

Estimation of Surface Normal of a Curved Surface Using Texture

Angeline M. Loh and Peter Kovesi

School of Computer Science & Software Engineering,
The University of Western Australia,
35 Stirling Highway, Crawley, W.A. 6009,
angie,pk@csse.uwa.edu.au

Abstract. Shape-from-Texture is the problem of estimating an object's shape by examining the apparent distortions in the surface texture of the object due to image capture. A new method is described for estimating the tilt angle of local surface patches, based on an analysis of the spectral inertia. The effectiveness of the method is demonstrated on synthetic images generated using the Brodatz textures, as well as a real image.

1 Introduction

Texture is an important visual cue to the perception of object shape. Three dimensional shape estimation is made possible if we have some a priori knowledge about the true surface texture. For example, Figure 1 shows a planewave covered in a texture. If it is known that the texture is homogeneous, a human can easily perceive the object's shape. Attempts to automate this process have been an active area of research in recent years. This problem is known as "Shape-from-Texture" and is currently unsolved.

The next section of this paper will describe the relevant work that has taken place in the area of Shape-from-Texture. A new method for estimating tilt of surface normals is then described in detail. Experimental results from tests with synthetic images and a real image are then presented.

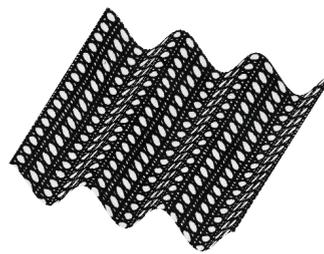


Fig. 1. A texture-covered planewave. Discrepancies between the known true texture and the texture captured in the image can lead to shape estimation.

2 Previous Work

Gibson first suggested in 1950 that the human perception of 3-dimensionality is influenced by texture gradients [8]. Since then, a plethora of computational methods have emerged which estimate shape using texture cues. Many of the earlier methods are *feature-based*, which involve detecting features such as texture elements (texels), edges, or line segments. Shape is then estimated by analysing properties of those features, such as their density gradient, area, size and departure from isotropy. Examples of this kind include the work of Stevens [19], Kender [12], Witkin [22], Ikeuchi [9], Aloimonos and Swain [1], Blostein and Ahuja [3] and Kanatani and Chou [11]. A number of disadvantages arise in the feature-based approach: decision making is often required in the feature-detection stage, where often some threshold is used. This can lead to inaccuracies which affect shape estimation. Also, these methods do not utilise the full information available in the image; they only use information contained in the detected features.

In more recent years, there has been a shift toward Shape-from-Texture methods which utilise spectral information. These methods compare the spectral representation of windowed image patches to recover orientation. Commonly used spectral representations are the Fourier transform, wavelet decomposition and Gabor transform. An advantage of the spectral approach is that it avoids the feature detection step. Also, windowing issues are made simple; a slight shift in the windowing function affects the image domain, but only affects the phase part of the spectral representation. Often, the phase information is discarded and only the amplitude information used.

The first work using a spectral approach was by Bajcsy and Lieberman [2], who used the Fourier power spectrum. They estimated the transformation between a pair of spectra using the location of peaks in each representation. Gårding [7] and Super and Bovik [21] did this estimation using moments. Brown and Shvaytser [5] assumed an isotropic texture and used the autocorrelation function. Krumm and Shafer [13] modeled the transformation between a pair of spectra as an affine transformation, and searched every combination of slant and tilt in a discrete set to estimate the orientation. Malik and Rosenholtz [16] proposed a method to solve for the affine transformation directly. Clerc and Mallat [6] used wavelets rather than the Fourier transform. Ribeiro et al. [18] report a method to recover the slant and tilt using the eigenvectors of the affine distortion matrices. Other spectral-based methods include that of Jau and Chin [10], who used the Wigner distribution, and Plantier et al. [17] who used a wavelet decomposition of the image. Methods which use an adaptive scale have been developed by Stone and Isard [20] and Lee and Kuo [14].

