

# The Adaptive Bisector Method: Separating Slant and Tilt in Estimating Shape from Texture\*

James V. Stone

Department of Computer Science,  
University of Wales, Aberystwyth.

JANET: zra@uk.ac.aber

## Abstract

Existing techniques for obtaining shape from texture estimate tilt and slant via a single computational mechanism. These techniques do not take advantage of the fact that tilt is relatively easy to estimate, and slant can be estimated more easily once tilt is known. This paper introduces the *adaptive bisector method*, which allows tilt and slant to be calculated separately, and by different computational mechanisms. The method makes minimal assumptions regarding the isotropy of surface textures.

Evidence from psychophysical studies suggests that human observers are able to provide accurate estimates of tilt[8], but are poor at estimating slant[4]. Whilst no psychophysical claims are made regarding the means by which tilt and slant are computed in this paper, it is claimed that a method which depends upon two different mechanisms provides a plausible functional model for how human observers obtain shape from texture.

Results for real and synthetic perspective images of textured planar surfaces are presented.

## 1 Introduction

Conventional methods for obtaining shape from texture[10, 1, 5, 2] use a single computational mechanism to estimate surface orientation. Such methods ignore constraints implicit in the underlying nature of the problem, namely, that tilt (the direction of surface slope) is relatively easy to compute on its own, and that slant (the amount of surface slope) can be computed more easily once the tilt is known.

The adaptive bisector method (ABM) is unique amongst shape from texture methods in addressing the problems of estimating slant and tilt separately. Evidence from psychophysical studies suggests that human observers are able to provide accurate estimates of tilt[8], but are poor at estimating slant[4]. Reasons for this dichotomy in the perception of slant and tilt are not hard to find. Whereas tilt can be computed without reference to the 3D structure of a surface, slant cannot. Once the tilt is known then slant becomes easier to estimate, as evidenced by the existence of psychophysical models [6, 11] which estimate slant only, and require that the tilt is known before slant can be

---

\*This work was undertaken as part of a D.Phil. thesis in the Dept. of Experimental Psychology at the University of Sussex.

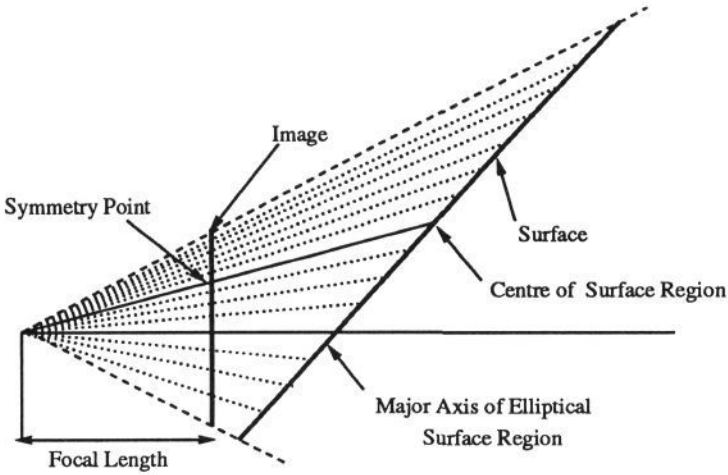


Figure 1: Back-projection of centre of elliptical surface region to the symmetry point in the image.

estimated. Computational results presented here support the implication from psychophysical studies that, whilst it is relatively easy to compute accurate estimates of tilt, estimating slant is both more expensive and less accurate.

As discussed in [7] tilt can be obtained either by finding the image direction in which texture density changes most rapidly, or via the image orientation in which texture density changes least. In practice finding tilt via either of these methods is problematic[10], and neither has been shown to work. Accordingly, we describe a new method for estimating tilt below. The point is that the slant of the surface need not be known in order to estimate tilt. Once the tilt has been established, slant can be estimated by making use of the rate of change of image texture density in the direction of tilt. Thus, the process of estimating surface orientation can be decomposed into two separate processes.

