of a visual texture. Under the assumption of homogeneity, the texture is represented by the novel method of identifying ridges of its Fourier transform. Local spatial frequencies are then computed using a minimal set of selected Gabor filters. In both orthographic and perspective projection cases, new geometric equations are presented to compute the shape of developable surfaces from frequencies. The results are validated with semi-synthetic and real pictures.

1 Introduction

Shape from Texture was first introduced by Gibson 50 years ago. In [1] he suggests that texture can provide an important shape cue. However, for a machine the solution to this problem is ill-posed. Shape from texture is generally about measuring the texture distortion in an image, and then reconstructing the surface 3D coordinates in the scene ([2], [3], [4], [5]). The model for the texture can be either deterministic or stochastic. The second allows a wider variety of textures ([4], [5], [6]) and implies local spectral measurements, usually with the Fourier transform ([5]), or more recently, wavelets ([3], [7]).

An initial assumption about the texture is always necessary, and a restrictive one can preclude applicability to real surfaces. [8] deals with texels, which are seldom found in nature, while [9] assumes isotropy, rarely the case. Homogeneity is more frequently used ([4], [2], [7]), and is the one we choose here. For deterministic textures it can be seen as periodicity, for stochastic textures it can be formalized as stationarity under translation ([5]). Under this condition we assume that all texture variations are produced only by projective geometry.

We consider the cases of an orthographic camera model, as in [6] and [10], and of a perspective camera model, as in [11] and [12]. In both cases advantages and disadvantages are presented and results are compared.

The present work takes its motivation from [12]. The texture is analyzed using Gabor filters to produce distortion information based on *local spatial frequency* (LSF). Unlike [12], we do not just rely on a dominant LSF, but we consider groups of LSFs. This extends [12] to exploit the multi-scale nature of textures.

To our knowledge the algorithm presented here is the first to consider the multi-scale nature of texture to the extent of using all main LSFs, most of the related work uses only two preferred directions in the spectral domain (e.g. [6],[13]).

Furthermore we present here new geometric equations to compute the 3D coordinates of a developable shape from its LSFs, both in the orthographic and in the perspective case.

Section 2 explains in detail how the texture is analyzed to produce distortion information. Section 3 presents the projective geometry. Section 4 shows how we can recover the 3D coordinates from the measured texture distortion. Finally, section 5 presents results.

2 Texture Description

Here we describe how to set 2D Gabor functions and their first derivatives from the information on texture supplied by the Fourier transform. The former provide local analysis to compute instantaneous frequencies, which are used to measure distortion and reconstruct the 3D coordinates of the texture surface.

2.1 Estimating the Instantaneous Frequencies

We can analyse an image $I(\mathbf{x})$ using a band-pass filter $h(\mathbf{x}, \mathbf{u})$, a function of a point $\mathbf{x}$ and of a central frequency $\mathbf{u}$, which is convolved with the image to provide the local spectrum. As in [12] we choose 2D Gabor functions:

$$h(\mathbf{x}, \mathbf{u}) = g(\mathbf{x}) e^{2\pi j \mathbf{x}\cdot\mathbf{u}} \quad \text{where} \quad g(\mathbf{x}) = \frac{1}{2\pi\gamma^2} e^{\frac{-(\mathbf{x}\cdot\mathbf{x})}{2\gamma^2}} , \tag{1}$$

with j the unit imaginary and $g(\mathbf{x})$ a 2D Gaussian function with variance γ^2.

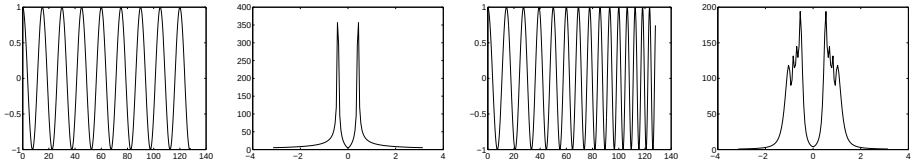
For a 2D cosine $f(\mathbf{x}) = \cos(2\pi\Omega(\mathbf{x}))$ the instantaneous frequency is given by

$$\tilde{\mathbf{u}}(\mathbf{x}) = (\tilde{u}(\mathbf{x}), \tilde{v}(\mathbf{x})) = \left(\frac{\partial\Omega}{\partial x}, \frac{\partial\Omega}{\partial y} \right). \tag{2}$$

