Shape from texture

John Aloimonos arnd Michael J. Swain
Department of C.Computer Science
The University of Rochester
Rochester, N.Y.14627

Abstract

Measurements on image texture interpreted under an approximate perspective image model can be used with an iterative constraint propagation algorithm to determine surface orientation. An extension of the ideas allows their robust application to natural images of textured planes. The techniques are demonstrated on synthetic and natural images.

1. Introduction

The recover) *of* 3-D (shape) information from a two dimensional image is an important task in image understanding. "Shape from ..." algorithms exist for intensity, motion, contour, and texture. This work extends work on *Shape from Texture.* The shape from texture problem has been studied extensively. Gibson (1950) first proposed the texture density gradient as the primary basis *of* surface perception by humans. Following Gibson's ideas, Bajcsy and Lieberman (1976) tried a heuristic use *of* the two dimensional Fourier spectrum to detect the gradient.

To formalize the shape from texture problem requires a model for the image formation system. Up to now three kinds of projections have been used: orthographic, perspective and spherical projection. Kender (1980), Kanade (1979) and Witkin (1981) studied the problem under orthographic projection, and Kender (1980) and Ohta (1981) address the problem under perspective projection. Ikeuchi (1984) used spherical projection. In his work the texture elements had to be known and symmetrical.

Our algorithm determines surface orientation from the apparent distortion of patterns on the image, provided that:

(1) The surface in view is smooth and is covered with uniformly repeated texture elements. All the texture elements on the surface are identical. These texture elements we call texels. The shape *of* the texels is of no importance for our theory.

(2) Bach texture element is assumed to lie on a plane (i.e. we assume that the surface in view is locally planar). This means that the size of the texels on the surface has to be small compared with a change of surface orientation there.

(3) The scene texture is imaged under an approximation to perspective projection similar to that of Ohta (1981). This projection preserves important perspective distortions but is computationally tractable.

Under the above assumptions, we develop a new gradient map that will enable us to define a "textural reflectance function.". Our theory is very similar to earlier work on Shape from Shading (Horn, 1977; Ikeuchi, 1981), with the image intensity at a point replaced with the area *of* the image texel at that point.

2. Mathematical preliminaries

In this section we define a way to approximate the distortion of a texel under perspective projection by a *2-D* affine transformation.

2.1 Approximation of the perspective projection by a 2T) affine transformation

Let a coordinate system OXYZ be fixed with respect to the camera, with the -Z. axis pointing along the optical axis, and O the nodal point of the eye (center of the lens). The image plane is assumed to be perpendicular to the 7 axis at the point (0,0,-1), (i.e. focal length = 1).

Under perspective projection, a point of the object surface is projected onto the image plane by a projecting ray defined by that point and the center of the lens.

Our approximation is done by dividing the projection process into two steps. (See figure 1). Consider a textured surface S in the world and a texel T on the surface, lying in a local plane Q with orientation given by the surface gradients (p,q). Consider also a plane *y,* parallel to the image plane and just in front *of* the surface S. The plane *y* has distance B from the origin, (i.e. its equation is -Z - *ji).*

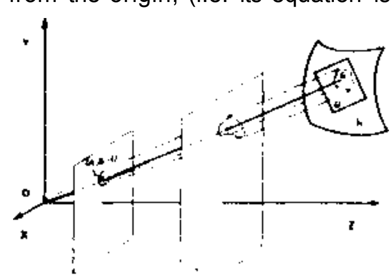

Figure 1

The steps of the projection process of the texel T onto the image plane are as follows:

(1) The surface texel is projected onto the plane *y*. This projection is performed parallel to the ray OG, where G is the mass center of the texel T. (Thus the image of the mass center of the texel is on the projected mass center of the texel, and the projection is parallel to the direction (A,B,-1) where (A,B) is the image of the mass center of the texel T).