## 2 The Adaptive Bisector Method

### 2.1 Preliminary Definitions

The local orientation of a surface is specified by two parameters. Given ‘world’ coordinates  $(X, Y, Z)$  these parameters can be either the partial derivatives  $P = -\partial Z/\partial X$ ,  $Q = -\partial Z/\partial Y$ , or the more intuitive descriptors, *slant*,  $\sigma = \tan^{-1}(P^2 + Q^2)^{1/2}$ , and *tilt*,  $\tau = \tan^{-1}(Q/P)$ . Slant specifies the amount of slope, whereas tilt indicates the direction of slope apparent in an image.

Consider an elliptical region,  $W_0$ , of a planar surface which projects to a circular image,  $w_0$  (see Figures 1 and 2). The major axis of  $W_0$  projects to an image line, the *tilt line*, which passes through the image origin at angle  $\tau$ . The tilt line bisects  $w_0$ , and (by definition) the back-projection of the tilt line bisects the surface region,  $W_0$ .

Of the set of image lines orthogonal to the tilt line exactly one back-projects

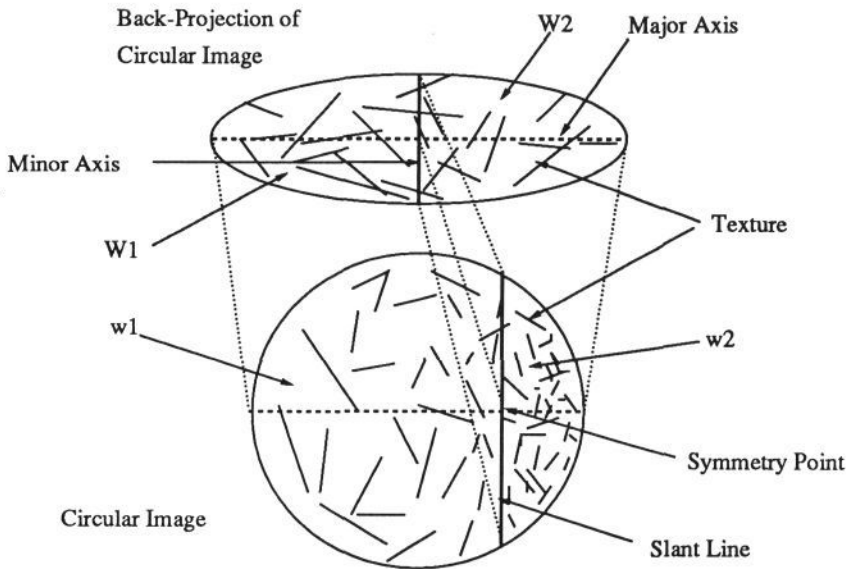


Figure 2: Back-projection of slant line from circular image to elliptical surface region.

to coincide with the minor axis of the surface ellipse. We define this image line as the *slant line*. Like the tilt line, the back-projection of the slant line bisects the surface region,  $W_0$ . The intersection of these two lines is the *symmetry point*,  $p_C = (x_C, y_C)$ . The symmetry point is the projection of the centre of the surface ellipse. Note that the image point  $p_C$  and the surface orientation are co-determined. Thus, if the location of  $p_C$  is known then it is possible to recover the surface orientation.

The remainder of this section describes how to estimate surface orientation by making use of the definitions given above. A summary of the method is as follows. First, the tilt line is estimated from the direction of the first moment of surface texture around the image origin. This provides an estimate of the tilt of the textured surface. Next, an estimate of the slant line is obtained by finding that image line which is orthogonal to the tilt line and which bisects the total amount of surface texture projecting into the image. The distance of the symmetry point,  $p_C$ , from the image origin can then be used to recover surface slant.

The back-projection of the slant line onto the surface divides the surface region,  $W_0$ , into two surface regions,  $W_1$  and  $W_2$ , with areas,  $A_1 = A_2$ . These regions correspond to image regions,  $w_1$  and  $w_2$ , respectively (see Figure 2). From the definition of the slant line it follows that:

$$A_1 - A_2 = 0 \quad (1)$$

If the amount of texture per unit surface area is constant then the amount of texture,  $L_1$ , in  $W_1$  is the same as the amount of texture,  $L_2$ , in  $W_2$ , so that:

$$L_1 - L_2 = 0 \quad (2)$$

These equations form the basis of the ABM, but in order for them to be useful we need to re-write equations (1,2) in terms of image area and image line length.