Our goal is to measure $\tilde{\mathbf{u}}(\mathbf{x})$. [14] shows that this can be done by considering a Gabor function $h(\mathbf{x}, \mathbf{u})$, and its two first order derivatives, $h_x(\mathbf{x}, \mathbf{u})$ and $h_y(\mathbf{x}, \mathbf{u})$:

$$\begin{aligned} |\tilde{u}(\mathbf{x})| &= \frac{|h_x(\mathbf{x}, \mathbf{u}) * I(\mathbf{x})|}{2\pi |h(\mathbf{x}, \mathbf{u}) * I(\mathbf{x})|} \\ |\tilde{v}(\mathbf{x})| &= \frac{|h_y(\mathbf{x}, \mathbf{u}) * I(\mathbf{x})|}{2\pi |h(\mathbf{x}, \mathbf{u}) * I(\mathbf{x})|}. \end{aligned} \tag{3}$$

This estimate can be assumed to be correct if the measured frequency is in the pass-band of the filter. This method implies that we choose the central frequencies $\mathbf{u}$ and the spatial constants γ of the Gabor functions, i.e. their centers and width. The filters have constant *fractional bandwidth* (bandwidth divided by center frequency). This allows us to measure higher frequencies more locally

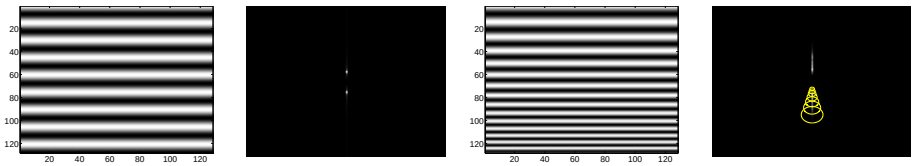


(a) 1D cosine at frequency $\tilde{u} \approx 0.42$ rad/s

(b) Amplitude of the spectrum of the cosine in 1(a)

(c) 1D cosine. Frequency varies from $\tilde{u} \approx 0.42$ rad/s to $\tilde{u} \approx 1.27$ rad/s

(d) Amplitude of the spectrum of the cosine in 1(c)

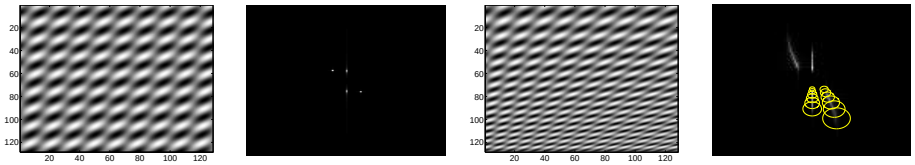


(e) 2D cosine at frequency $|\tilde{\mathbf{u}}| \approx 0.42$ rad/s

(f) Amplitude of the spectrum of the cosine in 1(e)

(g) 2D cosine. Frequency varies from $|\tilde{\mathbf{u}}| \approx 0.42$ rad/s to $|\tilde{\mathbf{u}}| \approx 1.27$ rad/s

(h) Spectrum amplitude of 1(g) and the chosen set of Gabor filters (circles) on it



(i) Two 2D cosines superposed, at frequencies $|\tilde{\mathbf{u}}_1| \approx 0.42$ rad/s and $|\tilde{\mathbf{u}}_2| \approx 0.63$ rad/s, 45° apart

(j) Amplitude of the spectrum of the image in 1(i)

(k) Result of slanting the image in 1(i) by 38°

(l) Spectrum amplitude of 1(k) and the chosen set of Gabor filters (circles) on it

Fig. 1. Setting the Gabor filters' parameters

than lower frequencies and is computationally less expensive. Moreover, as all filters so derived are geometrically similar it is simpler to compare their outputs.

We set the Gabor functions using the information from the Fourier transform of the texture. Unlike Super and Bovik ([12]), who sample the whole 2D frequency plane, we make a selection of Gabor filters using ridges in the Fourier transform of the image. In our algorithm every ridge determines a set of Gabor filters that covers the corresponding values of frequencies. Every ridge therefore determines different instantaneous frequencies and thus different distortion measures.

2.2 Setting the Gabor Filter Parameters

Let us consider a 1D cosine (figure 1(a)). The signal has length of 128 samples and frequency $\tilde{u} \approx 0.42$ rad/s (where π rad/s is by convention the biggest admissible

frequency). Figure 1(b) represents its spectrum amplitude, two symmetric spikes at the corresponding frequencies ($\approx \pm 0.42$ rad/s). A chirp is shown in figure 1(c), i.e. a cosine with frequency varying from $\tilde{u} \approx 0.42$ rad/s to $\tilde{u} \approx 1.27$ rad/s. Figure 1(d) illustrates its spectrum, where significant non-zero values span that range.