3 Surface Normal Calculation Using Spectral Inertia

A new method has been developed to calculate the tilt τ of the surface normal for local surface patches. This method works with homogeneous textures rather

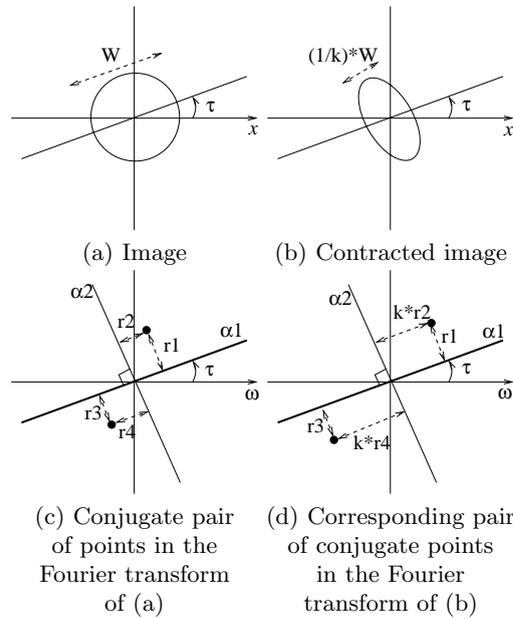


Fig. 2. Principle for using spectral inertia. The image is contracted in the direction τ by a factor of k . In the Fourier image, each point increases its distance from the $\alpha 2$ axis by a factor of k . Since $r 1$ and $r 3$ are unaffected, this results in no change in the inertia about the $\alpha 1$ axis.

than only isotropic textures, allowing its application to many real world images. The method is based on the fact that when an image is *contracted* in a direction τ , the resulting change in the amplitude spectrum of the Fourier transform is a *stretch* in that same direction.

Figure 2 shows the basic principle that when an image is contracted in some direction τ , each point in the Fourier image¹ moves in the direction τ , so that in theory there is no change in the inertia about the axis in that direction. In an ideal situation, the tilt angle could be easily estimated as such: obtain the Fourier images of the original and contracted images, and calculate the inertia, I , about every axis. This is done via the formula

$$I(\theta) = \frac{1}{2}(c + a) - \frac{1}{2}(a - c) \cos(2\theta) - \frac{1}{2}b \sin(2\theta) \quad (1)$$

where θ is the angle of the axis and a , b and c are the second moments, given by

$$\begin{aligned} a &= \sum_u \sum_v u^2 F(u, v)^2, \\ b &= 2 \sum_u \sum_v uv F(u, v)^2, \\ c &= \sum_u \sum_v v^2 F(u, v)^2, \end{aligned} \quad (2)$$

¹ From here onward, the term ‘Fourier image’ will actually refer to the *amplitude spectrum* of the Fourier image.

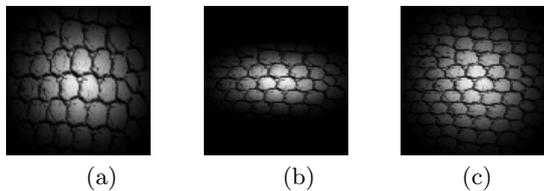


Fig. 3. In (a) we see a texture windowed with a Gaussian. The image in (a) was contracted by a factor of 2 in the direction 90° to produce the image shown in (b). The scaling property applies to the (a)-(b) pair. In (c) the *texture* has been contracted, but the *window* used is the *same Gaussian* as was used in (a). The scaling property is countered by using the same Gaussian in the (a)-(c) pair.

where u and v are the frequencies in the x and y directions respectively and F is the amplitude of points in the Fourier image. The tilt angle is simply the angle for which the inertia of the two Fourier images is equal.

In practice, there are factors which affect the behaviour described above. These include windowing issues and scaling of the Fourier transform when it is stretched; if $\mathcal{F}(f(x)) = F(\omega)$ is the Fourier transform of function $f(x)$, then $\mathcal{F}(f(kx)) = \frac{1}{k}F(\frac{\omega}{k})$. In plain English, if the Fourier transform is stretched out by a factor k , the magnitudes decrease by k .

This scaling action can be cancelled out by the windowing process chosen; consider the image of a texture windowed with a Gaussian shown in Figure 3 (a). This windowed image was contracted by a factor of 2 in the direction 90° to give the image shown in (b). When comparing the Fourier transform of (b) to that of (a), the corresponding peaks increase their distance from the 0° axis by a factor of 2. The scaling property tells us that these peaks also decrease in magnitude by a factor of 2. In figure 3 (c), we see the case where the *texture* has been contracted by a factor of 2 in the direction 90° , but the *window* used is the *same Gaussian* as was used in (a). The Fourier transforms of (b) and (c) will be similar, except the magnitudes for (b) will be 2 times larger than for (c) because twice as much of the contracted texture is visible in the windowed region. Thus, if we compare the Fourier transforms of (a) and (c), a shift in peaks will still occur, but the scaling action will not.