The equation of a plane can be written in terms of the 'world' coordinates  $(X, Y, Z)$  as  $AX + BY + CZ - D = 0$ . For an imaging system with focal length,  $f = 1$ , and with the image plane at  $Z = f$ , this equation can be written in terms of image coordinates  $(x, y) = (x/f, y/f) = (X/Z, Y/Z)$  as  $Z = K/(1 + Px + Qy)$ , where  $P = -\partial Z/\partial X$ ,  $Q = -\partial Z/\partial Y$ , and  $K = D/C$  determines the orthogonal distance from the plane to the origin. The term  $f$  is omitted hereafter so that  $x = x/f, y = y/f$ . By using functions,  $pdf$  and  $lcf$ , that map area and length, respectively, from surface to image we can re-write (1) and (2) in terms of image quantities and the surface orientation parameters,  $(P, Q)$ :

$$\int_{w_1} pdf(x, y, P, Q, K) da - \int_{w_2} pdf(x, y, P, Q, K) da = 0 \quad (3)$$

$$\sum_{l_i \in w_1} \int_{l_i} lcf(x, y, P, Q, \beta) ds - \sum_{l_j \in w_2} \int_{l_j} lcf(x, y, P, Q, \beta, K) ds = 0 \quad (4)$$

Where  $\beta$  is the orientation of an image line,  $l$ , parameterised by arc-length,  $s$ .

The function  $pdf$  is a *point density function* that maps the differential of image area,  $da$ , to a differential of surface area,  $dA$ . The function  $pdf$  is defined in [10] as:

$$dA/da = pdf(P, Q, x, y, K) = K^2(P^2 + Q^2)^{1/2}(1 + Px + Qy)^{-3} \quad (5)$$

The function  $lcf$  is a *line compression function* that maps the differential of image length,  $ds$ , to a differential of surface length,  $dS$ . The function  $lcf$  is defined in [10] as:

$$\begin{aligned} dS/ds = lcf(P, Q, x, y, K) = & (K/V)(Qx \sin(\beta) - \cos(\beta)(1 + Qy))^2 + \\ & (Py \cos(\beta) - \sin(\beta)(1 + Px))^2 + \\ & (P \cos(\beta) + Q \sin(\beta))^2)^{1/2} \end{aligned} \quad (6)$$

Where  $V = (1 + Px + Qy)^2$ .

For the present let us assume that  $Q = 0$  so that the tilt,  $\tau = \tan^{-1}(Q/P) = 0$ , and the slant  $\sigma = \tan^{-1}(P^2 + Q^2)^{1/2} = \tan^{-1}(P)$ . Each point,  $p_m$ , on the tilt line can be associated with a line which is orthogonal to the tilt line. Each of these lines defines two image regions,  $w_i$  and  $w_j$ . If a point,  $p_m$ , on the tilt line is the symmetry point then its position is associated with a value for the slant ( $\sigma = \tan^{-1}(P)$ ). Additionally, if  $p_m$  is the symmetry point then  $w_i$  and  $w_j$  are  $w_1$  and  $w_2$ , and their back-projections onto the surface contain the same amount of surface texture. Thus each point,  $p_m$ , on the tilt line is associated with a particular value of  $P$ , which can be obtained from the inverse of (3). The value of  $P$  can be used to compute (from (4)) an estimate of the difference  $\delta L = L_1 - L_2$ . The value of  $P$  (and the corresponding value of  $p_m$  with its associated pair of regions) that satisfies (4) is the estimate of the surface slant ( $= \tan^{-1}(P)$ ). Thus we are actually seeking a triplet of co-determined entities,  $\sigma, p_m$ , and the image region-pair  $(w_1, w_2)$ . Before we can set about finding the triplet that satisfies (4) we need to address two remaining problems; how to estimate tilt, and how to obtain  $P$  given a value for  $p_m$ .