Analogously we show a 2D image generated by a 2D cosine with frequency $|\tilde{\mathbf{u}}| \approx 0.42$ rad/s (figure 1(e)) and its spectrum (figure 1(f)), given by symmetric spikes in the frequency plane. We then compare this to figures 1(g) and 1(h), the image of a 2D cosine with frequency ranging from $|\tilde{\mathbf{u}}| \approx 0.42$ rad/s to $|\tilde{\mathbf{u}}| \approx 1.27$ rad/s and its spectrum (circles are fully explained later). In the latter, significant non-zero values form a ridge corresponding to the range of frequencies. Figures 1(g) and 1(h) were actually generated by slanting (see section 3) image 1(e) through 38°. The ridges of the amplitude of the Fourier transform of the image represent the 2D frequencies contained in the texture.

The algorithm we propose analyzes the spectrum of the texture to determine its ridges, and then uses this information to define the sets of Gabor functions used. Figure 1(h) shows the chosen set of central frequencies $\mathbf{u}$ (the centers of the circles) and the set of spatial constants γ (their radii); half of the spectrum is considered because of its redundancy. There is significant overlapping (50%) to produce a robust LSF estimation. However, unlike in [12], where 63 central frequencies and spatial constants sample the whole 2D frequency plane, here the number used varies with the image. 7 $\mathbf{u}$'s and γ's are used in figure 1(h). This implies a significant reduction of the computational expense: in [12] 63 $\mathbf{u}$'s and γ's correspond to 378 convolutions (the Gabor filter and its first order derivatives and an equivalent number of post-smoothing filters); in this case 7 $\mathbf{u}$'s and γ's mean 42 convolutions, with a computational saving of about 89%.

We now consider multiple frequencies. Figure 1(i) shows a superposition of the cosine from the previous example ($|\tilde{\mathbf{u}}_1| \approx 0.42$ rad/s) on another with frequency $|\tilde{\mathbf{u}}_2| \approx 0.63$ rad/s, separated by 45° degrees from the first in the frequency plane. In this case the spectrum amplitude (figure 1(j)) has four peaks, corresponding to the values of the two frequencies of the cosines. In fact we can associate two instantaneous frequencies to each point, which coexist at every pixel. Figure 1(k) shows the result of applying the same slant as in 1(g): each cosine now has a continuously-varying frequency. Moreover the two LSFs change independently from each other, i.e. the first assumes the same values as in figure 1(h) and the second forms corresponding ridges in the direction of the slant. This is illustrated in figure 1(l), where the two ridges are functions of the original frequency and of the distortion due to the slant.

Our algorithm detects the two ridges and sets two groups of Gabor filters. In each group the central frequencies, $\mathbf{u}$, and the spatial constants, γ, are defined so that the filters cover the respective ridge area (figure 1(l)). Every set of filters is processed as in the previous example, i.e. as if the texture contained only one corresponding LSF. Thus each set of filters reconstructs an instantaneous frequency for each pixel. LSFs measure distortion and are combined to reconstruct the shape (see section 4). In this sense we exploit the multi-scale nature of the texture, because all different-scale frequencies are considered in the final result.

3 Projection of Texture

Here we describe the viewing geometry and a projection model, to provide a relationship between the surface texture and the image plane frequencies as a function of the shape of the surface. We then present a surface texture model.

3.1 Viewing Geometry and Projection Model

We adopt the viewing geometry and projection model of [12], and from that we take our motivation to extend the theory to general developable surfaces. They deal with the case of a planar surface and assume a pin-hole camera model. Their coordinate systems are given in figure 2. In this figure, the origin of the

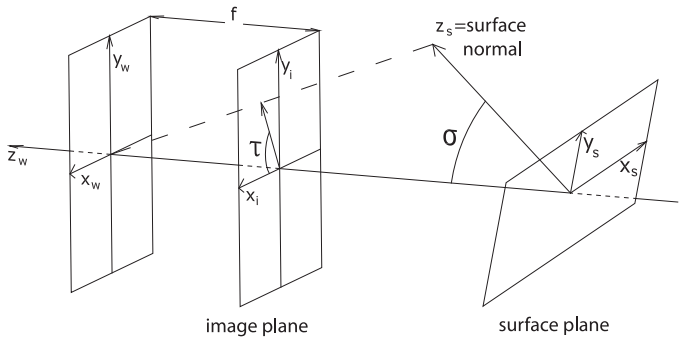


Fig. 2. Viewing geometry and projection model

world coordinate system $\mathbf{x}_w = (x_w, y_w, z_w)$ coincides with the focal point and the optical axis coincides with the $-z_w$-direction. The image plane coordinate system $\mathbf{x}_i = (x_i, y_i)$ is placed at $z_w = f < 0$, with $|f|$ being the focal length, such that $x_i = x_w$ and $y_i = y_w$. The orientation of the surface is described using the slant-tilt system: the slant σ is the angle between the surface normal and the optical axis, with values ranging from $0°$ to $90°$; the tilt τ is the angle between the x_i-axis and the projection on the image plane of the surface normal, with values between $-180°$ and $180°$. The surface is described by the coordinate system $\mathbf{x}_s = (x_s, y_s, z_s)$: the x_s-axis is aligned with the perceived tilt direction, the z_s-axis is aligned with the surface normal, y_s forms a right handed orthogonal coordinate system and the origin of $\mathbf{x}_s$ is on the intersection of the surface with the z_w-axis, at $z_w = z_0 < 0$.

