Shape from Texture via Fourier Analysis

Fabio Galasso and Joan Lasenby

University of Cambridge, Cambridge, UK

Abstract. Many models and algorithms have been proposed since the shape from texture problem was tackled by the pioneering work of Gibson in 1950. In the present work, a general assumption of stochastic homogeneity is chosen so as to include a wide range of natural textures. Under this assumption, the Fourier transform of the image and a minimal set of Gabor filters are used to efficiently estimate all the main local spatial frequencies of the texture, i.e. so as to compute distortion measures. Then a known method which uses singular value decomposition to process the frequencies under orthographic projection is considered. The method is extended to general perspective cases and used to reconstruct the 3D shape of real pictures and video sequences. The robustness of the algorithm is proven on general shapes, and results are compared with the literature when possible.

1 Introduction

Shape from texture dates back to nearly 60 years ago, when Gibson introduced the visual effects of texture cues in the 3D perception of the human visual system ([1]). Since then many authors have contributed to the study of the shape from texture problem with different texture and shape assumptions (e.g. [2], [3], [4], [5], [6], [7], [8], [9], [10]). Our texture model is stochastic, as in [2], [5], [4], [6] and [10], so as to allow a wider variety of textures than a deterministic model. This implies the use of local spectral measurements achieved with the Fourier transform and with Gabor filters. Our initial assumption is homogeneity. It is frequently used (e.g. [2], [3], [10]) as it results more applicable to real textures than other more restrictive ones, like texels (e.g. [9]), seldom found in nature, or isotropy (e.g. [11]), rarely the case. Homogeneity can be seen as periodicity for deterministic textures, and is formalized as stationarity under translation for stochastic textures ([5]). Under this condition we assume that all texture variations are produced only by projective geometry.

Shape from texture is generally about measuring the texture distortion in an image and then reconstructing the surface 3D coordinates in the scene ([3], [2], [5], [10]). The present work shows how the method of distortion estimation based on *local spatial frequency* (LSF) introduced by Galasso and Lasenby and presented in [12] can easily be integrated with the reconstruction method based on *singular value decomposition* (SVD) of Dugué and Elghadi introduced in [6]. The former is the first to use the multi-scale nature of texture, whereas most of the related work uses only two preferred directions in the spectral domain (e.g.

[4],[13]). The latter is a robust method for automatic 3D reconstruction from LSFs which only assumes that the image contains a frontal point. While the method in [6] only considers orthographic projections, here we also extend the method to deal with perspective projections. Moreover we compare some of the achieved results with those in [12], and we illustrate how the method proposed here is more robust, can effectively use redundancy to improve estimates, and is applicable to general non-developable surfaces. Finally application of the method is shown on a real video sequence, evidencing the sensitivity of the new algorithm to small changes in the shape of a texture object, which could open the way to using shape from texture as a non-invasive shape variation analysis.

The article is structured as follows: section 2 explains 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 on real pictures.

2 Texture Description

Here we describe how to compute the instantaneous frequencies for the different LSFs in the texture, with the use of Fourier transform and minimal sets of 2D Gabor functions. LSFs provide the distortion measures, from which the 3D coordinates of the texture surface are then reconstructed.

2.1 Estimating the Instantaneous Frequencies

We can analyze an image $I(\mathbf{x})$ using a pass-band 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 [14] and [12] we choose 2D Gabor functions:

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

These functions satisfy exactly the lower bound for the measure of a frequency $\mathbf{u}$ and its position $\mathbf{x}$, given by the uncertainty principle ([15]). Furthermore the functions are geometrically similar and can be easily arranged to form banks of filters.

Our goal is to estimate the instantaneous frequencies $\mathbf{u}$ of the image points. [14], [12] and references therein show 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 $h_y(\mathbf{x}, \mathbf{u})$:

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

The estimate, $\tilde{\mathbf{u}} = (\tilde{u}, \tilde{v})$, can be assumed to be correct if the measured frequency is in the pass-band of the filter. The method implies therefore that we set the

