Shape from Texture: Fast Estimation of Planar Surface Orientation via Fourier Analysis

Fabio Galasso and Joan Lasenby
Department of Engineering, University of Cambridge
Cambridge, CB2 1PZ, UK
fg257@cam.ac.uk, jl221@cam.ac.uk

Abstract

Shape from texture has received much attention in the past few decades. We propose a computationally efficient method to extract 3D planar surface orientation 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. Under perspective projection, frequencies are backprojected and orientation is computed so as to minimize the variance of the frequencies' backprojections. A comparative study with two existing methods, and experimentation on simulated and real texture images is given.

1 Introduction

Shape from Texture was first introduced by Gibson 50 years ago. In [7] 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 ([6], [8], [9], [11]). The model for the texture can be either deterministic or stochastic. The second allows a wider variety of textures ([9], [11], [13]) and implies local spectral measurements, usually with the Fourier transform ([11]), or more recently, wavelets ([8], [3]).

An initial assumption about the texture is always necessary, and few of the existing papers are applicable to real surfaces because of restrictive assumptions. [10] deals with texels, which are seldom found in nature, while [14] assumes isotropy, rarely the case. Homogeneity is more frequently used ([9], [6], [3]), 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 ([11]). Under this condition we assume that all texture variations are produced only by projective geometry.

We assume here a perspective or pin-hole camera model, as in [4] and [12], because perspective effects (e.g. shrinking) are usually found in images of slanted planes. We do not consider the weak perspective case as this preserves homogeneity and therefore gives no information on plane orientation ([5] and references within).

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 exploiting all main LSFs, most of the related work uses only two preferred directions in the spectral domain (e.g. [13]).

Section 2 explains in detail how the texture is analyzed to produce distortion information, and justifies the chosen method. Section 3 presents the projective geometry. Section 4 shows how we can recover surface 3D coordinates from the measured texture distortion. Finally, section 5 presents results, comparing them with those in [8].

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

The analysis of an image $I(\mathbf{x})$ is usually done using a band-pass filter $h(\mathbf{x},\mathbf{u})$, a function of a point $\mathbf{x} = (x,y)$ and of a central frequency $\mathbf{u} = (u,v)$, 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.

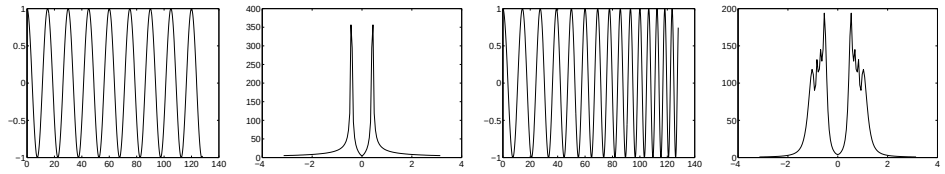
For a 2D cosine $f(\mathbf{x}) = \cos(2\pi\Omega(\mathbf{x}))$ the instantaneous frequency is given by

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

Our goal is to measure $\tilde{\mathbf{u}}(\mathbf{x})$. [1] 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 frequency we are measuring is in the pass-band of the filter. This method implies that we have to choose the central frequencies $\mathbf{u}$ of the Gabor functions, and the spatial constants γ, in order to set the centre and width of the filters. The filters have constant *fractional bandwidth* (bandwidth divided by its centre frequency). This allows us to measure higher frequencies more locally than lower frequencies and is computationally less expensive. Moreover, as all filters so derived are geometrically similar it is simpler to compare their outputs.