## 2.2 Estimating Tilt

The back-projection of a circular image,  $w_0$ , defines a conic section (an ellipse) on the surface. The image direction,  $\tau$ , in which the texture density increases most rapidly lies along the tilt line. The value of  $\tau$  can be obtained from the centre of mass,  $(x_A, y_A)$ , where:

$$\begin{aligned} x_A &= \frac{\int_{w_0} x \text{pdf}(x, y, P, Q, K) da}{\int_{w_0} \text{pdf}(x, y, P, Q, K) da} \\ y_A &= \frac{\int_{w_0} y \text{pdf}(x, y, P, Q, K) da}{\int_{w_0} \text{pdf}(x, y, P, Q, K) da} \end{aligned} \quad (7)$$

The point  $(x_A, y_A)$  is the (normalised) first moment of the surface area around the image origin. For an elliptical surface region,  $W_0$ , it can be shown [10] that the direction of tilt,  $\tau = \tan^{-1}(y_A/x_A)$ . If we assume a constant amount of texture (line length) per unit surface area then, for image line segments,  $l_i : i = 1..N$ , centred at  $p_i = (x_i, y_i)$ , the point  $(x_A, y_A)$  is approximated by:

$$\begin{aligned} x_L &= \frac{\sum_{l_i \in w_0} x_i \int_{l_i} \text{lc}f(x, y, P, Q, \beta, K) ds}{\sum_{l_i \in w_0} \int_{l_i} \text{lc}f(x, y, P, Q, \beta, K) ds} \\ y_L &= \frac{\sum_{l_i \in w_0} y_i \int_{l_i} \text{lc}f(x, y, P, Q, \beta, K) ds}{\sum_{l_i \in w_0} \int_{l_i} \text{lc}f(x, y, P, Q, \beta, K) ds} \end{aligned} \quad (8)$$

Therefore  $\tau$  can be estimated as  $\tau \simeq \tan^{-1}(y_L/x_L)$ .

In attempting to estimate tilt we initially set  $(P, Q) = (P_0, Q_0) = (0, 0)$ . For isotropic textures  $\tan^{-1}(y_L/x_L)$  provides an exact estimate of  $\tau$  if  $P_0 = Q_0 = 0$  (or if  $Q_0/P_0 = \tan(\tau)$ ). This is because an error generated by the mapping function  $\text{lc}f$  with respect to an image line at some orientation on one side of the tilt line is guaranteed to be compensated for by an error generated with respect to a corresponding line at a different orientation on the other side of the tilt line.

For anisotropic textures setting  $P_0 = Q_0 = 0$  reduces the accuracy of (8) as a positive function of the degree of anisotropy of the surface texture. This is because the error-correcting behaviour described above for isotropic textures diminishes with the degree of anisotropy of the surface texture. The actual method used to compute  $\tau$  involves iterative improvement such that  $P_i \rightarrow P, Q_i \rightarrow Q$ , and if  $P_0 = P$  and  $Q_0 = Q$  then (8) provides an exact estimate of  $\tau$ .

An additional source of error is caused by the differential image resolution associated with surface points at different distances from the image. Image data derived from surface points which are most distant from the focal point have minimal image resolution. Thus, image data becomes increasingly noisy within a given image as distance to the surface increases. As with errors caused by anisotropy of surface texture, errors caused by differential image resolution tend to be symmetric about the tilt line. This is because two image points which are equidistant from the tilt line (and which are on a line orthogonal to the tilt line) back-project to surface points which are the equidistant from the focal point.

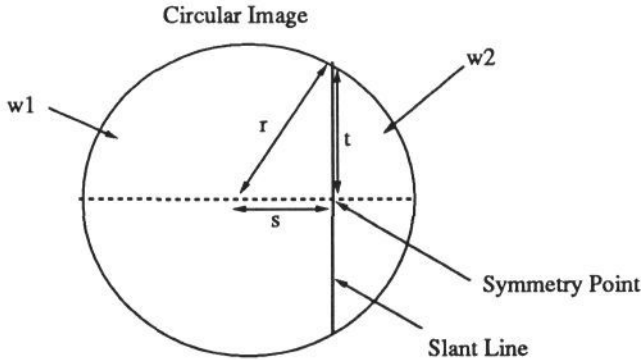


Figure 3: Computing the location of the symmetry point.