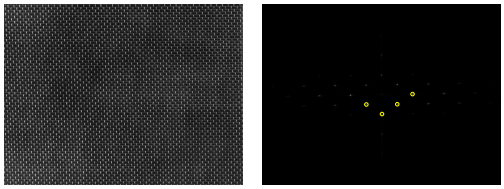
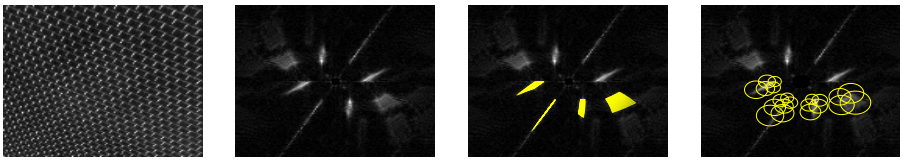


Fig. 1. Brodatz texture D6 (*left*) and its spectrum amplitude (*right*)

central frequencies $\mathbf{u}$ and the spatial constants γ of the Gabor functions, i.e. their centers and width, to relevant values.

Unlike Super and Bovik ([14]), who sample the whole 2D frequency plane, we choose to use the method of Galasso and Lasenby ([12]), i.e. we define small sets of Gabor functions from the information provided by the Fourier transform of the image.

2.2 Setting the Gabor Filter Parameters



(a) (b) (c) (d)

Fig. 2. (a) Brodatz texture D6 rendered on a plane; (b) its spectrum amplitude; (c) real LSFs (*yellow areas*) represented on the spectrum; (d) chosen Gabor functions (*yellow circles*) superimposed on the spectrum

We refer to [12] for a detailed explanation of how the amplitude of the Fourier transform represents the 2D frequencies of an image. Here we consider the Brodatz texture D6 and its spectrum amplitude in figure 1. From the spectrum we deduce that the texture is mainly characterized by four LSFs, marked with small yellow circles (only half of the spectrum need to be considered due to symmetry). By slanting and rotating the texture (fig. 2(a)), the four dots of figure 1 become corresponding areas in the spectrum (fig. 2(b)). Figure 2(c) represents the instantaneous frequencies of each of the image pixels for the four LSFs (yellow areas) on the spectrum (as the image is synthesized, the ground truth, i.e. the values of each LSF at each pixel, is known and marked on the spectrum). It is clear from this that non-zero values represent corresponding frequencies of the images, and that contiguous areas correspond to the same LSF. In our algorithm

each area is used to set a distinct group of Gabor functions (yellow circles in fig. 2(d)), i.e. to set distinct sets of central frequencies $\mathbf{u}$ (centers of the circles) and spatial constants γ (their radii).

A significant overlapping of the filters increases the robustness of the estimation. Furthermore the number of filters (and consequently convolutions) varies with the complexity of the image, but fewer are necessary than if sampling the whole spectrum (as in [14]), resulting in a significant reduction in computational expense.

Finally the method exploits the multi-scale nature of the texture, because all different-scale frequencies are estimated and used in the shape reconstruction.

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.

Figure 3 illustrates the viewing geometry of [14]. $\mathbf{x}_w = (x_w, y_w, z_w)$ is the

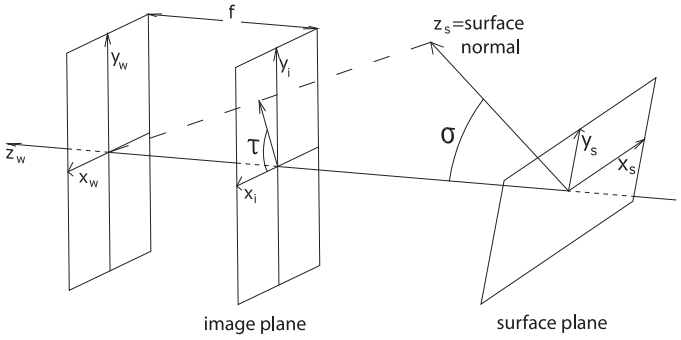


Fig. 3. Viewing geometry and projection model of [14]

world coordinate system and $\mathbf{x}_i = (x_i, y_i)$ is the image plane coordinate system, placed at $z_w = f < 0$, with $|f|$ being the focal length. 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; the tilt τ is the angle between the x_i-axis and the projection on the image plane of the surface normal. 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.

The above coordinate systems can be easily extended to general 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 [14] then apply to the tangent plane at the considered point.