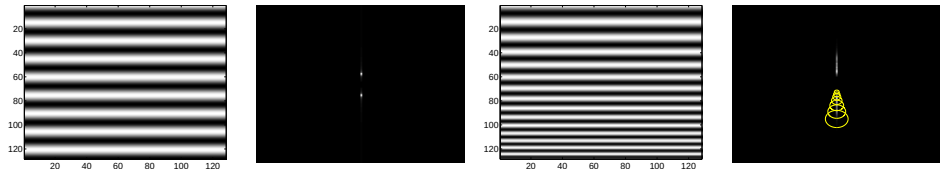


(a) 1D cosine at frequency $\tilde{u} \approx 0.42$ rad/s

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

(c) 1D cosine with frequency varying 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)

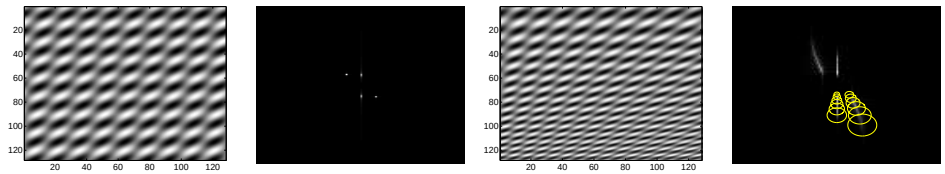


(e) 2D cosine at frequency $|\tilde{\mathbf{u}}| \approx 0.42$ rad/s

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

(g) 2D cosine with frequency continuously varying from $|\tilde{\mathbf{u}}| \approx 0.42$ rad/s to $|\tilde{\mathbf{u}}| \approx 1.27$ rad/s

(h) Amplitude of the spectrum 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° degrees apart

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

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

(l) Amplitude of the spectrum in 1(k) and the chosen set of Gabor filters (circles) on it

Figure 1: Setting the Gabor filters' parameters

We choose to 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 on the fre-

quency value. And we compare them 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 that range. 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 centres 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); our algorithm in this case uses 7 $\mathbf{u}$'s and γ's, meaning 42 convolutions, therefore a computational saving of about 89%.

We now consider the case of multiple frequencies. Figure 1(i) shows the cosine from the previous example ($|\tilde{\mathbf{u}}_1| \approx 0.42$ rad/s) on which we have superposed another cosine, with frequency $|\tilde{\mathbf{u}}_2| \approx 0.63$ rad/s, separated by 45° degrees from the first in the frequency plane. The amplitude of the spectrum of the image (figure 1(j)) shows four peaks, corresponding to the values of the two frequencies of the cosines. In this case we can associate two instantaneous frequencies to each point, which in fact coexist at every pixel. Figure 1(k) shows the result of applying the same slant as in figure 1(g): each cosine has now a continuously-varying frequency. Moreover the two LSFs change independently from each other. In fact the first cosine acquires the same continuously-varying frequency as in the previous section, and the second equivalently acquires a range of 2D frequencies varying in the direction of the slant. This is what the amplitude of the spectrum in figure 1(l) shows. In it we can observe two ridges, each of them associated with the original cosines, the spread indicating a variation or distortion due to the slant.

Our algorithm detects the two ridges and sets two groups of Gabor filters. In each group a series of values for the central frequencies, $\mathbf{u}$'s, and the spatial constants, γ's, are defined, so as to determine the filters to 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. These are used to measure the deformation of the texture due to the slanting, are processed independently and finally the results are combined (details are in 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 and the image plane as a function of the orientation. We then present a surface texture model.

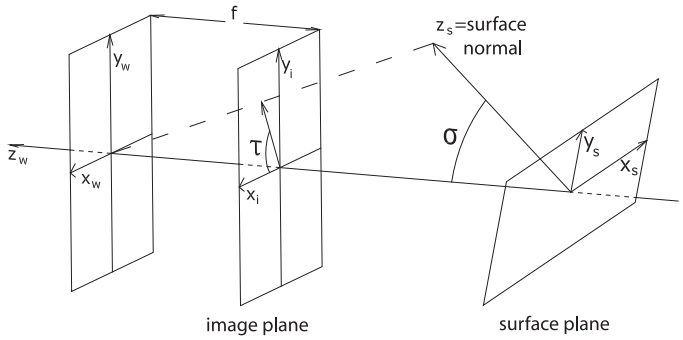


Figure 2: Viewing geometry and projection model

3.1 Viewing Geometry and Projection Model

We adopt the viewing geometry and projection model of [12]. They assume a pin-hole camera model and their coordinate systems are given in figure 2. In it the origin of the 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 $z = f < 0$, 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], to which we refer for details of the derivation, obtains 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 determinant of the coordinate transformation, is