[12] illustrates the equations for transforming 2D surface to 2D image coordinates, and vice versa, under perspective projection. Most importantly, they derive the relationship between the instantaneous frequencies on the image plane $\mathbf{u}_i = (u_i, v_i)$ and those on the surface plane $\mathbf{u}_s = (u_s, v_s)$:

$$\mathbf{u}_s = J^t(\mathbf{x}_i, \mathbf{x}_s) \cdot \mathbf{u}_i. \tag{4}$$

J^t, the transpose of the Jacobian of the transformation, is

$$J^t(\mathbf{x}_i) = \frac{\sin\sigma}{z_w}\begin{bmatrix} x_i & y_i \\ 0 & 0 \end{bmatrix} + \frac{f}{z_w}\begin{bmatrix} \cos\sigma\cos\tau & \cos\sigma\sin\tau \\ -\sin\tau & \cos\tau \end{bmatrix}, \tag{5}$$

where z_w is the corresponding coordinate, in the $\mathbf{x}_w$ coordinate system, of the surface point which projects to the image point $\mathbf{x}_i$.

The above described coordinate systems can be easily extended to general developable surfaces: assuming that the surface is smooth and that at any point it can be locally approximated with the corresponding tangent plane, the equations from [12] then apply to the tangent plane of the considered point.

Given the homogeneity assumption in section 1, frequencies $\mathbf{u}_s$ corresponding to the same LSF must be the same. However the $\mathbf{x}_s$ coordinate system changes from point to point on the surface texture according to the tilt-direction, because the x_s axis is aligned with it. In particular, between two image points $\mathbf{x}'_i$ and $\mathbf{x}''_i$ with measured frequencies $\mathbf{u}'_i$ and $\mathbf{u}''_i$, if the difference of tilt-direction of the corresponding points on the surface is $\Delta\tau = \tau'' - \tau'$, we have:

$$\mathbf{u}'_s = \mathbf{u}''_s \tag{6}$$

$$J^t(\mathbf{x}'_i)\cdot\mathbf{u}'_i = \begin{bmatrix} \cos\Delta\tau & \sin\Delta\tau \\ -\sin\Delta\tau & \cos\Delta\tau \end{bmatrix} J^t(\mathbf{x}''_i)\cdot\mathbf{u}''_i \tag{7}$$

where the matrix rotates the $\mathbf{x}_s$ coordinate systems by $\Delta\tau$ to align them.

Eq. 7 also applies in cases of orthographic projection: the relation between the surface tangent and the image coordinates in [12] reduces to

$$\mathbf{x}_i = \begin{bmatrix} \cos\sigma\cos\tau & -\sin\tau \\ \cos\sigma\sin\tau & \cos\tau \end{bmatrix}\mathbf{x}_s, \tag{8}$$

and the J^t in eq. 7 becomes $$J^t = \begin{bmatrix} \cos\sigma\cos\tau & \cos\sigma\sin\tau \\ -\sin\tau & \cos\tau \end{bmatrix}. \tag{9}$$

3.2 Surface Texture Model

We model textures as due to variations of surface reflectance, the proportion of reflected light. We assume surfaces with Lambertian reflection, and that the texture is therefore 'painted' on them, without roughness or self-occlusion.

Surface reflectance, $t_s(\mathbf{x}_s)$, and image reflectance, $t_i(\mathbf{x}_i)$, are related by

$$t_i(\mathbf{x}_i) = k(\mathbf{x}_i)\cdot t_s[\mathbf{x}_s(\mathbf{x}_i)], \tag{10}$$

where $\mathbf{x}_s(\mathbf{x}_i)$ represents the perspective backprojection, while $k(\mathbf{x}_i)$ is a multiplicative shading term. [11] shows how to estimate and remove k. However, if the scale of variation of t_s is small compared to the scale of variation of the shading term, then the latter can be assumed to be constant in any small neighborhood. Moreover, our method automatically normalizes for slow variations in illumination, shading and surface texture contrast, because it uses frequencies rather than amplitudes. Also no assumption is made about the textural nature of $t_s(\mathbf{x}_s)$, thus it might apply to various patterns, e.g. lines, blobs, etc.