Given the homogeneity assumption in section 1, surface texture frequencies $\mathbf{u}_s$ corresponding to the same LSF must be the same all over the surface texture. However the $\mathbf{x}_s$ coordinate system changes with the tilt-direction, as the x_s axis is aligned with it at each point. For aligning the $\mathbf{x}_s$ coordinate system we counter-rotate it by the angle $-\tau$ on the tangent plane, so as to have $\mathbf{x}_s$ oriented along the same direction in a virtual unfolded surface texture.

This system of coordinates was first introduced by Dugué and Elghadi [6], who illustrate the relationship between $\mathbf{x}_i$ and $\mathbf{x}_s$, and between the frequencies on the image and surface texture, $\mathbf{u}_i$ and $\mathbf{u}_s$, in the orthographic case. Here we generalize the orthographic relationships to the general perspective case.

The transformation matrix between the aligned $\mathbf{x}_s$ and the $\mathbf{x}_w$ coordinate system is

$$\mathbf{x}_w = \begin{bmatrix} \cos\sigma\cos\tau & -\sin\tau & \sin\sigma\cos\tau \\ \cos\sigma\sin\tau & \cos\tau & \sin\sigma\sin\tau \\ -\sin\sigma & 0 & \cos\sigma \end{bmatrix} \begin{bmatrix} \cos\tau & \sin\tau & 0 \\ -\sin\tau & \cos\tau & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}_s + \begin{bmatrix} 0 \\ 0 \\ z_0 \end{bmatrix} , \tag{3}$$

which we combine with the perspective projection equation between $\mathbf{x}_w$ and $\mathbf{x}_i$

$$\mathbf{x}_i = \frac{f}{z_w} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{x}_w , \tag{4}$$

to obtain the relation between the image plane $\mathbf{x}_i$ and the surface texture $\mathbf{x}_s$ coordinates (note that, from here on, we will take $\mathbf{x}_s$ to be (x_s, y_s), since on the surface texture $z_s = 0$):

$$\mathbf{x}_i = \frac{f}{z_w} R(\tau)P(\sigma)R(-\tau)\mathbf{x}_s , \quad \text{where} \quad z_w = z_0 - \sin\sigma(x_s\cos\tau + y_s\sin\tau) , \tag{5}$$

$$R(\tau) = \begin{bmatrix} \cos\tau & -\sin\tau \\ \sin\tau & \cos\tau \end{bmatrix} \quad \text{and} \quad P(\sigma) = \begin{bmatrix} \cos\sigma & 0 \\ 0 & 1 \end{bmatrix} . \tag{6}$$

$R(\tau)$ and $P(\sigma)$ are respectively responsible for rotating the system of coordinates of τ and foreshortening the distances parallel to the tilt direction by a coefficient $\cos\sigma$, z_0 is the z_w coordinate of the image point being considered.

As in [14], we assume that the image intensity $I(\mathbf{x}_i)$ is proportional to the surface reflectance $I_s(\mathbf{x}_s(\mathbf{x}_i))$, and that shading is negligible because it is assumed to vary slowly compared to I_s. Also, as in [14], we compute the relationship between the frequencies on the image plane $\mathbf{u}_i = (u_i, v_i)$ and on the surface texture $\mathbf{u}_s = (u_s, v_s)$ by using the transpose of the Jacobian of the vector function $\mathbf{x}_i(\mathbf{x}_s)$ in equation 5:

$$\mathbf{u}_s = J^t(\mathbf{x}_i, \mathbf{x}_s)\mathbf{u}_i \tag{7}$$

$$\text{with} \quad J^t(\mathbf{x}_i, \mathbf{x}_s) = \tfrac{\sin\sigma}{z_w} \begin{bmatrix} x_i\cos\tau & y_i\cos\tau \\ x_i\sin\tau & y_i\sin\tau \end{bmatrix} + \tfrac{f}{z_w} R(\tau)P(\sigma)R(-\tau) , \tag{8}$$

where z_w is the corresponding coordinate of the surface point which projects to the image point $\mathbf{x}_i$. In the orthographic case, as shown by [6], the Jacobian reduces to

$$J^t(\mathbf{x}_i, \mathbf{x}_s) = R(\tau)P(\sigma)R(-\tau) . \tag{9}$$

The homogeneity assumption can therefore be written as:

$$\mathbf{u}_{s1} = \mathbf{u}_{s2} \tag{10}$$

$$J^t(\mathbf{x}_{i1}, \mathbf{x}_{s1})\mathbf{u}_{i1} = J^t(\mathbf{x}_{i2}, \mathbf{x}_{s2})\mathbf{u}_{i2} \tag{11}$$

where $\mathbf{u}_{s1}$ and $\mathbf{u}_{s2}$, the surface texture frequencies at the points $\mathbf{x}_{s1}$ and $\mathbf{x}_{s2}$, are the back-projections of $\mathbf{u}_{i1}$ and $\mathbf{u}_{i2}$, the image frequencies belonging to the same LSF measured at two distinct image plane points $\mathbf{x}_{i1}$ and $\mathbf{x}_{i2}$.

4 Computing Surface Orientation

We explain here how our algorithm processes the LSFs computed in section 2 to produce the shape of the surface texture.

Let us consider equation 7 for two different LSFs $\mathbf{u}'_i = (u'_i, v'_i)$ and $\mathbf{u}''_i = (u''_i, v''_i)$ measured at the same image point $\mathbf{x}_i$:

$$\mathbf{u}'_s = J^t(\mathbf{x}_i, \mathbf{x}_s)\mathbf{u}'_i \quad \text{and} \quad \mathbf{u}''_s = J^t(\mathbf{x}_i, \mathbf{x}_s)\mathbf{u}''_i \; . \tag{12}$$

The above equations can be combined to write

$$U_s = J^t(\mathbf{x}_i, \mathbf{x}_s)U_i \quad \text{where} \quad U_s = \begin{bmatrix} u'_s & u''_s \\ v'_s & v''_s \end{bmatrix} \quad \text{and} \quad U_i = \begin{bmatrix} u'_i & u''_i \\ v'_i & v''_i \end{bmatrix} \; . \tag{13}$$

Here the U_s contains the surface texture frequencies relative to the two LSFs, the same all over the surface according to the homogeneity assumption, while U_i contains the image frequencies for the two LSFs, which are functions of the particular point.

In the orthographic case we substitute for $J^t(\mathbf{x}_i, \mathbf{x}_s)$ from 9 in the above equation:

$$U_i U_s^{-1} = R(\tau)P^{-1}(\sigma)R(-\tau) = \begin{bmatrix} \cos\tau & -\sin\tau \\ \sin\tau & \cos\tau \end{bmatrix} \begin{bmatrix} \frac{1}{\cos\sigma} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\tau & \sin\tau \\ -\sin\tau & \cos\tau \end{bmatrix} \; . \tag{14}$$

The above system of four equations in the two unknowns (σ,τ) holds for each pixel of the image. In the particular case of a frontal point ($\sigma = 0$), i.e. when the orientation of the tangent plane is parallel to the image plane, we have

$$U_s = U_i|_{\sigma=0} \; . \tag{15}$$

In fact U_s will be computed from the U_i estimated at a frontal point, or from equation 14 at a point of known orientation (see below for discussion on the frontal point).

Since R is orthonormal and P^{-1} is diagonal, equation 14 can be compared with the SVD of the left hand side of the equation, according to which we write

$$U_i U_s^{-1} = VDV^{-1} = [\mathbf{v}_M \mathbf{v}_m] \begin{bmatrix} \lambda_M & 0 \\ 0 & \lambda_m \end{bmatrix} [\mathbf{v}_M \mathbf{v}_m]^{-1} \; , \tag{16}$$

with V the orthogonal matrix of the eigenvectors ($\mathbf{v}_M$, $\mathbf{v}_m$) of $U_i U_s^{-1}$, and D the diagonal matrix of the corresponding eigenvalues ($|\lambda_M| > |\lambda_m|$). Recalling equation 14 we have

$$\sigma = \arccos\left(\frac{1}{\lambda_M}\right) \tag{17}$$

$$\tau = \angle(\mathbf{v}_M)\ . \tag{18}$$

In other words σ is computed from the bigger eigenvalue while τ is oriented in the direction of the corresponding eigenvector. This is also approximately valid when the surface is not completely unfoldable, in which case the term D_{22} of the diagonal matrix is not exactly 1, as explained in [6].