(2) The image on the plane *y* is projected perspectively onto the image plane. Since the plane *y* is parallel to the image plane, this perspective transformation is just a reduction by a factor of *Vp.*

Step 1 skews the image to account lor foreshortening, and step 2 scales the image in a location-depth dependent manner, as does perspective. The combined process is an affine transformation.

To represent the original pattern of the surface texel, we use an (a,b,c) coordinate system, with its origin at the mass center of the texel and the (a,b) plane identical to the plane Q To represent the pattern of the image texel, we use an (a,b,c') coordinate system, with its origin the point (A,B, 1), i.e. the mass center of the image texel, and the axes a',b\c' parallel to the axes X,Y,Z respectively. Then the transformation from (a,b) to (a,b') with the two step projection process of previous section is given by the affine transformation.

$$\begin{bmatrix} a' & b' \end{bmatrix} = \begin{bmatrix} a & b \end{bmatrix} \frac{1}{\beta} \begin{vmatrix} \frac{1+pA}{\sqrt{1+p^2}} & \frac{pB}{\sqrt{(1+p^2)}} \\ \frac{q(p+A)}{\sqrt{(1+p^2)(1+p^2+q^2)}} & \frac{qB-p^2-1}{\sqrt{(1+p^2)(1+p^2+q^2)}} \end{vmatrix}$$

It is clear that this transformation is the relation between two 2-D patterns, one in the 3-D space and the other its image on the image plane. We now use the above affine transformation to develop the desired constraint.

2.2 The constraint

The determinant of the matrix of an affine transformation is equal to the ratio of the areas of the two patterns before and after the transformation. Specifically, if S_W is the area of a world texel that lies on a plane with gradient (p,q) and S_I is the area of its image that has mass center (A,B), then we have:

$$\frac{S_I}{S_W} = \frac{1}{\beta^2} \det \begin{vmatrix} \frac{-1+pA}{\sqrt{1+p^2}} & \frac{pB}{\sqrt{(1+p^2)}} \\ -\frac{q(p+A)}{\sqrt{(1+p^2)(1+p^2+q^2)}} & \frac{qB-p^2-1}{\sqrt{(1+p^2)(1+p^2+q^2)}} \end{vmatrix}$$

or

$$S_I = \frac{S_W}{\beta^2} \cdot \frac{1-Ap-Bq}{\sqrt{1+p^2+q^2}} \qquad (1)$$

Equation (1) relates the area of a world texel S_W, its gradient (p,q), the area S_I of its image and its mass center (A,B). If we call the quantity S_I "textural intensity," and the quantity S_W/β^2 "textural albedo," then equation (1) is very similar to the image irradiance equation

V\ + p' + q'

where I is the intensity, (p,q) the gradient of the surface point whose image has intensity 1, A is the albedo at that point and (A,B,1) the direction of the light source (Horn, 1977; Ikeuchi, 1981).

Thus equation (1) can be used to recover surface orientation, using methods that have been discovered for the solution of the shape from shading problem (Ikeuchi, 1981).

2.3 A gradient map

Equation (1) of the previous section can be written as :

/ - R(p,q) (2)

where / is the textural intensity, i.e. the area ol an image texel with mass center (A,B), and

R(p.q) - A. Y-Ap Bq

with A the textural albedo, i.e. the quantity *sw-/B* and (p,q) the gradient *of* the plane on which the world texel lies. The function R(p,q) we call textural reflectance. If we fix the albedo A, and the position (A,B) of the texel on the image, then equation (2) can be represented conveniently as a series of contours of constant textural intensity. Figure 2 illustrates such a simple textural reflectance map.

