Surface Orientation from a Projected Grid

NEELIMA SHRIKHANDE AND GEORGE STOCKMAN

Abstract—**Two simple methods are given for obtaining the surface shape of objects using a projected grid. After the camera is calibrated to the 3-D workspace, the only input data needed for the computation of surface normals are grid intersect points in a single 2-D image. The first method performs nonlinear computations based on the distortion of the lengths of the grid edges and does not require a full calibration matrix. The second method requires that a full parallel projection model of the imaging is available, which enables it to compute 3-D normals using simple linear computations. The linear method performed better overall in the experiments, but both methods produced normals within 4–8 degrees of known 3-D directions. These methods appear to be superior to methods based on shape-from-shading because the results are comparable, yet the equipment setup is simpler and the processing is not very sensitive to object reflectance.**

Index Terms—**Active sensing, computer vision, needle map, shape-from, structured light, surface orientation.**

I. Introduction

The problem of inferring 3-D surface shape/orientation from a single 2-D image has been approached in different ways including shape-from-shading [8], [13], [4], shape-from-motion [12], [5], shape-from-stereo [7], and shape-from-texture [16], [3], [11]. Below, two simple methods are proposed for computing surface orientation from the image of a grid of a light projected onto object surfaces. The algorithms are similar to shape-from-texture methods except that the texture is externally imposed on the object via structured light and cannot be assumed to be an intrinsic characteristic of the object surface itself. Structured light had been used as early as 1970 [16] for detection of planar faces and has become very popular in industry for making 3-D measurements. Much of current industrial vision research is proceeding with the use of 3-D sensed data; and, in fact, we are also doing 3-D sensing via structured light. However, here we show that good 3-D surface shape information can be computed with a simple active lighting arrangement and without computation of actual 3-D surface points. Both planar and curved surfaces can be handled. Computed normals, often accurate to within a few degrees, are obtained for smooth curved surfaces such as a Coke can as well as for simpler planar surfaces.

II. The Sensing Environment

Fig. 1 gives a sketch of the sensing environment used. Our workspace is roughly 30 × 30 × 15 cm on a table top, and objects used for experiments are roughly of diameter 15 cm or smaller. (Conceptually, there should be no problem scaling the system for sensing objects of other sizes.) A grid of light is projected vertically downward onto the scene, which is viewed by a camera whose axis is roughly 45 degrees offset from the projector axis. Also, for stability in the computations, the optical axis makes blunt angles with both X light sheets and Y light sheets. The focal lengths of the lenses are 102 mm for the projector and 50 mm for the camera.

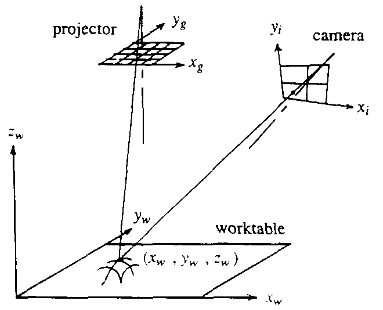


Fig. 1. Sensing environment.

Both camera and projector have standoff of about 150 cm, allowing the parallel projection model to be used for each of them. For convenience, the world coordinate frame in which surface normals are ultimately reported, is aligned with respect to the projected grid. The z_w axis is perpendicular to the worktable and along the optical axis of the projector. The x_w and y_w coordinates are measured along the X and Y sheets of the grid, respectively. For the experiments reported here, the grid spacing was about 10 mm on the worktable. As discussed in Asada *et al.* [1], the error of approximating the perspective projection by parallel projection is small if the standoffs of camera and projector are at least 20 times the object size. In our experimental environment, viewing a 6 cm diameter Coke can from 150 cm away gives a ratio of 25 to 1. However, with our 15 cm high cobra sculpture, the ratio is only 10 to 1 and we should expect some observable differences. Further discussion of these effects is given later in the paper.

The camera produces a TV signal which is then digitized for processing on our VICOM 1800 image processor. A look ahead at Figs. 3 and 4 shows what kind of images result (ignore the computed normals for now). Although we do have software for isolating the grid point intersections in the image [9], for convenience, we identified them by hand using the system mouse and cursor in this research.