From this it follows that, even though *lcf* makes use of incorrect values for  $(P, Q)$ , and uses these to back-project noisy image data, the resultant errors tend to be symmetric about the tilt line. Thus, the estimated orientation,  $\tan^{-1}(y_L/x_L)$ , of the tilt line is more robust with respect to these sources of error than is the length of the vector,  $(x_L, y_L)$ . Accordingly, errors in the estimated position,  $(x_L, y_L)$ , of  $(x_A, y_A)$  are not necessarily consonant with corresponding errors in the estimated angle,  $\tan^{-1}(y_L/x_L)$ , of  $\tau$ .

This method for estimating tilt makes use of moments about the image origin, and, in this respect, is similar to the methods described in [5, 10]. However, whereas the orientation of the vector  $(x_L, y_L)$  is an accurate estimate of the surface tilt, the length of this vector does not generally provide an accurate estimate of the surface slant (see [10] for a more detailed discussion). By using the quantity  $(x_L, y_L)$  to estimate tilt only, we extract information which is most accurately encoded by the measured quantity.

### 2.3 Estimating Slant

Given an estimate for tilt, the problem of estimating slant consists in finding the image line which passes through  $w_0$ , orthogonal to the tilt line, satisfying equation (4). If  $\tau$  is known then the coordinate system can be rotated around the optic ( $Z$ ) axis until, in the rotated coordinate system,  $(P', Q') = (\tan(\sigma), 0)$ . Rotating the coordinate system does not affect the value of  $\sigma$ , and is used only as a means of simplifying the equations that follow. Omitting the term  $Qy$  from *pdf*, we now have:

$$\int_{w_0} pdf(x', P', K) da' = \int_{w_1} pdf(x', P', K) da' + \int_{w_2} pdf(x', P', K) da' \quad (9)$$

Combining this with (3):

$$\int_{w_0} pdf(x', P', K) da' - 2 \int_{w_1} pdf(x', P', K) da' = 0 \quad (10)$$

The choice of  $w_1$  in the second term of (10) is arbitrary, and  $w_2$  would do just as well.

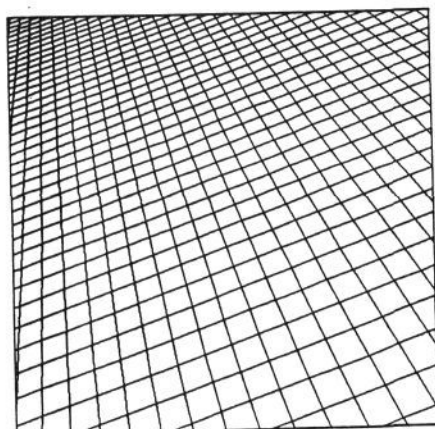


Figure 4: Synthetic image of grid texture. Figure 5: Line texture derived from elliptical distribution.

If  $(P'x') \ll 1$  then<sup>1</sup> the slant is small relative to the focal length, and (after [5]) we can approximate (5) by replacing the function  $pdf$  by the first two terms of its Taylor expansion:

$$\int_{w_0} 1 + 3P'x' da - 2 \int_{w_1} 1 + 3P'x' da' = 0 \quad (11)$$

By evaluating (11) and solving for  $P'$  it can be shown [10] that:

$$P' = \frac{st(r^2 \cos^{-1}(t/r) - 2)}{2(r^3 - t^3)} \quad (12)$$

where  $r$  is the radius of  $w_0$ ,  $s$  is the distance from  $p_m$  to the image origin, and  $t = (r^2 - s^2)^{1/2}$  (see Figure 3). Thus for a putative symmetry point,  $p_m$ , we can compute its distance,  $s$ , from the origin and from (12) obtain the corresponding estimate,  $P'$ , of  $\tan(\sigma)$ . This estimate is the correct value of  $\tan(\sigma)$  if  $p_m$  is the symmetry point. Note that, although this derivation was performed assuming  $Q' = 0$ , (12) can be used to obtain slant in general if  $s$  is known, because (by definition)  $P' = (P^2 + Q^2)^{1/2} = \tan^{-1}(\sigma)$ .