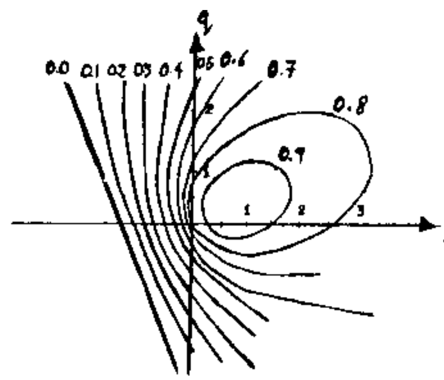

Fig. 2: The textural reflectance map for a point (A,B) = (-.7,-.3) with textural albedo λ=1. The reflectance map is plotted as a series of contours spaced one unit apart.

2.4 Recovering the textural albedo

We use equation (2) of the previous section to recover the local surface orientation. No matter what method we use we must know the textural albedo $\lambda = S_W/\beta^2$.

We cannot know β from a static monocular view; neither can we know S_W in general. But it turns out that we can compute approximately the ratio S_W/β^2, i.e. the textural albedo λ.

Consider three neighboring image texels T_1, T_2 and T_3 with areas I_1, I_2 and I_3 and we suppose that the world texels whose images are the texels T_1, T_2 and T_3 lie on the same plane with gradient (p,q). Then the following equations arise:

$$I_1 = \lambda\,(s_1, n) \quad (3)$$

$$I_2 = \lambda\,(s_2, n) \quad (4)$$

$$I_3 = \lambda\,(s_3, n) \quad (5)$$

where $n = (p,q,1)/\sqrt{1 + p^2 + q^2}$ and $s_i = (-A_i, -B_i, 1)$ for $i = 1,2,3$ and (A_i, B_i) the mass center of texel I_i. Eliminating the textural albedo λ from the equations (3), (4) and (5) we get:

$$n = k[I_1(s_2 \times s_3) + I_2(s_3 \times s_1) + I_3(s_1 \times s_2)]. \quad (6)$$

$$\lambda = 1/\,k[s_1, s_2, s_3], \quad (7)$$

for some constant k that makes n a unit vector, where

$$[s_1, s_2, s_3] = s_1(s_2 \times s_3)$$

and provided that $[s_1,s_2,s_3] \neq 0$, i.e. the vectors s_1, s_2, s_3 are not coplanar (linearly dependent).

The result of equation (7) is approximate due to the hypothesis that three neighboring texels lie on the same plane. But, if we perform this process in all the triples of neighboring points, and we take the average value for the albedo, then the result is highly improved. At the same time, we can get an approximate value for the surface normals at all the texels in the image (equation (6)). Then we can use these initial approximations to start the iterative algorithm that will be introduced in the next section.

2.5 Another way to recover the albedo

Following Ohta *et al.* (1980) and assuming local planarity, i.e. three neighboring texels belong to the same plane which we call Q, we have that:

$$\frac{f_1}{f_2} = \left(\frac{s_1}{s_2}\right)^{1/3}$$

where f_1, f_2 are the distances from two texels to the vaniging line *of* the plane Q along the line joining the two texels and s_1, S_2 are the areas of" the two texels in the image. Since $V_1 - f_2|$ is just the distance between the two texels in the image and it is known, a point on the vanishing line may be determined. With a third texel, two points may be determined, which give the equation of the vanishing line. Since the equation of the vanishing line *of* the plane 0 is px + qy = I, the orientation of the plane Q can be determined, and from that an approximation of the textural albedo is found.

3. Additional constraints and propagation of the constraints

In this section we introduce the smoothness constraint (Ikeuchi, 1981) and we present an iterative algorithm of the same flavour as the one introduced by Ikeuchi.