Geometrically, the imaging transformation maps an entire ray of 3-D points (x_w, y_w, z_w) onto image point (x_i, y_i). The corresponding algebraic statement is given in (1). We work in units of mm for 3-D coordinates and pixels (integers) for image coordinates.

When the camera standoff is large relative to the focal length and object size, the imaging process may be approximated by a parallel projection (of simpler form than a perspective projection), as shown in (1). A geometric interpretation of the imaging process is as follows. The 3-D world points are projected onto a plane (cross section) that is perpendicular to the optical axis, forming a projection of the scene at the same scale. This projection is then mapped into the actual image plane by accounting for any rotation of the image and scaling due to the spacing of sensor rows and columns, and translation due to the location of the camera axis in the world space. The intrinsic parameters of the parallel projection are thus the camera axis direction (2 parameters), the rotation of the image (1), scale factors in the projection plane (2), and the offset of the image origin in the projection plane (2), but not focal length. These seven parameters are encoded in the elements c_{ij} of the matrix modeling the projection.

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \boldsymbol{P} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}. \quad (1)$$

This camera matrix form $\boldsymbol{P}$ can be fit to calibration data points $[(x_w, y_w, z_w), (x_i, y_i)]$ using least squares to get the *best* set of

Manuscript received February 23, 1988; revised December 30, 1988. Recommended for acceptance by O. D. Faugeras. This work was supported in part by the National Science Foundation under Grants IRI-8743697 and DCR-8600371.

N. Shrikhande is with the Department of Computer Science and the Center for Computer Vision and Robotics Research, Central Michigan University, Mount Pleasant, MI 48858.

G. Stockman is with the Department of Computer Science, Michigan State University, East Lansing, MI 48824.

IEEE Log Number 8927497.

elements c_{ij} and is well documented in other works [2], [6]. There are also simpler procedures possible; for example, approximate elements can be obtained from the observation that $c_{11} = \partial x_i / \partial x_w$, $c_{23} = \partial y_i / \partial z_w$, etc.

III. Computational Model

The objective is to determine the normal N_p at surface points of the 3-D scene. We assume that the surface varies slowly with respect to the grid spacing so that the grid intersects the surface approximately in quadrilaterals. These are viewed in the image taken by a camera as distorted quadrilaterals (Figs. 2-4). The amount of distortion is dependent on the normal N_p to the plane and the viewpoint. Two adjacent sides of the image quadrilaterals provide the information necessary to solve for the normal. The direction of the normal is derived with respect to the world coordinate system, and not the camera coordinate system. The following assumptions are made in developing the computation.

Camera Assumptions: Parallel projection is assumed to be an accurate camera model. We assume that the camera model is known; or, for one algorithm, just the camera axis and the x and y scale factors are assumed known.

Projector Assumptions: The projector axis is parallel to the z_w-axis. The projector is sufficiently far removed from the scene to allow the parallel projection model. The projector projects mutually perpendicular sheets of light on the scene. The X sheets have normals parallel to the y_w-axis and the Y sheets have normals parallel to the x_w-axis. The distance between successive X sheets equals the distance between successive Y sheets which we call the grid size d (this last assumption is not needed in Algorithm 2).

Object Assumptions: The objects are opaque and are smooth relative to the grid spacing.

A. Algorithms for Computing Normals

Let N_p, N_x, and N_y denote the normals to planes P, the X sheets, and the Y sheets, respectively. Let N_a be the unit vector in the direction of the optical axis of the camera. (Remember that P is only approximately planar in the case that the surface is curved.) See Fig. 2.

$$\begin{array}{ll} N_p = (l, m, n) & \text{unknown} \\ N_x = (0, 1, 0) & \text{assumption} \\ N_y = (1, 0, 0) & \text{assumption} \\ N_a = (a_x, a_y, a_z) & \text{assumption} \end{array} \tag{2}$$

Step 1: Compute the size of the projected grid in 3-D.

Let $L_1 = P \cap X$ be the (free) 3-D vector denoting the intersection of the plane P and the portion of an X sheet between two successive Y sheets. Similarly, let $L_2 = P \cap Y$.