Figure 4 demonstrates the method proposed so far. A planewave texture windowed with a Gaussian is shown in (a). In Figure (b) we see the texture when it has been contracted by a factor of 1.6 in the direction of 50° . The corresponding Fourier transforms are shown in (c) and (d) respectively.

The inertia of the two Fourier transforms as a function of angle of axis are shown in Figure 5, denoted I_1 and I_2 . The dashed line depicts the true tilt angle of 50° . The intersection of the two inertia lies on the required 50° line².

² Note that intersections can also occur at other angles. A detailed explanation of these ‘false’ intersections is not given due to lack of space. Suffice to say, however

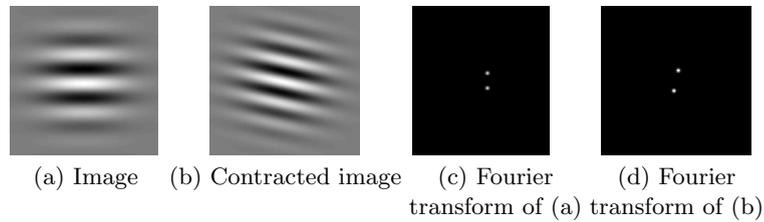


Fig. 4. Planewaves and their Fourier images

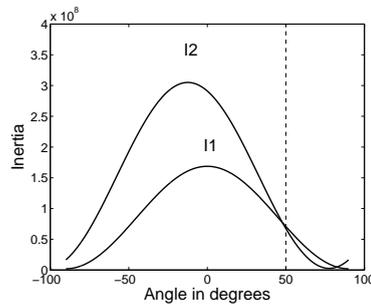


Fig. 5. Spectral inertia of the two planewaves. I_1 corresponds to the original planewave and I_2 corresponds to the contracted planewave. The two curves intersect at the tilt angle of 50° , as required.

The texture shown in Figure 4 responds well to the method because it behaves in a manner which is close to ideal. When using real images, however, the method outlined so far does not perform well. Unwanted effects are added to the Fourier transform via blurring and contrast differences. Blurring tends to draw the power in the Fourier image closer to the origin. Contrast differences scale the inertia.

In light of the above, the method outlined so far for estimating tilt angle requires modification when real images are input. What we seek to establish is the *axis about which the points in the Fourier transform tend to ‘congregate’ when the texture is contracted*. This axis gives the tilt angle. Figure 6 depicts the ‘congregating’ behaviour. An image of a texture taken from the Brodatz database is shown in Figure 6 (a). Its Fourier image is shown in (b). The texture was contracted by a factor of 2.5 in the direction of 40° to give the Fourier transform shown in (c). In comparing (b) with (c), the peaks appear to gather about the axis in the direction of 40° , denoted by the black line.

The ‘congregation’ of points near the tilt axis tends to decrease the inertia about that axis. In fact, when comparing the Fourier image of the original texture with that of the compressed texture, the inertia will decrease by a *maximal*

that ‘false’ solutions do not occur if there are enough peaks in the Fourier image, as is the case with almost all real textures.

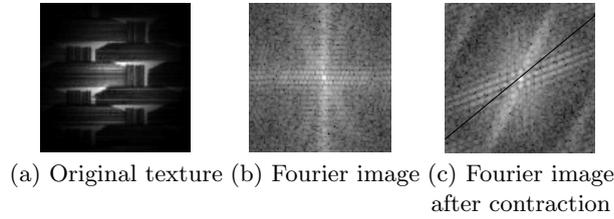


Fig. 6. ‘Congregation’ of Fourier peaks. A Brodatz texture is shown in (a). The corresponding log-Fourier image is shown in (b). When the texture is contracted by a factor of 2.5 in the direction of 40° , its log-Fourier transform appears as shown in (c). The black line denotes the axis in the direction of 40° .