4 Computing Surface Orientation

We explain here how our algorithm processes the image texture to produce the shape of the surface texture.

After the LSFs are computed as indicated in section 2, eq. 7 is used to estimate the corresponding σ's and τ's on the surface (the orientation parameters), results are then combined and used to reconstruct the shape.

In the orthographic case, substituting for J^t from eq. 9 in eq. 7 we get two equations in 4 unknowns: σ', τ', σ'', τ''. Assuming we know the orientation (σ', τ') of the tangent plane at point P', the surface texture point which projects to $\mathbf{x}'_i$, we can solve eq. 7 for σ'' and τ'' (the orientation parameters at point P'', respectively corresponding to $\mathbf{x}''_i$). The analytical solution for τ'' is

$$\tau'' - \tau' = \tan^{-1} \frac{u' \sin\tau' - v' \cos\tau' - u'' \sin\tau' + v'' \cos\tau'}{\cos\sigma'(u' \cos\tau' + v' \sin\tau') + u'' \cos\tau' + v'' \sin\tau'} = \Delta\tau, \quad (11)$$

which provides two values in the $[0, 2\pi]$ range, as for the well-known tilt-ambiguity:

$$\tau'' = \begin{cases} \tau' + \Delta\tau \\ \tau' + \Delta\tau + \pi \end{cases}. \quad (12)$$

σ'' is given by

$$\sigma'' = \cos^{-1}\left[\frac{\cos\sigma'(u' \cos\tau' + v' \sin\tau') + u'' \cos\tau' + v'' \sin\tau'}{\cos(\tau'' - \tau')(u'' \cos\tau'' + v'' \sin\tau'')} - 1\right]. \quad (13)$$

The perspective equations are obtained by substituting for J^t from eq. 5 in eq. 7. In this case, besides the σ's and τ's, z' and z'' are unknown, the z_w coordinates at the surface points P' and P''. As in the orthographic case, we assume z', σ' and τ' are known, i.e. the position and orientation of P' are assumed known. Furthermore z'' is inferred from the assumed orientation parameters, and then refined once the values of σ'' and τ'' are computed: in the first instance z'' is assigned a value as if P'' lied on the tangent plane at P' (its orientation is given by σ' and τ'), this produces two equations in two unknowns and enables us to compute σ'' and τ'', which are then used to refine z''. Experimental results show that assuming P'' initially on the tangent plane at P' gives accurate results for σ'' and τ'' if P' and P'' correspond to close points on the image plane. Our algorithm therefore uses a known point P to compute the orientation parameters of its neighboring points. Below is the closed form solution for τ''

$$\tau'' - \tau' = \tan^{-1}\left(-\frac{b}{a}\right) = \Delta\tau, \quad (14)$$

$$\text{where} \quad a = \frac{\sin\sigma'}{z'}(u'x' + v'y') + \frac{f}{z'}\cos\sigma'(u' \cos\tau' + v' \sin\tau') + \frac{f}{z''}(u'' \cos\tau' + v'' \sin\tau'), \quad (15)$$

$$b = \frac{f}{z'}(v' \cos\tau' - u' \sin\tau') + \frac{f}{z''}(u'' \sin\tau' - v'' \cos\tau'), \quad (16)$$

giving two solutions (eq. 12) corresponding to the well-known tilt-ambiguity. σ'' is then given by

$$\sigma'' = \sin^{-1} \frac{-c(b+d) \pm |d|\sqrt{c^2 - 2bd - b^2}}{c^2 + d^2}, \tag{17}$$

$$\text{where } b \text{ is as above and } c = \frac{1}{z''} \sin(\tau'' - \tau')(u''x'' + v''y''), \tag{18}$$

$$d = \frac{f}{z''} \sin(\tau'' - \tau')(u'' \cos\tau'' + v'' \sin\tau''). \tag{19}$$

The two sets of equations show how a single LSF can be used to compute the σ's and τ's, point after point, at all image points, and hence reconstruct the shape. However we choose to use the LSFs combined in pairs in order to get a more robust estimation: for each image point each of the two LSFs gives a solution (σ, τ), then the one that best approximates the equation defined by the other is chosen. Experimental results show that this produces more robust estimates than if finding a solution numerically by combining the equations for the two, or even more, LSFs. Note that this does not reduce the applicability of our method because most real textures have at least two LSFs.