Let U_{L_1} and U_{L_2} denote the unit vectors along L_1 and L_2, respectively. Then

$$U_{L_1} = \frac{N_x \times N_p}{|N_x \times N_p|}$$

and

$$|L_1| = \frac{d}{|U_{L_1} \cdot N_y|}.$$

Therefore,

$$L_1 = \frac{d}{|U_{L_1} \cdot N_y|} U_{L_1} = \left(d, 0, \frac{-ld}{n}\right)$$

similarly,

$$L_2 = \frac{d}{|U_{L_2} \cdot N_x|} U_{L_2} = \left(0, d, \frac{-md}{n}\right).$$

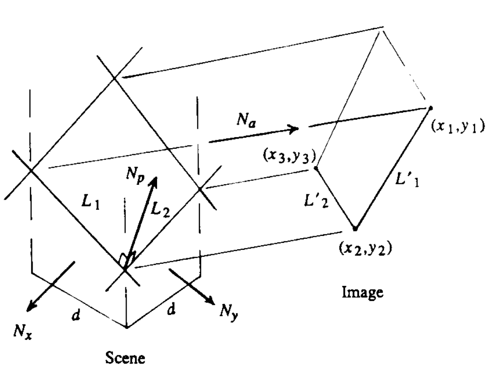


Fig. 2. Imaging a single projected grid cell.

Step 2: Compute the relation between L_1, L_2, and their images L_1' and L_2'. Let L_1' and L_2' be the projections of L_1 and L_2, respectively. By the parallel projection assumption,

$$L_1' = L_1 - (L_1 \cdot N_a)N_a$$

and

$$L_2' = L_2 - (L_2 \cdot N_a)N_a.$$

B. Algorithm 1

Compute Surface Normals via Lengths of the Sides of the Image Quadrilaterals:

If r_1 and r_2 denote the lengths of L_1' and L_2' in the image plane, then

$$\begin{aligned} |L_1'| &= r_1 \\ |L_2'| &= r_2 \\ |N_p| &= 1. \end{aligned} \tag{3}$$

These are three nonlinear equations in three unknowns which can be solved to obtain N_p. See Appendix A for for details of the mathematical derivations. The x_w component of the normal is dependent only on the length of L_1' and the y_w component of the normal is dependent on the length of L_2'. The lengths r_1 and r_2 are measured in image coordinates (pixels) and must be translated into world coordinates using scale factors from calibration. Thus the direction of the camera axis, the lengths of adjacent sides of an imaged grid cell and the scale factors are needed to execute this algorithm. In general, there are multiple solutions to (3). The number of solutions can be reduced to two by simple considerations. A unique solution can be obtained by further calibration in most cases. The details of the ambiguity resolution are given in Appendix B.

$$N_p = f_1(r_1, r_2) \tag{4}$$

where f_1 is constructed using parameters N_a, S_x, S_y gotten once from calibration.

B. Algorithm 2

Compute Surface Normals via the Full Calibration Matrix:

Note that L_1 is the intersection of two planes: an X sheet and the back projection of L_1' into the 3-D workspace. Let $L_1 = (x_w, 0, z_w)$: (as a free vector in world coordinates). L_1' is easily computed from L_1 by projection of its endpoints. Location information is factored out.

$$L_1' = \begin{bmatrix} x_1 - x_2 \\ y_1 - y_2 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{bmatrix} \begin{bmatrix} x_w \\ 0 \\ z_w \end{bmatrix} \tag{5}$$

If all the parameters c_{ij} are known and L_1' is known then the two equations in the two unknowns x_w and z_w can be solved to obtain

L_1. L_2 can be obtained in the same manner. The surface normal N_p can then be computed via (6).

$$N_p = \frac{L_1 \times L_2}{|L_1 \times L_2|} = f_2(L_1', L_2') \qquad (6)$$

where f_2 is constructed using the full calibration matrix $\boldsymbol{P}$ matrix given in (1).

IV. Results of Experiments

Both Algorithm 1 and Algorithm 2 were tested with real objects using our structured light bench as described in Section II. Several planar faced objects were used so that ground truth data could be easily obtained for evaluating the accuracy of the two algorithms. In addition, three objects with curved surfaces were used—a Coke can (Fig. 3), an orange (Fig. 4), and a clay sculpture (Fig. 4). For Algorithm 1, calibration consisted of manually imaging premeasured targets and noting corresponding x and y pixel positions in the image to get the scale factors and camera axis. For Algorithm 2, the parallel projection matrix C was obtained via calibration as in [6].