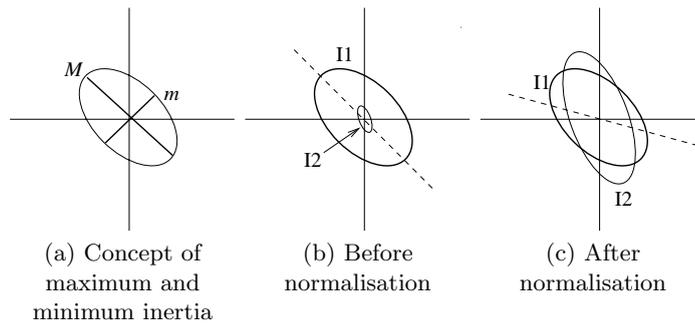


Fig. 7. Normalisation of inertia. Figure (a) gives a representation of spectral inertia in the form of an ellipse. In (b), I_1 is the spectral inertia of some frontal patch. I_2 is the spectral inertia of a sample patch. The dotted line is the angle where I_2 is maximally less than I_1 . The case where I_2 has been normalised to I_1 is shown in (c). The dotted line now shows the correct tilt angle.

amount about the tilt axis. It is this tilt axis that we wish to identify, as it leads us to a direct estimation of the tilt angle.

Suppose we are estimating the tilt angle in a sample patch of texture, and do this by comparing it to a patch containing the frontal (true) texture. We wish to find the axis about which the spectral inertia of the sample patch is maximally less than the spectral inertia of the frontal patch. In order to be compared, the two inertia must firstly be normalised to avoid the effects of blur and contrast variations.

The spectral inertia may be represented by an ellipse with major diameter M and minor diameter m , where M and m are the maximum and minimum inertia respectively. This is depicted in Figure 7 (a).

In Figure 7 (b) we see two ellipses: I_1 represents the spectral inertia of some frontal patch and I_2 represents the spectral inertia of a sample patch. The dotted line shows the angle at which I_2 is maximally less than I_1 . This dotted line

inaccurately estimates the tilt axis, since normalisation has not been performed. Normalisation, which is the process of matching the areas of the two ellipses as described later, results in an accurate estimate of the tilt angle.

We are only interested in the general change in *shape* in order to give the best guess of tilt angle. The *size* of the ellipses is unimportant and easily affected by blur, lighting etc. I_1 can be thought of as some unit inertia shape, multiplied by some factor c_1 , where c_1 is unpredictably affected by blur and contrast variations in the image. Similarly, I_2 can be thought of as another unit inertia shape multiplied by a different factor, c_2 . I_2 can be normalised to I_1 if it is scaled by a factor $\frac{c_1}{c_2}$. This is done by multiplying I_2 by a factor $\sqrt{\frac{M_1 m_1}{M_2 m_2}}$ where M and m are the maximum and minimum inertia respectively, and the subscripts denote the inertia to which we refer. This process of scaling I_2 equalises the area of the two ellipses. Figure 7 (c) shows I_2 after it has been normalised to I_1 . The tilt angle is then correctly chosen as the angle where I_2 is maximally less than I_1 .

In summary, the initial approach of estimating tilt angle using the *intersection* of the two inertia is only accurate if an ideal texture is used. The addition of blur and contrast variations require that we analyse instead the *change in the shape* of the inertia, and this can only be done in a normalised framework.

Currently, the algorithm works on images of surfaces displaying a frontal point i.e. a point where the tangent surface is parallel to the image plane, and therefore the true texture is displayed. An orthogonal camera model is employed, which means that the algorithm is only valid for images captured in a small visual angle. The algorithm for estimating the tilt of a curved surface, can be summarised as follows:

1. Begin with an image of a uniformly-textured surface. Window the image into local patches using a Gaussian window.
2. Identify a window displaying a frontal point on the surface. In this current work, frontal points are found manually however we are developing a method to automatically do this.
3. For each windowed patch, take the Fourier transform.
4. Calculate the spectral inertia as a function of angle for the local patch and the frontal patch.
5. Multiply the inertia in the local patch by $\sqrt{\frac{M_1 m_1}{M_2 m_2}}$, where M_1 and m_1 are the maximum and minimum inertia in the frontal patch, and M_2 and m_2 are similar but refer to the local patch.
6. Identify the axis for which the inertia of the local patch is maximally less than the inertia of the frontal patch. The angle of this axis is the estimated tilt angle.