In the perspective case we substitute for $J^t(\mathbf{x}_i, \mathbf{x}_s)$ from equation 8 in equation 13:

$$\frac{f}{z_w}\left(I - Y(\sigma,\tau,z_w)\right)^{-1} U_i U_s^{-1} = \begin{bmatrix}\cos\tau & -\sin\tau \\ \sin\tau & \cos\tau\end{bmatrix}\begin{bmatrix}\frac{1}{\cos\sigma} & 0 \\ 0 & 1\end{bmatrix}\begin{bmatrix}\cos\tau & \sin\tau \\ -\sin\tau & \cos\tau\end{bmatrix} \tag{19}$$

$$\text{where}\quad Y(\sigma,\tau,z_w) = \frac{\sin\sigma}{\sin\sigma(x_i\cos\tau + y_i\sin\tau) + f\cos\sigma}\begin{bmatrix}x_i\cos\tau & y_i\cos\tau \\ x_i\sin\tau & y_i\sin\tau\end{bmatrix} \tag{20}$$

Again the matrix U_s is computed from the U_i frequencies of a frontal point:

$$U_s = \frac{z_0}{f} U_i|_{\sigma=0}\ , \tag{21}$$

where f is the focal length and z_0 is the z_w coordinate of the frontal point.

As in the orthographic case, assuming we know the value of the left hand side, we can decompose using the SVD:

$$\frac{f}{z_w}\left(I - Y(\sigma,\tau,z_w)\right)^{-1} U_i U_s^{-1} = VDV^{-1} = [\mathbf{v}_M\mathbf{v}_m]\begin{bmatrix}\lambda_M & 0 \\ 0 & \lambda_m\end{bmatrix}[\mathbf{v}_M\mathbf{v}_m]^{-1}\ , \tag{22}$$

from which we compute (σ,τ) using equations 17 and 18. However, since Y actually depends on (σ, τ, z_w), we propose an iterative solution:

- the slant σ and tilt τ are estimated orthographically;
- the computed σ's and τ's are integrated to reconstruct the 3D shape, i.e. the z_w is estimated with the method of Frankot and Chellappa ([16]);
- the matrix $Y(\sigma, \tau, z_w)$ and the left hand side of equation 19 are computed;
- σ's and τ's are updated from the SVD and the procedure iterates until convergence.

Convergence of the algorithm has been verified empirically, furthermore we observe that matrix Y is usually small with respect to the unitary matrix I. In fact, a few iterations are usually enough to reach convergence, and this is achieved even when initializing the algorithm with the wrong (τ,σ,z_w) estimate.

The procedure needs a frontal point or a point with known orientation. As in [13], we can either manually input the orientation of a point (hence U_s is

computed from equations 14 or 19), or assume that the image displays a frontal point (in which case U_s is computed using equations 15 or 21).

The presence and position of a frontal point can be detected using the method of [4], which easily integrates with our algorithm. As shown in [12], we only consider two LSFs to write down the product of the canonical moments $\sqrt{mM}$ of [4] as

$$\sqrt{mM} = |u_1 v_2 - u_2 v_1|. \tag{23}$$

It can be shown that the above is the inverse of the area gradient (see [3]), computed using the two frequencies. Its minimum gives the frontal point.

With the method described, we get a reconstructed shape for each pair of LSFs. By using equation 7 and the determined (σ,τ,z_w), we can backproject the frequencies $\mathbf{u}_i$ to the surface texture. 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.