3.1 An iterative propagation algorithm

We have already proved that every distortion value (image texel area) for a specific image position corresponds to a contour in the gradient space (See section 3.4). So, the problem has infinite solutions (is "ill-posed") and this is the reason that we introduce the smoothness assumption (a "regularization condition"). A smoothness constraint can be used to reduce the locus *of* possible orientations to a unique orientation, through an iterative algorithm.

frying to develop a global error function that should be minimized in order to give the desired value, we measure the departure from smoothness and the error in the textural reflectance equation (equation 2). The error in smoothness we measure (after Ikeuchi, 1981) as follows:

$$s_{i,j} = \frac{(p_{i+1,j} - p_{i,j})^2 + (p_{i,j+1} - p_{i,j})^2 + (q_{i+1,j} - q_{i,j})^2 + (q_{i,j+1} - q_{i,j})^2}{4}$$

where p_{ij} and q_{ij} denote the orientation at the surface poin whose image is the point (i,j). The error in the textura reflectance equation, can be given by :

$$e_{i,j} = (I_{i,j} - R(p_{i,j}, q_{i,j}))^2$$

where $I_{i,j}$ is the distortion value (texel area) at the poin (i,j) and R the textural reflectance.

An acceptable solution should minimize the sum of th error terms in all the grid nodes. If E is such a global erro function, then

$$E = \sum\sum (s_{i,j} + \omega e_{i,j})$$

and the factor ω gives a weight to the errors in th textural gradient map relative to the "distance" fron smoothness. To minimize E, we differentiate with respec to $p_{i,j}$ and $q_{i,j}$, and setting the resulting derivatives to zer and rearranging the equations we obtain:

$$p_{i,j} = pa_{i,j} + \omega\,[I_{i,j} - R(p_{i,j}, q_{i,j})]\ \partial R/\partial p$$

$$q_{i,j} = qa_{i,j} + \omega\,[I_{i,j} - R(p_{i,j}, q_{i,j})]\ \partial R/\partial q$$

where $pa_{i,j}$ and $qa_{i,j}$ are the average values of p and q around the point (i,j) respectively. The above equations suggest an adjustment of p and q in the direction of the gradient of the textural reflectance function, by an amount that is proportional to the error in the textural reflectance equation (equation 2). So it is natural to use the following iterative rule for the estimation of the p and q everywhere in the image:

$$p_{i,j}^{n+1} = pa_{i,j}^{n} + \omega\,[I_{i,j} - R(p_{i,j}^{n}, q_{i,j}^{n})]\ \partial R/\partial p$$

$$q_{i,j}^{n+1} = qa_{i,j}^{n} + \omega\,[I_{i,j} - R(p_{i,j}^{n}, q_{i,j}^{n})]\ \partial R/\partial p$$

In the above equations the partial derivatives of the textural reflectance are evaluated on the values of p and q of the n-th iteration. Finally, to avoid numerical instabilities we modify the above formulas to the following form (Ikeuchi & Horn, 1981):

$$p_{i,j}^{n+1} = pa_{i,j}^{n} + \omega\,[I_{i,j} - R(pa_{i,j}^{n}, qa_{i,j}^{n})]\ \partial R/\partial p$$

$$q_{i,j}^{n+1} = qa_{i,j}^{n} + \omega\,[I_{i,j} - R(pa_{i,j}^{n}, qa_{i,j}^{n})]\ \partial R/\partial p$$

$R(p\text{-}q_{i,j'})$ is a function on a four-dimensional space, unlike the R(p,q) of orthographic Shape from Shading. The shading R(p,q) can be determined empirically, but the textural reflectance $R(p_{ir}q_{tj})$ is an analytic, geometrical entity arising from imaging geometry, and thus only the global constant (texture) albedo varies from texture to texture and scene to scene.

4. Experiments

The algorithm was tested on artificial images of a plane, cylinder and sphere. There are four distinct steps into which the program may be broken down:

1) Location of texels
2) Minimum triangulation *of* the texel centers
3) Calculation *of* initial orientations and textural albedo.
4) Iterative process.