## 2.4 Combining Estimates of Slant and Tilt

We are now in a position to define a procedure for estimating  $(P, Q)$ . We begin by setting the initial estimate of surface orientation  $(P_0, Q_0) = (0, 0)$ , and using the method described earlier to obtain an initial estimate,  $\tau_1 = \tan^{-1}(y_L/x_L)$ , of  $\tau$ . (Whilst estimating  $\tau$  the back-projected length of each image line (evaluated as part of (8)) is recorded, so that it can be used in estimating slant). Setting  $P = P_0, Q = Q_0$  in (4), we then estimate slant,  $\sigma_1$ . This is achieved by searching along the estimated tilt line for a point,  $p_m$ , that

<sup>1</sup>Recall that  $(x, y) = (x/f, y/f)$ .



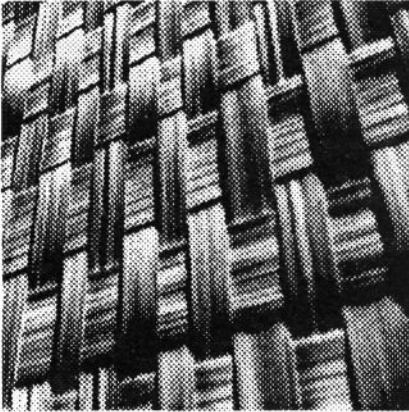


Figure 6: Algorithmically rotated image.



Figure 7: Image of wallpaper.

minimises (4). When this line search is complete the distance,  $s$ , of  $p_m$  from the image origin can be used to obtain, from (12), an estimate,  $\sigma_1 = \tan^{-1}(P')$ , of the surface slant. We then re-express  $(\sigma_1, \tau_1)$  in terms of partial derivatives to obtain  $(P_1, Q_1)$ . This is used to obtain (from (4)) a new estimate,  $\tau_2$ , of  $\tau$ , and then  $\sigma_2$ , as above. This process is repeated until the difference between successive estimates of the surface normal is less than  $0.1^\circ$ . The procedure for estimating slant involves use of the data recorded whilst computing the tilt, and does not therefore utilise a new value for  $P$  (from (12)) for each candidate slant line. In essence, the slant line is found by moving  $p_m$  along the tilt line until the pre-computed estimates of surface line length on either side of the candidate slant line (defined by  $p_m$ ) are the same. Thus, each successive estimate of  $(P, Q)$  requires each image edge to be back-projected once only. Re-computing the lengths of back-projected image lines for each candidate slant line would reduce the number of iterations required, but the computational cost of so doing would be prohibitive. Moreover, results indicate that accuracy is not noticeably altered using the form of relaxation described above.

### 3 RESULTS

Results for the ABM were obtained using synthetic line textures (Figures 4 and 5), an algorithmically rotated grey level image of a photograph (from [3]) of textured surface (Figure 6), and an image of a (real) textured surface. The image area used in each image is defined by a circle with diameter equal to the width of the square images shown in Figures 4-7. The orientation of surfaces in Figures 4-6 is  $(\sigma, \tau) = (40^\circ, -20^\circ)$ . The orientation of the surface shown in Figure 7 was estimated as  $(\sigma, \tau) = (27.23^\circ, 0.050^\circ)$  by manually marking each texel with a dot, and using the method (for dot textures) described in [5]. The focal length of the imaging system is equal to half the image height for all images. Each grey-level image consists of a  $512 \times 512$  array of pixels, with



| Figure | Slant  | Tilt    | Error |
|--------|--------|---------|-------|
| 4      | 41.60° | -20.60° | 1.66° |
| 5      | 41.21° | -18.20° | 1.70° |
| 6      | 35.98° | -32.11° | 8.46° |
| 7      | 32.37° | -1.09°  | 5.41° |

Table 1: Results for Figures 4-7.

256 grey levels. The error value given for each image is the angle between the actual and estimated surface normals.