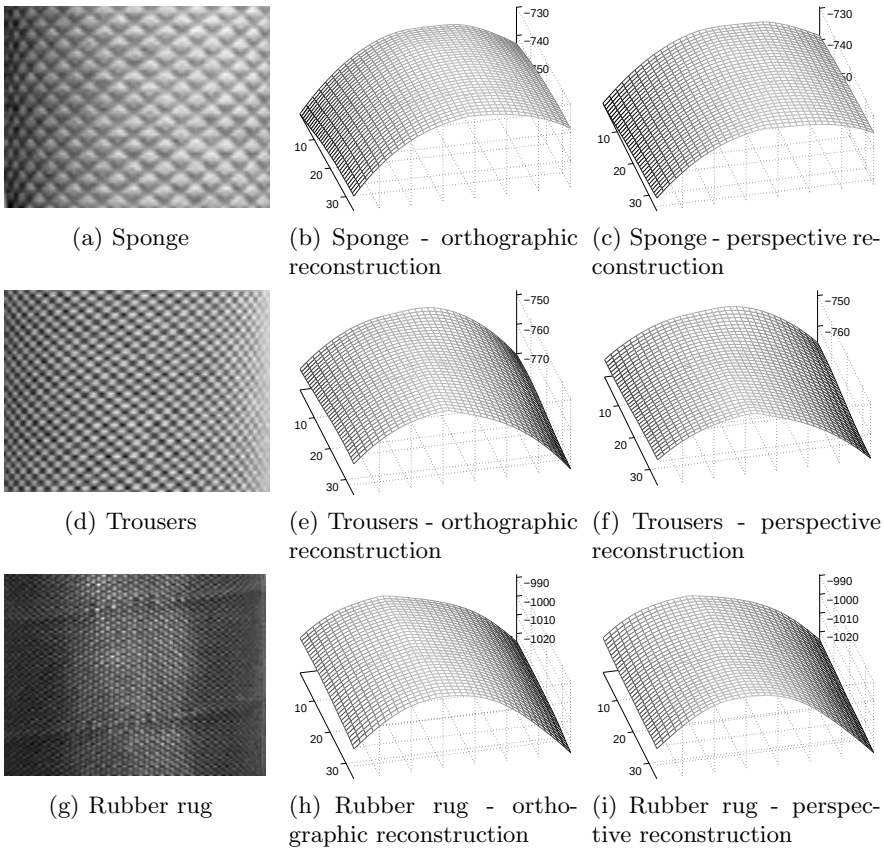
Using two LSFs does not reduce the applicability of our method, because most real textures have at least two LSFs. On the other hand, the over-specification of the systems of equations and the mathematical structure imposed by the SVD provide robustness to noise, as discussed in section 5. Finally, the algorithm lends itself to parallel implementation, because filters and LSFs can be processed independently and implemented separately.

5 Results

We demonstrate our method on two sets of real images. The first set (fig. 4(a),(d),(g)) is taken from [12], with which we compare our results. Figures 4(b),(e),(h) show the orthographic reconstructions, while figures 4(c),(f),(i) show the perspective reconstructions. Table 5 shows the average error for the orientation parameters σ and τ ($|\epsilon_\sigma|$ and $|\epsilon_\tau|$ respectively), alongside the corresponding error from [12]. Compared with the most recent literature (e.g. [10]), our results are accurate, and the pictures in figure 4 show robustness against changes in illumination, self shadowing (e.g. 4(a)) and imperfections (e.g. 4(g)). The slightly

Table 1. Estimation errors for σ's and τ's for orthographic and perspective cases (degrees), compared with results from [12]

	Average error - Orthographic				**Average error - Perspective**			
	New Method		**Method in [12]**		**New Method**		**Method in [12]**	
Image	$\lvert\epsilon_\sigma\rvert$	$\lvert\epsilon_\tau\rvert$	$\lvert\epsilon_\sigma\rvert$	$\lvert\epsilon_\tau\rvert$	$\lvert\epsilon_\sigma\rvert$	$\lvert\epsilon_\tau\rvert$	$\lvert\epsilon_\sigma\rvert$	$\lvert\epsilon_\tau\rvert$
sponge	1.11	4.26	1.55	4.30	1.07	4.48	1.21	4.27
trousers	2.19	5.14	1.48	1.92	2.14	5.50	1.86	1.87
rubber rug	3.28	5.23	4.28	0.44	2.92	5.45	4.39	0.62



(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

higher error in some cases is repaid by increased robustness and by the applicability the new algorithm has to non-developable shapes, as discussed later on in this section.

The second set of images is extracted from a video sequence of a textured cloth laid on a slowly deflating balloon. The video was shot with a Pulnix TM-6EX, at a resolution of 400x300 with 256 levels of gray. Every frame was processed and the 3D shape reconstructed for each of them (top row in fig. 5 shows a sequence of four frames [1,100,200,300], mid and bottom rows show the orthographic and perspective reconstructions respectively). Reconstructions are valid despite image quality being non-optimal. Furthermore the proposed distortion measure and reconstruction algorithm were able to detect small changes in the shape from frame to frame.