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

$$\text{with} \quad z_w = z_0 - x_s \sin\sigma = \frac{f z_0 \cos\sigma}{\sin\sigma(x_i\cos\tau + y_i\sin\tau) + f\cos\sigma}. \tag{6}$$

We use the above to backproject a LSF computed on the image plane to the surface plane.

3.2 Surface Texture Model

We model textures as due to variations of surface reflectance, the proportion of incident light reflected. We assume that the surfaces have a 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 the following:

$$t_i(\mathbf{x}_i) = k(\mathbf{x}_i) \cdot t_s[\mathbf{x}_s(\mathbf{x}_i)], \tag{7}$$

where $\mathbf{x}_s(\mathbf{x}_i)$ represents the perspective backprojection, while $k(\mathbf{x}_i)$ is a multiplicative shading term. [4] 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

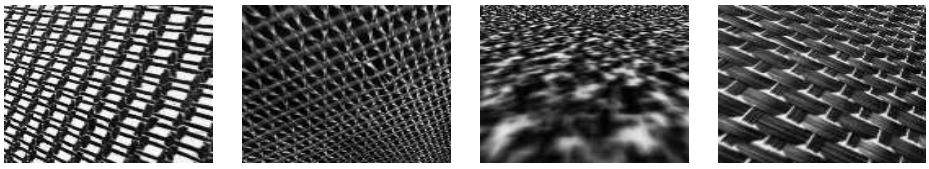
We explain here how our algorithm processes the image texture to produce the orientation of the surface texture.

As discussed in the introduction, we assume homogeneity, in the specific form that the relevant LSFs of the textured surface are constant in the surface region under analysis. Our assumption includes as a corollary that the variance of each LSF on the surface plane is zero. The theoretical zero value means a minimum in the case of real data, and this assumption is used to compute the surface orientation, i.e. the slant σ and tilt τ.

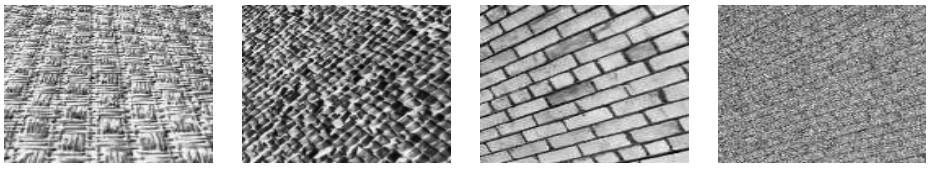
The structure of the proposed algorithm is therefore:

- The spectrum amplitude of the image texture 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 (signed) instantaneous frequencies are computed at each point (eq. 3);
  - a 2D search over the plane σ-τ is implemented: for each couple (σ,τ) the instantaneous frequency is backprojected using equation 4, and the variance $V_{\sigma,\tau}$ is computed;
  - the values of σ and τ corresponding to the minimum variance are chosen, and the variance is also returned.
- The algorithm chooses the best couple (σ,τ) as that giving the lowest variance.

The minimum variance ($V_{\sigma,\tau}$) method requires the estimated instantaneous frequencies to pertain to the same slanted and tilted LSF in every group. This is not assured if we use a grid of Gabor filters and choose the largest amplitude output, as in [12]. In this case, maximum outputs might then correspond to different groups of LSFs for different pixels in textures with more than one dominant frequency, which invalidates the orientation estimation. Our algorithm allows us to estimate instantaneous frequencies pertaining to distinct groups because it uses separated sets of filters. This improves its robustness.