Results for the anisotropic synthetic line textures shown in Figures 4 and 5 are displayed in Table 1. The line segments of the surface texture shown in Figure 5 are derived from an elliptical distribution with ratio of major:minor axes of 5:2; all line segments have the equal length and are randomly distributed on the surface. The initial and final estimates of tilt for these images did not differ by more than 2°, indicating that even when the slant is not known the tilt can be accurately estimated. The number of ABM iterations was 5 for both synthetic images. Results for the grey level images shown in Figures 6 and 7 (see Table 1) were obtained using the ABM with the adaptive multi-scale filtering (AMSF) technique<sup>2</sup>[9, 10]. As is usual with AMSF a ‘shape from texture’ method (e.g. ABM) is used to provide successive estimates of surface orientation, and each of these estimates is used as part of the next filtering operation. This type of nested iteration has been shown [10] to be effective for several different shape from texture methods(e.g. [5, 1])<sup>3</sup>. After each filtering operation of AMSF the estimate of tilt did not vary by more than 2-3°, although (as is to be expected) the estimate of surface orientation varied across successive filtering operations as the AMSF method converged. The number of AMSF iterations required was 15 for Figure 6 and 10 for Figure 7, and the number of ABM iterations (between AMSF filtering operations) was typically around 5.

## 4 Conclusion

We have demonstrated a method for estimating slant and tilt using separate computational mechanisms. Each of these mechanisms uses quantities ( $(y_L/x_L)$  and the origin-symmetry point distance) which robustly index slant and tilt, respectively. Using these two quantities has enable us to utilise two separate

<sup>2</sup>This technique iteratively converges to a set of image filters such that all image filters back-project to surface ‘filters’ with the same centre frequency as each other. This ensures that data from a single band of spatial frequencies on the surface are utilised in estimating surface orientation.

<sup>3</sup>The method used here utilises circular image filters, as described in [10], rather than elliptical image filters as utilised for results presented in [9]; results are not significantly different for these two variants of AMSF.

computational mechanisms which take full advantage of the informational content offered by each of the measured quantities.

Results obtained using real and synthetic images are qualitatively similar to those obtained from psychophysical studies. This indicates that the ABM is at least consistent with the functional characteristics of computational mechanisms responsible for obtaining shape from texture in human observers.

The general principles upon which the ABM is based can be used to estimate surface orientation of curved surfaces. This can be achieved by using multiple circular image regions, and estimating the surface orientation associated with each.

**Acknowledgements:** Thanks to Raymond Lister and Stephen Isard for comments on drafts of this paper.

## References

- [1] Aloimonos J, "Shape from Texture", *Biological Cybernetics*, 58, pp 345-360, 1988.
- [2] Blake A, Marinos C, "Shape from Texture: Estimation, Isotropy and Moments", *Artificial Intelligence*, 45, pp 323-380, 1990.
- [3] Brodatz P, "Textures", Dover Publications Inc., New York, 1986.
- [4] Cutting J E, Millard R T, "Three Gradients and the Perception of Flat and Curved Surfaces", *Journal of Experimental Psychology: General*, 113(2), pp 198-216, 1984.
- [5] Kanatani K, and Chou T, "Shape From Texture: General Principle", *Artificial Intelligence*, 38, 1, pp 1-49, 1989.
- [6] Kube P, "Using Frequency and Orientation Tuned Channels to Determine Surface Slant", *Eighth Annual Conference of the Cognitive Society*, Amhurst, Massachusetts 1986.
- [7] Stevens K A, "The Information Content of Texture Gradients", *Biological Cybernetics*, 42, pp 95-105, 1981.
- [8] Stevens K A, "Surface Tilt: The Visual Encoding of Surface Orientation", *Biological Cybernetics*, 46, pp 183-195, 1983.
- [9] Stone J V, "Shape From Texture: Textural Invariance and the Problem of Scale in Perspective Images of Textured Surfaces", *British Machine Vision Conference*, Oxford, England, pp 181-187, 1990.
- [10] Stone J V, "Shape From Texture: A Computational Analysis", *D.Phil. Thesis*, Experimental Psychology, University of Sussex, England, 1991.
- [11] Turner MR, Gerstein GL, Bacjys R, "Underestimation of visual texture slant by human observers: a model", *Biological Cybernetics*, 65, pp 215-226, 1991.