Tables I and II summarize the accuracy of the two algorithms on six planar faces. From 10 to 20 normals were computed for each planar face so that an average error could be stated, and an error of the average computed normal could be defined and stated. Comparison of the tables reveals the Algorithm 2 is generally better than Algorithm 1 but not uniformly better. Both algorithms give accuracy comparable to other similar methods reported in the literature. Ray *et al.* reported an accuracy of 5 degrees using photometric stereo with 3 carefully matched light sources [13]. Accuracies of 6–7 degrees under ideal conditions using multiple diffuse light sources have been reported by Capionetti *et al.* ([4] and personal correspondence). Wang *et al.* have stated an accuracy of about 5 degrees using a method of separate projection of X and Y sheets on the scene [15].

Results of computing normals for the curved objects are shown via the needle maps in Figs. 3 and 4. The normals (needles) placed at the grid points are computed using two neighboring grid points along the X and Y light stripes, but not the curved path between them. Thus, the normal is computed as if the surface were a plane passing through the three 3-D surface points. For the data shown in Fig. 4, we had actual 3-D surface locations for each grid point. These were obtained from a previous experiment [9]. Using the 3-D surface locations to define "ground truth planes," the accuracy of the normals (computed from only the three 2-D image points), was comparable to the accuracy described above for the planar surfaces.

V. Error Analysis

Space does not permit the development of the error analysis for each computation and the reader is referred to [14] for details. A summary of only the main points is given here. For both the algorithms the error in computing N_p is a function of both N_p and the view direction N_a. With Algorithm 1, for every position of the camera N_a there is some surface orientation N_p for which the computation of N_p will be unstable. For Algorithm 2, it is easy to see that there are views which should be avoided in order to avoid large error. If N_a is nearly perpendicular to N_x (i.e., the dot product of (a_x, a_y, a_z) and $(0, 1, 0)$ is approximately 0) then the computation of the intersection of the X-sheet and the imaging plane through L_1' used to derive L_1 will be unstable. Similarly, we do not want to have N_a and N_y nearly perpendicular. Therefore, to avoid large computational error, we must set up the environment so that $|a_x|$, $|a_y| > \delta > 0$.

One source of error lies in detecting the grid point intersections in the image. Also, the length of side of a quadrilateral on a Coke can projected in the image will vary roughly one pixel due to the parallel projection approximation. This effect will be worse with the larger cobra sculpture. Rather than pursue analytical error characterization, we examined the effect of error due to image point location error by adding a uniform random error to the original measurements and repeating the computation of the surface normals. Pixel displacements of -1, 0, or $+1$ were equally likely, and were assigned to both the x_i and y_i coordinates of the image points. Results of the new computations are summarized in Tables III and IV. For both algorithms the error in the computed normals increased by about a third overall, with the error for Algorithm 1 being contained within 9 degrees and the error for Algorithm 2 being contained within 7 degrees. We note that techniques for intersection detection giving subpixel accuracy are expected to improve, not degrade, the accuracy reported in Tables I and II.

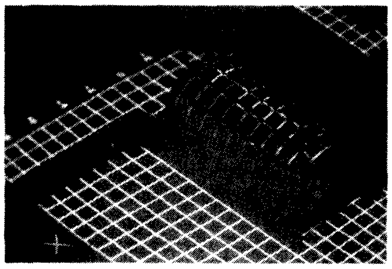


Fig. 3. Coke can image with computed normals overlaid.

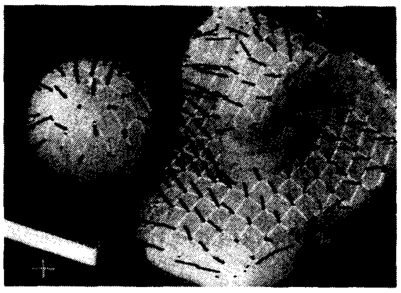


Fig. 4. Orange and cobra image with computed normals overlaid.