Figure 6(a) shows the perspective reconstruction of the first frame of the real video sequence in figure 5, achieved using the algorithm from [12]. It is clear

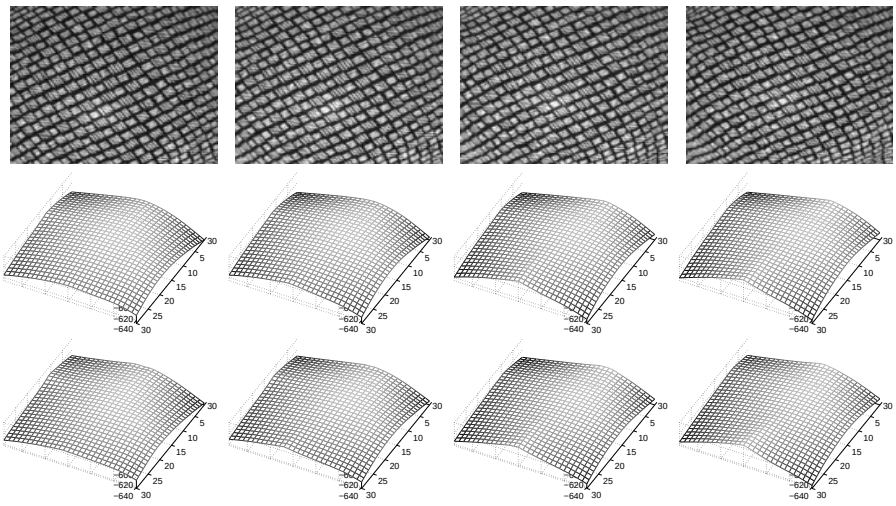
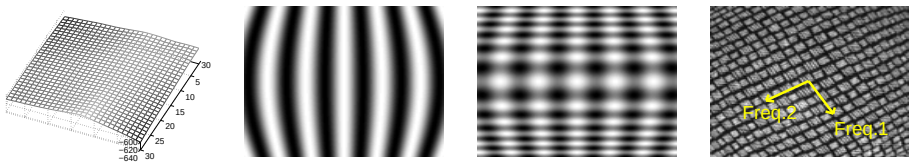


Fig. 5. Frames extracted from the real video sequence of a textured cloth laid on a deflating balloon. Top row: frames 1,100,200,300; Mid row: respective orthographic reconstructions; Bottom row: corresponding perspective reconstructions

from the reconstruction that at some pixels the orientation parameters were misestimated. In fact the algorithm from [12] estimates two sets of orientation parameters separately with the two LSFs considered for each pixel, and then chooses the one set which minimizes the errors for both the equations for the two LSFs. However it does not always chooses the better estimation. In fact when a tilt direction is orthogonal to the spectral direction of a LSF, then the foreshortening effect is null and the shape can only be recovered by considering the distortion due to perspective convergence to vanishing points (e.g. compare figure 6(b), LSF with spectral direction orthogonal to the tilt direction of the cosine shape underneath, and figure 6(c), a second LSF is superimposed with a spectral direction parallel to the tilt direction, which makes it effectively possible to recognize the shape of the cosine). In addition we note that laying a textured cloth on a sphere-like shape, i.e. non-developable, includes some natural stretching, meaning that the homogeneity assumption is only approximately true. This implies noise in the distortion estimation, which adds to the noise always present in real pictures. Figure 6(d) shows the spectral directions of the two main LSFs for the considered frame superimposed on the image. These noise sources imply that, for example, some pixels having a tilt along the spectral direction of frequency 2 are estimated using frequency 1, and are therefore misestimated.

On the other hand, the algorithm proposed here considers the two LSFs together. If their spectral directions do not coincide, our method can deal with all tilt directions. Moreover, thanks to an effective redundancy and to the use of eigenvalues and eigenvectors, the algorithm proposed here is definitely more ro-



(a) (b) (c) (d)

Fig. 6. (a) perspective reconstruction of frame 1 from the video sequence in fig.5, achieved using the algorithm of [12]; (b) cosine textured with a LSF orthogonal to the tilt direction; (c) cosine textured with a LSF orthogonal to the tilt direction (same as (b)) and a second LSF directed parallel to the tilt direction; (d) frame 1 from the video sequence in fig.5 with the spectral directions of its two main LSFs superimposed in the center

bust: it can cope with more noise and it can reconstruct general shapes, including non-developable shapes.