(a) D20 (b) D52 (c) D57 (d) D65



(e) D82 (f) D84 (g) D95 (h) D95 with white Gaussian noise (SNR=−5dB)

Figure 3: Images synthesized from Brodatz textures

Image	**True** τ	τ_{GL}	$\lvert\tau_{GL}-\tau\rvert$	τ_{HLC}	$\lvert\tau_{HLC}-\tau\rvert$	**True** σ	σ_{GL}	$\lvert\sigma_{GL}-\sigma\rvert$	σ_{HLC}	$\lvert\sigma_{HLC}-\sigma\rvert$
D20	160	160.45	0.45	160.05	0.05	37	37.36	0.36	37.53	0.53
D52	-60	-60.28	0.28	-61.08	1.08	50	49.55	0.45	49.58	0.42
D57	90	91.00	1.00	91.41	1.41	70	70.37	0.37	67.30	2.70
D65	60	59.48	0.52	55.00	5.00	50	48.71	1.29	54.38	4.38
D82	90	89.63	0.37	90.71	0.71	50	49.16	0.84	48.62	1.38
D84	135	134.16	0.84	128.58	6.42	35	35.32	0.32	33.26	1.74
D95	-155	-154.92	0.08	-158.57	3.57	27	25.89	1.11	28.45	1.45

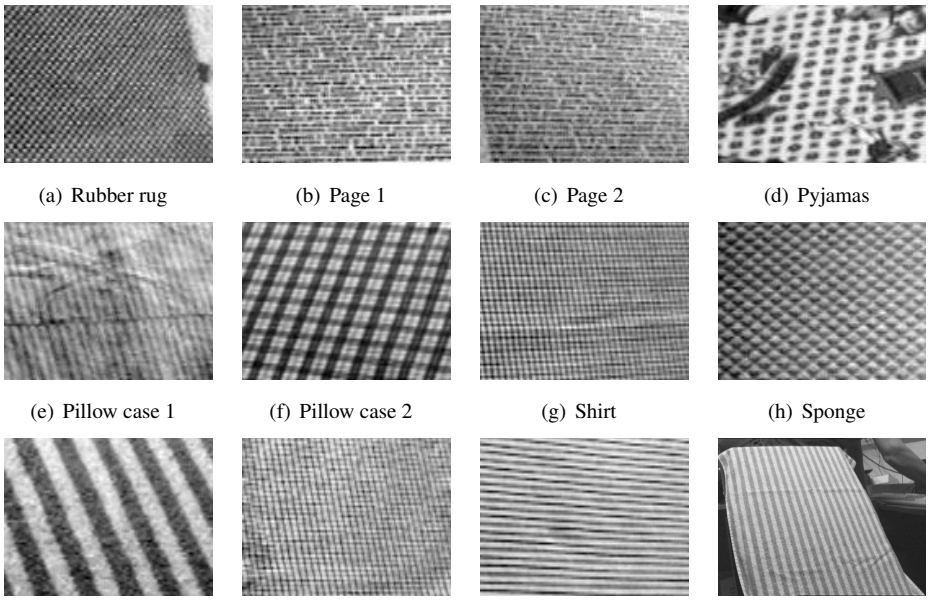
Table 1: Tilt and slant results of our method (τ_{GL}, σ_{GL}) on images synthesized from Brodatz database textures, compared to the results of [8] (τ_{HLC}, σ_{HLC}) (angles in degrees)

All the relevant frequencies are used. Eventually, we choose the pair (σ,τ) with the lowest $V_{\sigma,\tau}$ as we assume that lower values of residual variance, closer to the ideal zero value, correspond to better orientation estimates. As results from all ridges are accurate, future work might address combining these to produce better estimates.

Finally, the algorithm lends itself well to parallel implementations, because each ridge and filter can be processed independently and implemented by different units.