TABLE I
Differences Between Ground Truth and Normals Computed Via Algorithm 1

Plane	Ground Truth Normal	Measured Normal (average)	Average Error (in degrees)	Angle Between Average Normal and Ground Truth
1	(0,0,1)	(-.06,.05,.99)	5.98	4.51
2	(0,0,1)	(-.01,.00,.99)	4.42	0.58
3	(0,0,1)	(-.02,.04,.99)	6.61	2.58
4	(-.49,.50,.70)	(-.51,.55,.64)	6.90	4.67
5	(.50,.50,.70)	(.49,.52,.69)	3.12	1.40
6	(.02,.74,.66)	(.17,.73,.64)	8.78	8.79

TABLE II
Differences Between Ground Truth Normals and Normals Computed Via Algorithm 2

Plane	Ground Truth Normal	Measured Normal (average)	Average Error (in degrees)	Angle Between Average Normal and Ground Truth
1	(0,0,1)	(-.02,.10,.99)	3.56	1.29
2	(0,0,1)	(.04,.08,.99)	3.27	1.40
3	(0,0,1)	(-.02,.00,.99)	3.69	1.15
4	(-.49,.50,.70)	(-.47,.55,.69)	3.73	3.10
5	(.50,.50,.70)	(.48,.53,.68)	4.08	2.33
6	(.02,.74,.66)	(.01,.67,.68)	6.18	3.73

VI. Concluding Discussion

We have described two algorithms to compute the surface orientations from a single image using striped lighting. These algorithms use the measurements of sides of the imaged grid to obtain

TABLE III
COMPARISON OF NORMALS COMPUTED AFTER RANDOM VARIATIONS OF UP TO ONE PIXEL MADE IN IMAGE COORDINATES (ALGORITHM 1)

Plane	Ground Truth Normal	Measured Normal (average)	Average Error (in degrees)	Angle Between Average Normal and Ground Truth
1	(0,0,1)	(-.06,-.02,.99)	7.75	3.65
2	(0,0,1)	(-.05,.09,.99)	8.93	5.93
3	(0,0,1)	(.03,.02,.99)	7.17	2.08
4	(-.49,.50,.70)	(-.51,.55,.65)	7.84	4.67
5	(.50,.50,.70)	(.53,.55,.64)	5.78	4.81
6	(.02,.74,.66)	(-.01,.68,.72)	9.07	5.20

TABLE IV
COMPARISON OF NORMALS COMPUTED AFTER RANDOM VARIATIONS OF UP TO ONE PIXEL MADE IN IMAGE COORDINATES

Plane	Ground Truth Normal	Measured Normal (average)	Average Error (in degrees)	Angle Between Average Normal and Ground Truth
1	(0,0,1)	(-.07,.02,.99)	4.49	4.20
2	(0,0,1)	(-.05,.04,.99)	5.21	3.70
3	(0,0,1)	(0,.11,.99)	7.01	6.34
4	(-.49,.50,.70)	(-.46,.53,.70)	4.08	2.45
5	(.50,.50,.70)	(.48,.53,.67)	7.17	2.58
6	(.02,.74,.66)	(-.02,.67,.72)	6.92	5.81

the normal to the surface on which the grid is projected. The setup is simple and inexpensive. Except during calibration, there are no 3-D points to be input to the computations and there are no 3-D locations computed. The algorithms do not need to know exactly which stripes are actually being imaged to form the grid cells. Instead, use of the parallel projection allows computation of the normal direction vector regardless of where it is actually applied in the 3-D space.

Examination of the computations of Algorithm 2 reveals that the endpoints of L_1' and L_2' are not assumed to be grid intersects. Thus for curved surfaces, these endpoints used in the computation could be pulled in closer to the central grid intersect to create an approximating plane that is closer to the ideal tangent plane. Unfortunately, the resulting computations will be more sensitive to errors in image point locations, and thus a limiting process cannot be pursued very far.

The paper by Asada *et al.* [1], published while this correspondence was under review, reports a similar procedure for computing surface normals from structured light. A set of alternating bright and dark parallel stripes are projected onto objects. Surface normals can be computed at points along the edges of the stripes. Denser sets of normals can be computed from their method than for the method given here with a slight increase in processing and in the error of measurement. Some work in aggregating surface elements into planar faces is also given.