In 1), the connected regions in the image are detected. Their centers of gravity are taken to be the locations of the texels. Their size is recorded and the texels which are in the boundary are marked (Ballard & Brown, 1982). In 2), the points denoting the centers *of* the texels are triangulated so that the sum of the length of the lines is minimum (Aho, Hopcroft & Ullman). In 3), the initial orientations were calculated using the method in Section 3.6. The estimate of A was calculated from the local orientation with the lowest value *of* p and q. Due to curvature of the surface, convex objects tend to give an overestimate of A while concave objects tend to give an underestimate. These errors are minimized when the surface of the object is most nearly perpendicular to the image plane. The algorithm is quite insensitive to initial orientations given to texels whose orientations were allowed to vary through the iterative process. Boundary texels were not allowed to change. The error in calculating their values was the predominant factor in influencing the total error. The iterative process took under 10 iterations. The process always converged for our synthetic images. The final error values were

	fractional error
plane	negligible
sphere	.005
cylinder	.015

The errors in the above table denote the average percent error at each texel. The error at each texel was taken to be

$$1/4\pi * \varphi .$$

where φ = solid angle subtended by rotating the calculated orientation about the actual orientation. Figure 3 gives a pictorial description of the error at each texel.

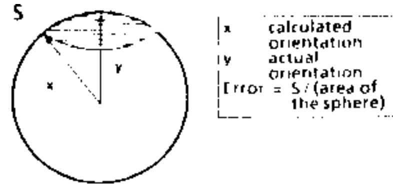

Figure 3.

Finally, azmuthal equidistant coordinates (AHC) were used through the iterative process instead of the gradient space p and q, since AHC change linearly with change in orientation. Higure 4 shows the image of a sphere which is covered with a repeated pattern, Higure 5 shows the reconstructed sphere using the algorithms *of* Sections 3.5 and 3.6, and Higure 6 shows the reconstructed sphere after the relaxation. Figures 7, 8, and 9 and 10, 11, and 12 show the analogous pictures for a cylinder and plane respectively.

Figure 4

Figure 5

Figure 6

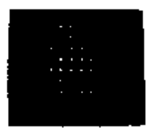

Figure 7

Figure 8

Figure 9

Figure 10

Figure 11 12

5. An extension to natural images

In natural images, the assumption that the world surface is covered with the same texels is not very realistic. This section refers to recent work by Aloimonos and Chou (1985) on shape from the images *of* textured planes, based on the uniform density assumption under strong segmentation (identification *of* texels) and weak segmentation (edge finding). Consider a world plane - / - px + qy -F c, and the plane *y* passing through the center of mass of a small area s of the world plane. If we consider an area S_1 in the image, then in order to find the area in the world plane whose projection is S_1, we must multiply the area s_1 with the factor

$$R_I = abs\left(\frac{c^2\sqrt{1+p^2+q^2}}{(1-Ap-Bq)^3}\right)$$

where (A,B) is the center of gravity of the image area S_I.

5.1 Exploiting the uniform density assumption

The uniform density assumption states that if K, Λ are any two regions in the world plane and they contain K_I and K_2 texels respectively, then $K_I/\text{area}(K) = K_2/\text{area}(\Lambda)$. So considering any two regions s_I and s_2 in the image with areas S_I and S_2 that contain K_I and K_2 texels respectively, then under the assumption that the world texels are uniformly distributed, we have $K_I/S_IR_I = K_2/S_2R_2$ with

$$R_1 = abs\left(\frac{c^2\sqrt{1+p^2+q^2}}{(1-A_1p-B_1q)^3}\right) \quad \text{and} \quad R_2 = abs\left(\frac{c^2\sqrt{1+p^2+q^2}}{(1-A_2p-B_2q)^3}\right)$$

and (A_1,B_2), (A_2,B_2) the centers of gravity of the image regions S_I and S_2 respectively.