Much of the recent work in Shape-from-Texture focuses on solving the affine transformation between pairs of spectra. Surface normals are then interpreted from the transformation. In contrast, the method proposed in this work does not solve a transformation, and instead estimates the tilt angle using a property of that angle: under normalised conditions, the inertia about the tilt axis decreases

by a maximal amount as the texture is slanted. There is no need to search over any large search space. Since we are using the Fourier transform, no choice of filter scale, spacing and number is required, however an appropriate window size must be chosen.

A common difficulty in Shape-from-Texture is instability of the tilt angle near the frontal position. At the frontal point itself tilt is undefined, and in nearby areas tilt is difficult to calculate accurately. Under these conditions some algorithms can yield complex values for tilt [15]. The method outlined in this paper is guaranteed to yield a real (and hence usable) value for tilt, by choosing a best guess from the range of real angles. Apart from the standard π shift ambiguity³, some methods introduce additional ambiguities through the structure of their mathematical formulation [21]. This is not the case here. Complex windowing issues in the spatial domain are made simple in the spectral domain, where using the same Gaussian window over the entire image only results in a scaling of spectral information. Also, no preprocessing step is required to remove the lighting variations on the textured surface, as this is compensated for, along with some amount of blurring, in the inertia normalisation step.

4 Results

The algorithm was tested on the images in the Brodatz texture collection [4]. The initial testing procedure involved comparing two images of the same texture: the first image displaying the frontal texture, and the second image displaying the contracted version.

The two images were obtained using the following steps: window the centre portion of the texture image, using a Gaussian window. This gives the first image. Contract the original texture image by a factor k in direction τ , where k and τ are some initial test values. Window the center portion of this contracted image with the same Gaussian to obtain the second image.

When the two images were input into the algorithm, the aim was to estimate the angle τ . Then k and τ were varied to test the algorithm under different multiplying factors and tilt angles.

Figure 8 (a) displays information relating to the example in Figure 6 (a homogeneous but not isotropic texture), when the texture was contracted by a factor of 1.6 in the direction of 40° . The spectral inertia of the original image is labeled I_1 , and the spectral inertia of the contracted image is labeled I_2 .

The normalised inertia is labeled I_3 . The difference between I_3 and I_1 is also shown. In this case, I_3 is maximally less than I_1 at the angle 38° , which closely matches the tilt angle of 40° , shown by the dashed line.

Figure 8 (b) displays the outcome of experiments with the same texture when the angle of tilt, τ was varied. On average, the estimated tilt angle was 5.5° in

³ This ambiguity is demonstrated by the fact that under orthographic projection, it is impossible to distinguish between a concave textured sphere and a convex one, based on texture alone.

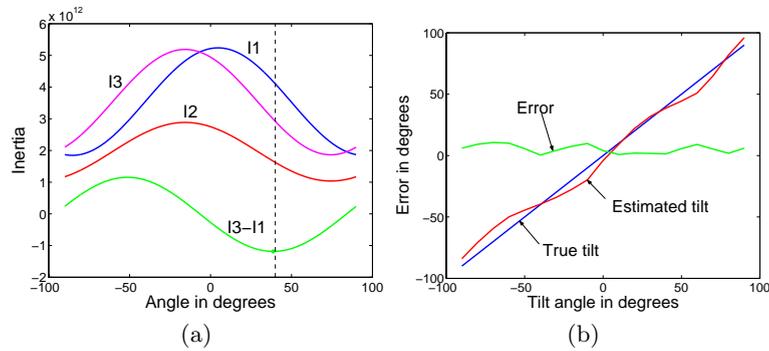


Fig. 8. Result of tilt angle estimation. In (a), the angle of 40° is accurately estimated. Tests over various angles are shown in (b).

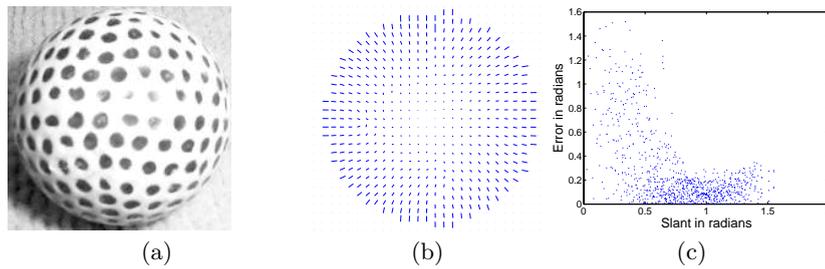


Fig. 9. An image of a golf ball is shown in (a). The corresponding needle diagram is shown in (b). A plot of error versus slant for every image point is displayed in (c).

error, with the worst case not more than 11° in error. The algorithm was then applied to the entire Brodatz database, giving similar results.