For computing the shape, our algorithm needs a starting point with known orientation. As in [13], we consider the following alternatives: the orientation of a point can be manually input, or we can assume that the image displays a frontal point (for which $\sigma = 0$), which can be detected using the method of [6]. The latter easily integrates with our algorithm, because it uses the same Gabor filters which we adopted for computing the LSFs, and is therefore convenient because most of the computation is shared with the LSF estimation. As described in section 2.2, we do not use the whole lattice of filters, but we choose to separate and process only the relevant LSFs which we determine via the Fourier transform. In particular, the spectral second-order moments defined in [6] are here computed using two LSFs, as if the spectrum of the image contained only them:

$$F(\mathbf{u}) = \delta(\mathbf{u} - \mathbf{u}_1) + \delta(\mathbf{u} - \mathbf{u}_2). \tag{20}$$

The product of the canonical moments $\sqrt{mM}$ (see [6]) reduces therefore to

$$\sqrt{mM} = |u_1 v_2 - u_2 v_1|. \tag{21}$$

It can be shown that the above is the inverse of the area gradient (see [2]), computed using the two frequencies. As in [6], the minimum gives the frontal point ($\sigma = 0$), while the gradient direction gives the initial τ value.

With the method described, we get two reconstructed shapes for each pair of LSFs. By using eq. 4 and 7 and the determined σ's and τ's, we can backproject the frequencies $\mathbf{u}_i$ to the surface texture and align them. The homogeneous assumption states that the frequencies computed in this way should be the same and that their variance should therefore be zero. Hence we choose the shape associated with the lowest variance, assuming that lower values, closer to the ideal zero value, correspond to better estimates.

Below is the proposed algorithm:

- The spectrum amplitude of the image is analyzed and ridges are detected.
- Each ridge determines a set of Gabor functions and their first derivatives, so that the filters cover the frequencies pertaining to the particular ridge.
- For each set of filters the following steps are repeated:
  - the image is convolved with the Gabor filters and their derivatives, and the outputs are smoothed with a Gaussian to reduce noise;
  - the Gabor filter with largest amplitude output is selected at each point;
  - the instantaneous frequencies are computed at each point (eq. 3).
- For each pair of LSFs the orientation of a point is manually input or a frontal point is found (eq. 21).
- For each pair of LSFs a shape is reconstructed (eq. 7) and the variance of backprojected and aligned image frequencies is computed.
- The best shape is chosen according to the lowest variance.

We plan to use the backprojection of frequencies, and their theoretical equal values, in order to identify and discard possible outliers. Also, as many of the reconstructed shapes from the pairs of LSFs are accurate, future work might address combining these to produce better estimates.

Finally, the algorithm lends itself to parallel implementation, because ridges and filters can be processed independently and implemented by different units.