When used to process the whole real video sequence, the algorithm of [12] failed on some frames. The main reason for this is that it estimates each pixel of the image from a previously estimated close pixel. In this way a single error is propagated and can compromise the whole reconstruction. On the other hand our new algorithm estimates every pixel from the frontal point, therefore single errors are isolated and naturally ignored in the final integration of σ's and τ's.

As stated in section 1, the homogeneity assumption requires some sort of periodicity/stationarity: the algorithm works well in cases where the periodicity of the texture is more than one order of magnitude bigger than the main frequency associated with the shape. As in [12], the well known tilt ambiguity is solved assuming convexity. However using a technique such as an EM algorithm, will make our algorithm applicable to more complex shapes.

Finding a robust solution to shape from texture will open the way to possible applications such as the rendering and the retexturing of clothing. The sensitivity of our algorithm to small changes, as in the video sequence result, also makes it possible to use laid-on textures as a non-invasive way of studying shape changes of objects.

6 Conclusions

A method to recover the 3D shape of surfaces from the spectral variation of visual textures has been presented. In particular, the distortion measure has been characterized by local spatial frequencies and estimated with the Fourier transform of the image and with a minimal set of Gabor filters. Then the 3D coordinates of the shape have been reconstructed with a known method which uses singular value decomposition to process the frequencies under orthographic projection. Such a method has here been extended to the general perspective cases and used to reconstruct the 3D shape of real pictures and video sequences.

The results are accurate and comparable with the most recent literature. Robustness against shading, variations in illuminations, and occlusions have been proven. Furthermore the proposed algorithm can effectively use redundancy and is applicable to general non-developable shapes. Finally its sensitivity to small shape changes has been illustrated, and its use to non-invasively study textured-object modifications has been proposed for further study.

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. Kanatani, K., Chou, T.C.: Shape from texture: general principle. Artificial Intelligence **38** (1989) 1–48
3. Gårding, J.: Shape from texture for smooth curved surfaces in perspective projection. JMIV **2** (1992) 327–350
4. Super, B.J., Bovik, A.C.: Shape from texture using local spectral moments. IEEE Trans. Patt. Anal. Mach. Intell. **17** (1995) 333–343
5. Malik, J., Rosenholtz, R.: Computing local surface orientation and shape from texture for curved surfaces. IJCV **23** (1997) 149–168
6. Dugué, A.G., Elghadi, M.: Shape from texture by local frequencies estimation. In: SCIA, Kangerlussuaq, Greenland (1999) 533–544
7. Ribeiro, E., Hancock, E.R.: Estimating the perspective pose of texture planes using spectral analysis on the unit sphere. Pattern Recognition **35** (2002) 2141–2163
8. Clerc, M., Mallat, S.: The texture gradient equation for recovering shape from texture. IEEE Trans. Patt. Anal. Mach. Intell. **24** (2002) 536–549
9. Loh, A.M., Hartley, R.: Shape from non-homogeneous, non-stationary, anisotropic, perspective texture. In: BMVC. (2005) 69–78
10. Massot, C., Hérault, J.: Model of frequency analysis in the visual cortex and the shape from texture problem. IJCV **76** (2008) 165–182
11. Witkin, A.P.: Recovering surface shape and orientation from texture. Artificial Intelligence **17** (1981) 17–45
12. Galasso, F., Lasenby, J.: Shape from texture of developable surfaces via fourier analysis. In: ISVC, Lake Tahoe, CA/NV (2007) 702–713
13. Loh, A.M., Kovesi, P.: Shape from texture without estimating transformations. Technical report, UWA-CSSE-05-001 (2005)
14. Super, B.J., Bovik, A.C.: Planar surface orientation from texture spatial frequencies. Pattern Recognition **28** (1995) 729–743
15. Gabor, D.: Theory of communications. J. Inst. Elec. Engrs **93** (1946) 429–457
16. Frankot, R.T., Chellappa, R.: A method for enforcing integrability in shape from shading algorithms. IEEE Trans. Patt. Anal. Mach. Intell. **10** (1988) 439–451