There are at least three techniques for computing surface orientation from structured light—although the computational methods differ, the environments and reported accuracies are similar [1], [14], [15]. It is important to contrast the use of structured light, with the currently described method as a particular example, to the use of of shape-from-shading. First of all, by using structured light only a sparse set of normals is produced—roughly 1 normal per 100 pixels in Figs. 3 and 4. However, variations in object albedo causes little problem, e.g., refer to the Coke can in Fig. 4, because only a binary image is being processed. The results indicate that accuracy as good as or better than that for shape-from-shading can be obtained without as much illumination design and intensity calibration. If a higher density of surface normals is required, a set of grids offset from each other can be used. The photometric stereo methods use multiple images as well—typically 3. Along another line, the projected grid gives the strong nonlocal information about surface continuity and connectivity easily supporting higher level grouping operations. It is therefore our conclusion that the structured light techniques are superior to the shape-from-shading technique in environments where active lighting is feasible.

APPENDIX A
ALGEBRAIC DETAILS OF ALGORITHM 1

The details of the mathematical model of Algorithm 1 are given here. Refer to Fig. 2. Let N_x, N_y, and N_p denote the normals to the X sheet, Y sheet, and the plane P, respectively. Then

$$L_1 = P \cap X.$$

Therefore the unit vector U_{L_1} along L_1 is given by:

$$U_{L_1} = \frac{N_x \times N_p}{|N_x \times N_p|} = \frac{(0, 1, 0) \times (l, m, n)}{|(0, 1, 0) \times (l, m, n)|} = \frac{(n, 0, -l)}{\sqrt{n^2 + l^2}}.$$

Further,

$$\frac{d}{|L_1|} = \cos\theta_1 = U_{L_1} \cdot (1, 0, 0) = \frac{n}{\sqrt{n^2 + l^2}}$$

which implies that

$$|L_1| = \frac{d\sqrt{n^2 + l^2}}{n}.$$

Therefore we can write the vector L_1 as:

$$L_1 = |L_1| U_{L_1} = \left(\frac{d,\ 0,\ \pm ld}{n}\right). \tag{1}$$

Similarly, we can write L_2 as

$$L_2 = \left(\frac{0,\ d,\ \pm md}{n}\right). \tag{2}$$

Let $N_a = (a_1, a_2, a_3)$ be the unit normal along the camera axis. The projections of L_1 and L_2 on the image plane perpendicular to N_a are given in L_1' and L_2' respectively; where

$$L_1' = L_1 - (L_1 \cdot N_a)N_a$$

and

$$L_2' = L_2 - (L_2 \cdot N_a)N_a$$

since

$$(L_1 \cdot N_a) = \left(d, 0, \pm\frac{ld}{n}\right) \cdot (a_1, a_2, a_3) = da_1 \pm \frac{lda_3}{n}$$

$$(L_1 \cdot N_a)N_a = \left(da_1 \pm \frac{lda_3}{n}\right)(a_1, a_2, a_3)$$

$$= \left(da_1^2 \pm \frac{lda_1^2}{n},\ da_1a_2 \pm \frac{da_3a_2}{n},\ da_3a_1 \pm \frac{lda_3^2}{n}\right).$$

Therefore, we get the equation

$$L_1' = L_1 - (L_1 \cdot N_a)N_a = \Bigg(d - da_1^2 \pm lda_1\frac{a_3}{n},\ -da_1a_2 \pm lda_3\frac{a_2}{n},\ \pm\frac{ld}{n} - da_3a_1 \pm \frac{lda_3^2}{n}\Bigg)$$

$$= d\Bigg((1 - a_1^2) \pm \frac{la_1a_3}{n},\ -a_1a_2 \pm \frac{la_2a_3}{n},\ \pm\frac{l}{n}(1 - a_3^2) - da_3a_1\Bigg).$$

Squaring and dividing by d^2 and combining like terms, we get the following equation:

$$l^2k_2 + 2lnk_3 + n^2k_1 = 0$$

where

$$k_1 = 1 - a_1^2 - \frac{|L_1'|^2}{d^2};\quad k_2 = 1 - a_3^2;\quad k_3 = \pm a_1a_3. \tag{3}$$