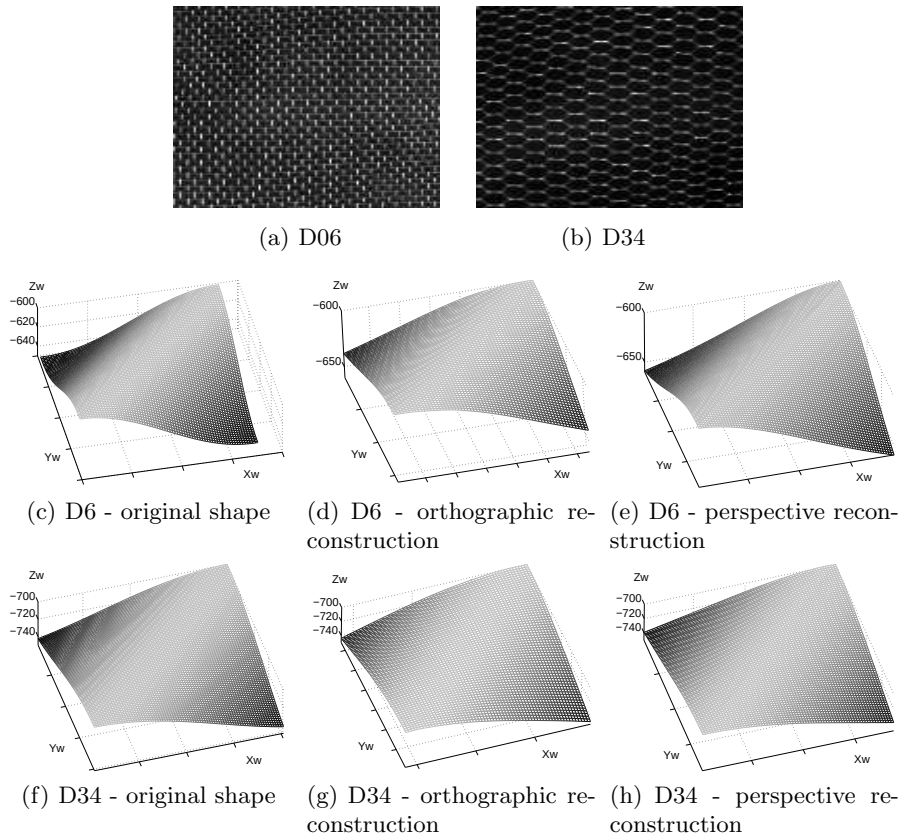
5 Results

We demonstrate our method on two sets of images. The first images (figures 3(a),(b)) were synthesized by mapping real textures from the Brodatz database ([15]) onto cosine surfaces (figures 3(c),(f)) and then rendering as a new image. The second (figures 4(a),(d),(g)) are real images acquired with a Pulnix TM-6EX camera. They are the central parts of 640x480 pictures of textured objects laid on a cylinder. Being real images, they are affected by variations in illumination, self shadowing (4(a)) and imperfections (4(g)). The images were not retouched before processing, and are 128x128 pixels with 256 levels of gray, with the exception of 4(g), which shows the case of a bigger extracted image (180x180).

Figures 3(d),(g) and 4(b),(e),(h) show the shapes reconstructed via the orthographic equations (9,7), while figures 3(e),(h) and 4(c),(f),(i) show those reconstructed via the perspective equations (5,7). In all cases the shapes were achieved

Table 1. Estimation errors for σ and τ for orthographic and perspective cases (degrees)

	Average error - Orthographic		**Average error - Perspective**	
Image	$\lvert\epsilon_\sigma\rvert$	$\lvert\epsilon_\tau\rvert$	$\lvert\epsilon_\sigma\rvert$	$\lvert\epsilon_\tau\rvert$
D6	5.24	3.10	1.37	3.10
D34	4.80	1.98	2.88	1.93
sponge	1.55	4.30	1.21	4.27
trousers	1.48	1.92	1.86	1.87
rubber rug	4.28	0.44	4.39	0.62



(a) D06 (b) D34

(c) D6 - original shape (d) D6 - orthographic reconstruction (e) D6 - perspective reconstruction

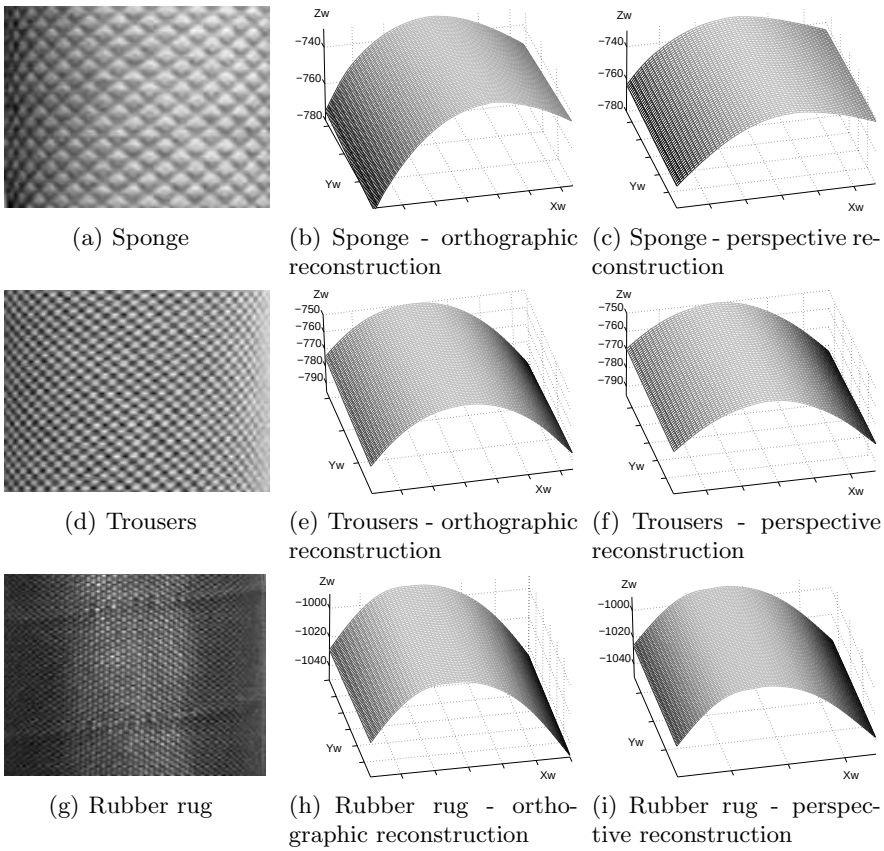
(f) D34 - original shape (g) D34 - orthographic reconstruction (h) D34 - perspective reconstruction

Fig. 3. Images synthesized from Brodatz textures, original shapes and reconstructions

by using the values of the orientation parameters (σ,τ), which were computed for all corresponding pixels of the original images. Although the estimates were not smoothed, the surfaces do look smooth and close to the original (or to part of a cylinder for real images). Table 1 shows the average errors of the estimated orientation parameters. In fact the ground truth was known for the first set of images, and it could be extracted from the images for the second set because the diameters of the cylinders and their distances from the camera were known. The errors confirm both the accuracy and robustness of our algorithm.