The algorithm was then tested on a real image. Figure 9 (a) shows the image of a golf ball. The image was windowed into local patches, and the tilt angle was estimated for each point on the surface. The slant angle was manually input (after the tilt was calculated) in order to produce the full needle diagram as shown in Figure 9 (b). The experiment yielded an average error in tilt angle of 12.5° . It was found, as expected, that the highest errors came from areas near the frontal point, as shown in Figure 9 (c).

5 Conclusion

In Shape-from-Texture the goal is often to estimate the slant and tilt of surface normals of a textured surface. In this paper, an algorithm for estimating the tilt angle has been described. It is based on an analysis of the change in shape of the inertia curve, with respect to angle, when a texture is contracted. The algorithm has been shown to work well on a real texture in the presence of blur and lighting variations.

Acknowledgments

This work was funded by an Australian Postgraduate Award (APA) and a Western Australian IVEC Doctoral Scholarships (WAIDS) top-up.

References

1. Y. Aloimonos and M.J. Swain. Shape from texture. *Biological Cybernetics*, 58(5):345–360, 1988.
2. R.K. Bajcsy and L.I. Lieberman. Texture gradient as a depth cue. *CGIP*, 5(1):52–67, 1976.
3. D. Blostein and N. Ahuja. Shape from texture: Integrating texture-element extraction and surface estimation. *PAMI*, 11(12):1233–1251, December 1989.
4. P. Brodatz. *Textures: A photographic album for artists and designers*. Dover Publications, New York, 1966.
5. L.G. Brown and H. Shvaytser. Surface orientation from projective foreshortening of isotropic texture autocorrelation. *PAMI*, 12(6):584–588, June 1990.
6. M. Clerc and S. Mallat. The texture gradient equation for recovering shape from texture. *PAMI*, 24(4):536–549, April 2002.
7. J. Garding. Shape from texture for smooth curved surfaces. In *ECCV92*, pages 630–638, 1992.
8. J.J. Gibson. *The Perception of the Visual World*. Houghton Mifflin, 1950.
9. K. Ikeuchi. Shape from regular patterns. *Artificial Intelligence*, 22(1):49–75, 1984.
10. J.Y. Jau and R.T. Chin. Shape from texture using the wigner distribution. *CVGIP*, 52(2):248–263, November 1990.
11. K.I. Kanatani and T.C. Chou. Shape from texture: General principle. *AI*, 38(1):1–48, February 1989.
12. J.R. Kender. Shape from texture: An aggregation transform that maps a class of textures into surface orientation. In *IJCAI79*, pages 475–480, 1979.
13. J. Krumm and S.A. Shafer. A characterizable shape-from-texture algorithm using the spectrogram. In *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pages 322–325, 1994.
14. K.M. Lee and C.C.J. Kuo. Direct shape from texture using a parametric surface model and an adaptive filtering technique. In *CVPR98*, pages 402–407, 1998.
15. A.M. Loh and P. Kovesi. Shape from texture using local spectral inertia. In *Proceedings of the Twelfth University of Western Australia School of Computer Science and Software Engineering Research Conference*, pages 13–19, 2003.
16. J. Malik and R. Rosenholtz. A differential method for computing local shape-from-texture for planar and curved surfaces. In *CVPR93*, pages 267–273, 1993.
17. J. Plantier, S. Lelandais, and L. Boutté. Orientation computation of an inclined textured plane: Study of results for regular macro textures. In *ICIP02*, pages III: 853–856, 2002.
18. E. Ribeiro, P.L. Worthington, and E.R. Hancock. An eigendecomposition method for shape-from-texture. In *ICPR00*, pages Vol I: 758–761, 2000.
19. K.A. Stevens. The information content of texture gradients. *BioCyber*, 42:95–105, 1981.
20. J.V. Stone and S.D. Isard. Adaptive scale filtering: A general-method for obtaining shape from texture. *PAMI*, 17(7):713–718, July 1995.
21. B.J. Super and A.C. Bovik. Shape from texture using local spectral moments. *PAMI*, 17(4):333–343, April 1995.
22. A.P. Witkin. Recovering surface shape and orientation from texture. *AI*, 17(1-3):17–45, August 1981.