From this we get:

$$\left[\left(\frac{K_2}{K_1}\frac{S_1}{S_2}\right)^{\frac{1}{3}} A_2 - A_1\right] p + \left[\left(\frac{K_2}{K_1}\frac{S_1}{S_2}\right)^{\frac{1}{3}} B_2 - B_1\right] q = \left(\frac{K_2}{K_1}\frac{S_1}{S_2}\right)^{\frac{1}{3}} - 1.$$

The above equation represents a line in p-q space. Any two regions in the image constrain (p,q) to lie on a line in the gradient space. Thus, taking any two pairs of regions we can solve explicitly for p and q. (To overcome undesirable results due to errors from the digitization process and the density fluctuations of the regions, we employed a least-square-fit mechanism by considering several image pairs. A Hough transform estimation method might also be appropriate.) Figure 13 is the image of a plane parallel to the image plane, covered with random dots (texels). Figure 14 is the image of the dotted plane rotated and translated with tilt - 135° and slant - 30°. Our program, based on the scheme described above, recovered tilt = 134.4° and slant = 29.75°.

Figure 13

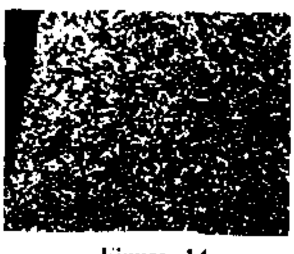

Figure 14

5.2 Solving the problem with a weaker segmentation

In the previous section a method was developed to recover the orientation of a textured plane from its image, based on the assumption *of* uniform texel density; by uniform density, we mean that the number of texels per unit area of the world plane is the same. Application of this method to natural images did not seem to work very well, because no good methods have been developed up to now that can identify texels in an image, and this algorithm depends critically on the number of texels per unit area. On the other hand, in the recent literature (Bandyopadhyay, 1984; Ballard & Brown, 1982; Marr, 1979) there are several methods for the computation *of* partial boundaries of the texels (edges) at every point in a textured image. Fet us redefine density to be the total length of the texel boundaries per unit area. The uniform densitv assumption states that this density is the same everywhere in the world plane. This assumption is not far from the previous one and seems to be true for a large subset of natural images, in contrast with Witkin's isotropy assumption, which does not seem to hold true for many natural images. Aloimonos & Chou (1985) describe a method that Finds the orientation of the world plane under the new assumption and below we describe expriments based on that method. Figure 15 presents the image of a plane parallel to the image plane, covered with random line segments. Figure 16 presents the image *of* this plane rotated with tilt 135° and slant - 30°. The program recovered tilt = 133.77° and slant - 30.40°. Figure 17 presents the image of a plane (parallel to the image plane) covered with randomly generated small circles. Figure 18 presents the image of this plane rotated with tilt - 135° and slant - 30°. The program recoverd tilt - 135.54° and slant = 29.77°. The natural images used were first preprocessed to find the boundaries of texels (edges) by applying the modified Frei-Chen operators introduced by Bandopadhyay (1984). Figure 19 shows the photograph of a textured floor with slant = 45° and tilt - 108°. Figure 20 shows the edges after the preprocessing. The algorithm produced slant = 45.87° and lilt = - 109.43°. Finally, Figure 21 shows the photograph of a part of a grass field with slant = 60° and tilt -- 0°. Figure 22 shows the images of its edges after the preprocessing. The program recovered slant - 63.057° and tilt 1.076°.