As stated in section 1, the homogeneity assumption requires some sort of periodicity/stationarity: the algorithm worked well up to cases when the periodicity of the texture was just one order of magnitude bigger than the main frequency associated with the shape.

Furthermore, we address the possibility that ridges might superimpose. This may be the case when a texture composed of close frequencies is slanted. The superposition can be spotted as it results in gaps in the frequency estimation.



(a) Sponge

(b) Sponge - orthographic reconstruction

(c) Sponge - perspective reconstruction

(d) Trousers

(e) Trousers - orthographic reconstruction

(f) Trousers - perspective reconstruction

(g) Rubber rug

(h) Rubber rug - orthographic reconstruction

(i) Rubber rug - perspective reconstruction

Fig. 4. Real images of textures and reconstruction

By considering smaller patches of the image (e.g. 96x96 instead of 128x128), the range of variation of frequencies analyzed by the Fourier transform is smaller and hence there is less chance of observing the superposition.

The pictures in this paper show the application of the algorithm to relatively simple developable shapes, where the tilt-ambiguity is solved by assuming convexity. However we plan to extend our simple and robust method to deal with general shapes, using a technique such as the EM algorithm to disambiguate the tilt. Finding a robust solution to shape from texture will open the way to possible applications such as the rendering and the retexturing of clothing ([10]).

6 Conclusions

The study presented here has characterized the variations of the dominant LSFs in textures via the ridges of their Fourier transforms, and used those to estimate

the shapes of developable and convex surface textures. Examples of reconstruction of both semi-synthetic and real images have been given, and errors have been computed in both cases. Our algorithm is accurate, simple to implement, and has the potential to be extended to general surfaces, by addressing non-developable surfaces and implementing an EM-type algorithm to solve the tilt-ambiguity.

To our knowledge, the proposed algorithm is the first to consider the multi-scale nature of texture to the extent of exploiting all main LSFs. Furthermore, it is robust against shading, variations in illuminations, and occlusions. Finally, it is based on the Fourier transform of the image and on a minimal number of convolutions, results are therefore computationally fast.

Acknowledgements

Fabio Galasso is grateful to St. John's college and Giovanni Aldobrandini for the 'A Luisa Aldobrandini' Scholarship, supporting his graduate studies.

References

1. Gibson, J.J.: The Perception of the Visual World. Houghton Mifflin, Boston, Massachusetts (1950)
2. Gårding, J.: Shape from texture for smooth curved surfaces in perspective projection. JMIV **2** (1992) 327–350
3. Hwang, W.L., Lu, C.S., Chung, P.C.: Shape from texture: Estimation of planar surface orientation through the ridge surfaces of continuous wavelet transform. IEEE Trans. Image Processing **7** (1998) 773–780
4. Kanatani, K., Chou, T.C.: Shape from texture: general principle. Artificial Intelligence **38** (1989) 1–48
5. Malik, J., Rosenholtz, R.: Computing local surface orientation and shape from texture for curved surfaces. IJCV **23** (1997) 149–168
6. Super, B.J., Bovik, A.C.: Shape from texture using local spectral moments. IEEE Trans. Patt. Anal. Mach. Intell. **17** (1995) 333–343
7. Clerc, M., Mallat, S.: Shape from texture through deformations. In: Int. Conf. Comp. Vision. (1999) 405–410
8. Loh, A.M., Hartley, R.: Shape from non-homogeneous, non-stationary, anisotropic, perspective texture. In: BMVC. (2005) 69–78
9. Witkin, A.P.: Recovering surface shape and orientation from texture. Artificial Intelligence **17** (1981) 17–45
10. Lobay, A., Forsyth, D.A.: Shape from texture without boundaries. IJCV **67** (2006) 71–91
11. Clerc, M., Mallat, S.: The texture gradient equation for recovering shape from texture. IEEE Trans. Patt. Anal. Mach. Intell. **24** (2002) 536–549
12. Super, B.J., Bovik, A.C.: Planar surface orientation from texture spatial frequencies. Pattern Recognition **28** (1995) 729–743
13. Loh, A.M., Kovesi, P.: Shape from texture without estimating transformations. Technical report, UWA-CSSE-05-001 (2005)
14. Bovik, A.C., Havlicek, J.P., Desai, M.D.: Theorems for discrete filtered modulated signals. In: ICASSP, Minneapolis, MN (1993) 805–810
15. Brodatz, P.: Textures: A Photographic Album for Artists and Designers. Dover, New York (1966)