Solving for l, using the quadratic formula, we get

$$l = \frac{-2k_3 n \pm \sqrt{4k_3^2 n^2 - 4n^2 k_1 k_2}}{2k_2} = n \cdot s_1 \tag{4}$$

where

$$s_1 = \frac{-k_3 \pm \sqrt{k_3^2 - k_1 k_2}}{k_2}.$$

Symmetrically, we get equations for m:

$$m^2 k_2 + 2mnk_5 + n^2 k_6 = 0$$

and

$$m = \frac{-k_5 \pm \sqrt{k_5^2 - k_2 k_6}}{k_2} \cdot n \tag{5}$$

where

$$k_5 = \pm a_2 a_3 \quad \text{and} \quad k_6 = 1 - a_2^2 - \frac{|L_2'|^2}{d^2} \tag{6}$$

and

$$m = s_2 \cdot n$$

where

$$s_2 = \frac{-k_5 \pm \sqrt{k_5^2 - k_2 k_6}}{k_2}.$$

Since $l^2 + m^2 + n^2 = 1$, the normal $N_p = (l, m, n)$ to the plane P is given by

$$n = \frac{1}{\sqrt{1 + s_1^2 + s_2^2}}; \quad m = n \cdot s_2; \quad l = n \cdot s_1 \tag{7}$$

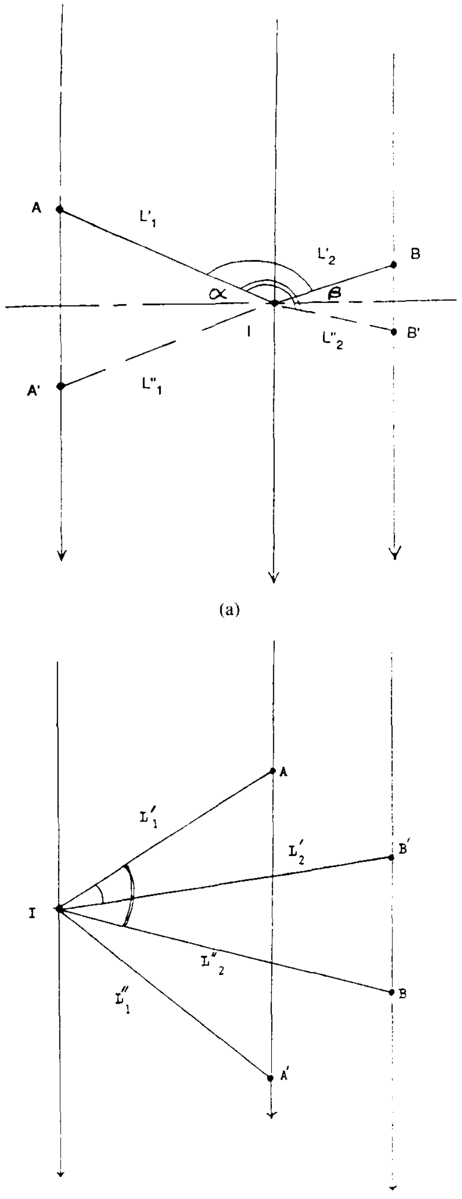
with the constants as defined above in (1)–(6).

Appendix B

In general, there will be several sets of solutions to (7). Since the grid is actually projected onto the plane, the projector axis must make an acute angle with the normal N_p. Since the projector axis is assumed to be parallel to the Z axis, the Z component of the normal n must be positive. In addition, since the grid is actually seen in the image, the camera axis must also make an acute angle with N_p. There are two computed normals in 3-D which satisfy both these constraints.

With the aid of Fig. 5 we now argue that there are (almost always) two planes in 3-D that could create radials of length $|L_1'|$ and $|L_2'|$ in the image and that with further calibration, one of them can be chosen as the sensed surface element. Let I be the intersection of L_1' and L_2 and let A and B be the points of termination of L_1' and L_2'. Establish a coordinate system in the image with I as the ordinate axis. The positive direction of the ordinate axis is toward the direction of the source of the projected ray. (I is the image of the projector ray creating the 3-D grid intersect.) Construct lines through points A and B and parallel to the ordinate axis. Since the lines form a parallel set, there are other points A' and B' such that $d(A', I) = d(A, I)$ and $d(B', I) = d(B, I)$. (In degenerate cases $A = A'$ or $B = B'$ and there will be only one 3-D plane possibly creating them.) Thus (7) will be satisfied using A or A' (B or B') interchanged; and it appears that there will actually be four solutions to N_p. However, because the object is opaque, two solutions are discarded due to visibility criteria as follows. If the I ray images between the A and B rays [Fig. 5(a)] then each candidate solution BIA, BIB', $B'IA$, or $B'IA'$ is feasible only if the angle bIa is less than or equal to Π, where $b = B$ or B', $a = A$ or A'. If the A-ray images between the I and B rays [Fig. 5(b)] then the angles bIa must be positive for a solution to be feasible. Clearly, there are only two feasible solutions in either case shown in Fig. 5, except for the accidental case when there is only one.