5 Results

We demonstrate our method on two sets of images. The first (figures 3(a)-(g)) is derived from [8], whose results we use for comparison. The images in this set were synthesized by mapping real textures from the Brodatz database ([2]) onto an inclined surface and then rendering it as a new image. Table 1 shows the results achieved compared with those from [8]. Our average estimation errors for τ and σ are 0.51° and 0.68° respectively, while Hwang et al. ([8]) achieve corresponding values of 2.6° and 1.8°. The accuracy of our method is significantly higher. As in [8], we add various levels of white Gaussian noise (SNR ranging from 20 to -5 dB) to the images of the textures D20, D52, D82, D95



(a) Rubber rug (b) Page 1 (c) Page 2 (d) Pyjamas

(e) Pillow case 1 (f) Pillow case 2 (g) Shirt (h) Sponge

(i) Towel (j) Trousers (k) T-shirt (l) Towel in 4(i) - Whole picture

Figure 4: Real images of texture planes

(the latter with SNR=-5dB is shown in figure 3(h)). Note that our estimates are always closer to the noiseless result than those of [8], thus indicating increased robustness.

The second set (figures 4(a)-(k)) consists of real images. All of them are the central 128x128 parts of 640x480 pictures. Figure 4(l) presents the whole image from which figure 4(i) was derived. As can be seen, the textured object was laid flat on a panel of known orientation (obtained using a multiple camera system prior to the experiment) and photographed with a Pulnix TM-6EX camera. The chosen textured objects were mainly fabrics, but also included some different materials. It is clear that the pictures are affected by variations in illumination and self shadowing (4(h)), creases (4(e)), imperfections (4(b), 4(a)) and occlusions (4(d)). Table 3 shows the results we obtained, compared to the ground truth. On average, tilt and slant were estimated with an error of 1.3° and 1.5°

	SNR (dB)				
Image (τ/σ)	∞	20	10	0	-5
D20 (160/37)	160.4/37.4	159.5/37.1	159.6/36.8	159.6/37.3	157.3/37.3
D52 (-60/50)	-60.3/49.5	-58.7/47.6	-64.4/49	-67.2/46.6	-61.6/34.3
D82 (90/50)	89.6/49.2	-89.1/51.3	90.6/52.3	86.1/45.4	X
D95 (-155/27)	-154.9/25.9	-158.8/25.2	-159.8/26.2	-160.8/25.4	-160.6/24.5

Table 2: Surface orientations (τ/σ) estimated using our method on noisy images - true values are in parenthesis (X indicates that the results were not reliable) (angles in degrees)

Image	**True** τ	τ_{GL}	$\lvert\tau_{GL}-\tau\rvert$	**True** σ	σ_{GL}	$\lvert\sigma_{GL}-\sigma\rvert$
rubber	118.8	118.3	0.5	35.3	33.4	1.9
page1	-152.8	-153.3	0.5	23.6	23.4	0.2
page2	123.6	121.3	2.3	36.9	34.0	2.9
pyjamas	-152.8	-151.2	1.6	23.6	20.2	3.4
p'case1	-123.6	123.2	0.4	36.9	34.4	2.5
p'case2	-146.5	-147.7	1.2	32.7	33.6	0.9
shirt	103.2	107.5	4.3	33.6	31.2	2.4
sponge	-158.3	-157.9	0.4	25.5	25.1	0.4
towel	146.4	146.2	0.2	38.8	39.9	1.1
trousers	118.8	118.7	0.1	35.3	35.2	0.1
T-shirt	123.6	121.3	2.3	36.9	35.7	1.2

Table 3: Tilt and slant results of our method (τ_{GL},σ_{GL}) on real images (angles in degrees)

respectively. These data confirm both the accuracy and the robustness of our algorithm.