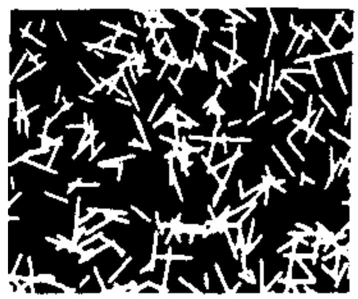

Figure 15

Figure 16

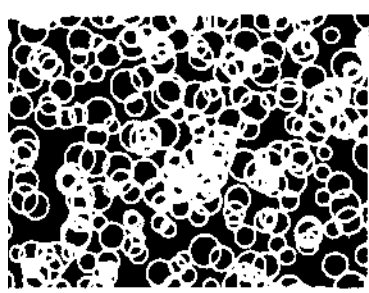

Figure 17

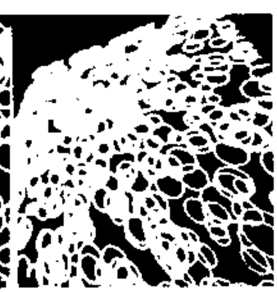

Figure 18

Figure 19

Figure 20

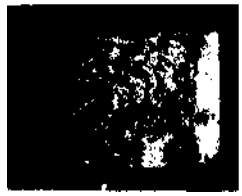

Figure 21

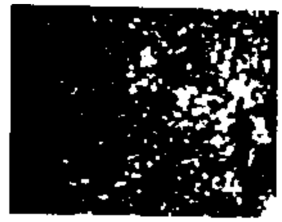

Figure 22.

Aloimonos and Chou (1985) present a theoretical analysis of the error introduced by the affine approximation to the perspective projection.

Conclusion

In tins paper we gave a method for "shape from texture" computation that extends results in the literature. Our algorithm can work in a richer domain than already published methods. For reasons involving the cosine foreshortening law, the result here is similar to the original Shape from Shading constraint applied to lambertian surfaces. We believe that this may have some important implications, since for example the same hardware may be used for shape recognition, regardless of whether we use shading or texture. Finally, an extension of the algorithm to a form that works well for many natural images was presented.

Acknowledgments

We would like to thank Christopher Brown for his constructive criticism. Our thanks also go to Dana Ballard, Amit Bandyopadhyay and Paul Chou for their help during the preparation of this paper. This research was sponsored by the National Science Foundation under Grant MCS-8302038.

References

|1| Aho, A.V., Hoperoft, J.I". and Ullman, J.I)., *The Design and Analysis of Computer Algorithms.* Addison Wesley Publishing Co., Reading. MA.. 1974

(2) Aloimonos, J. and Chou, P.. "Detection of' surface onenialion and motion from texture," forthcoming Technical Report, Dept. of Computer Science, Univ. of Rochester

[3] Bandyopadhyay. A., "Interesting points, disparities and correspondence," *Proceedings,* DARPA Image Understanding Workshop, October 19X4.

|4] Bajcsy, R. and Lleberman, I ., "Texture gradients as a depth cue." *Comp. Graphics and Image Processing* 5, 1976, 5?-67

[5] Ballard, D.H. and Brown, CM., *Computer Vision,* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1982.

[6) Gibson, J.J., *The Perception of the Visual World,* Houghton Miffin, Boston, 1950.

|7| Horn, B.K.P., "Understanding image intensities." *Artificial Intelligence* 8, (2), 1977, 201-231.

[8] Ikeuchi, K., "Shape from regular patterns." *Artificial Intelligence* 22, 1984, 49-75.

|9| Ikeuchi, K. and Horn, B.K.P., "Numerical shape from shading and occluding boundaries." *Artificial Intelligence* 17, 1981. 141-184.

|10| Kanadc, I., "Recovery of the three dimensional shape of an object from a single view," CMU-CS-79-15.1. Dept. of Comp. Science. Carnegie-Mellon Univ., 1979.

|11| Kender, J.R., "Shape from texture: an aggregation transform that maps a class of textures into surface orientation," *Proceedings,* 1JCA1, 1980, 475-480.

[12] Kender, J.R, "Shape from texture: A computational paradigm." *Proceedings,* DARPA Image Understanding Workshop, April 1979, 79-84.

[13] Marr, D. and Hildreth, H., "A theory of edge detection." AI Memo 518, MIT, 1979.

|14| Ohta, Y., Maenobu, K.. and Sakai, T., "Obtaining surface orientation of from texels under perspective projection," *Proceedings,* IJCAI, 1980, 746-751.

[15] Witkin. A.. "Recovering surface shape and orientation from texture," *Artificial Intelligence* 17, 1981, 17-45.