All processed images were 128x128 pixels with 256 levels of gray. The backprojection of the computed LSFs for each value (σ,τ) was done just for the middle section of the image (here 64x64), so as to avoid edge effects. The constant fractional bandwidth is one half, and the space constant of the post-smoothing Gaussian filter is 1/12 of the image. We could not apply our method to those images in [12] because we could not gather all the data of the original setup. Processing the 18 images, 46 ridges of the Fourier transform were detected, that determined 232 Gabor functions. On average the number of convolutions per image was therefore 77.33. Compared to [12], where 378 convolutions per image are used, we save 79.54% in computational power.

As stated in section 1, the homogeneity assumption requires some sort of periodicity/stationarity: the algorithm can deal with as little as 6 cycles/picture.

Finally, we address the possibility that ridges might superimpose. This may be the case when a texture composed of close frequencies is slanted. Such a superposition can easily be spotted by our algorithm, as it results in gaps in the frequency estimation. We solve it by considering a smaller patch of the image, e.g. 96x96 instead of 128x128. In this way the range of variation of frequencies analyzed by the Fourier transform is smaller and hence there is less chance of observing the superposition.

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 orientation of surface textures. Numerical results have been given on both semi-synthetic and real images and compared where possible with other work. Our algorithm is more accurate, simple to implement, and has the potential to be extended to more complex surface shapes.

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, and performs well in the presence of added Gaussian noise. Finally, it is based on the Fourier transform of the image and on a

minimal number of convolutions, results are therefore computationally fast.

Acknowledgements

The authors thank Dr. B. Super, and Dr W.-L. Hwang, Dr C.-S. Lu, Dr P.-C. Chung for providing texture images. Fabio Galasso is grateful to St.John's college and Giovanni Aldobrandini for the 'A Luisa Aldobrandini' Scholarship, supporting his graduate studies.

References

[1] Alan C. Bovik, Joseph P. Havlicek, and Mita D. Desai. Theorems for discrete filtered modulated signals. In *ICASSP*, pages 805–810, Minneapolis, MN, 1993.

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

[3] Maureen Clerc and Stéphane Mallat. Shape from texture through deformations. In *Int. Conf. Comp. Vision*, pages 405–410, 1999.

[4] Maureen Clerc and Stéphane Mallat. The texture gradient equation for recovering shape from texture. *IEEE Trans. Patt. Anal. Mach. Intell.*, 24(4):536–549, 2002.

[5] Antonio Criminisi and Andrew Zisserman. Shape from texture: Homogeneity revisited. In *BMVC*, pages 82–91, 2000.

[6] Jonas Gårding. Shape from texture for smooth curved surfaces in perspective projection. *JMIV*, 2:327–350, 1992.

[7] James J. Gibson. *The Perception of the Visual World*. Houghton Mifflin, Boston, Massachusetts, 1950.

[8] Wen-Liang Hwang, Chun-Shien Lu, and Pau-Choo Chung. Shape from texture: Estimation of planar surface orientation through the ridge surfaces of continuous wavelet transform. *IEEE Trans. Image Processing*, 7(5):773–780, May 1998.

[9] Kenichi Kanatani and Tsai-Chia Chou. Shape from texture: general principle. *Artificial Intelligence*, 38(1):1–48, 1989.

[10] Angeline M. Loh and Richard Hartley. Shape from non-homogeneous, non-stationary, anisotropic, perspective texture. In *BMVC*, pages 69–78, 2005.

[11] Jitendra Malik and Ruth Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *IJCV*, 23(2):149–168, 1997.

[12] Boaz J. Super and Alan C. Bovik. Planar surface orientation from texture spatial frequencies. *Pattern Recognition*, 28(5):729–743, 1995.

[13] Boaz J. Super and Alan C. Bovik. Shape from texture using local spectral moments. *IEEE Trans. Patt. Anal. Mach. Intell.*, 17(4):333–343, 1995.

[14] Andrew P. Witkin. Recovering surface shape and orientation from texture. *Artificial Intelligence*, 17:17–45, 1981.