At this point it has been shown how to compute two possible normals N_p, and, in the spirit of Wang *et al.* [15], we can rely on outside constraints to make a final choice.



(a)

(b)

Fig. 5. Sketch of imaging situation oriented with respect to the image of the projector ray that creates the visible grid point I.

Acknowledgment

We acknowledge the assistance of J. Giampa and J. Schneider in performing the experiments and computations.

References

[1] M. Asada, H. Ichikawa, and S. Tsuji, "Determining surface orientation by projecting a stripe pattern," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 5, pp. 749–754, Sept. 1988.

[2] D. Ballard and C. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[3] R. Bajcsy and L. Lieberman, "Texture gradient as a depth cue," *Comput. Graphics Image Processing*, vol. 5, no. 1, pp. 52–67, 1976.

[4] L. Caponetti, M. Chiardia, A. Distante, G. Pasquariello, A. Serafini, E. Stella, N. Ancona, and G. Attolico, "A three-dimensional vision system for bin-picking," in *Proc. CVPR '86*, Miami Beach, FL, June 22-26, 1986, pp. 407-411.

[5] L. Dreschler and H. Nagel, "Volumetric model and 3-D trajectory of a moving car derived from monocular TV frame sequences of a street scene," *Comput. Graphics Image Processing*, vol. 20, no. 3, pp. 199-228, 1982.

[6] E. Hall, J. Tio, C. McPherson, and F. Sadjadi, "Measuring curved surfaces for robot vision," *Computer*, vol. 15, pp. 42-54, 1982.

[7] B. K. P. Horn, *Robot Vision*. New York: McGraw-Hill, 1986.

[8] ——, "Obtaining shape from shading information," in *Psychology of Computer Vision*, P. H. Winston, Ed. New York: McGraw-Hill, 1975.

[9] G. Hu and G. Stockman, "Sensing 3-D surface patches using a projected grid," in *Proc. CVPR '86*, Miami Beach, FL, June 22-26, 1986, pp. 602-607; also to appear in *IEEE Trans. Pattern Anal. Machine Intell.*

[10] K. Ikeuchi, "Shape from regular patterns," *Artificial Intell.*, vol. 22, no. 1, pp. 49-75, 1984.

[11] J. Kender, "Shape from texture: A brief overview and a new aggregation transform," in *Proc. DARPA Image Understanding Workshop*, Nov. 14-15, 1978.

[12] H. H. Nagel, "On the derivation of 3-D rigid point configurations from image sequences," *Proc. Pattern Recognition and Image Processing Conf.*, Dallas, TX, 1981, pp. 103-108.

[13] R. Ray, J. Birk, and R. Kelley, "Error analysis of surface normals determined by radiometry," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, no. 6, pp. 631-644, Nov. 1983.

[14] N. Shrikhande and G. Stockman, "Computing surface orientation using a projected grid," Tech. Rep. MSU-ENGR-88-009, Sept. 1988.

[15] Y. F. Wang, A. Mitiche, and J. K. Aggarwal, "Computation of surface orientation and structure of objects using grid coding," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, no. 1, pp. 129-136, Jan. 1987.

[16] P. Will and K. Pennington, "Grid coding: A preprocessing technique for robot and machine vision," *Artificial Intell.*, vol. 2, no. 3/4, pp. 319-329, Winter 1971.

Image Enhancement for Segmentation by Self-Induced Autoregressive Filtering

MARK C. K. YANG AND CHUNG-CHUN YANG

Abstract—**A filter induced by the autoregressive pattern from an image is used to enhance the image itself. The enhanced image usually reveals the wave pattern and nonstationary structures more clearly than the original. Unlike the Fourier analysis, local structures in the image are still retained. The merits of this method have been demonstrated on simulated data and SAR ocean imagery.**

Index Terms—**Image enhancement, random field, texture, 2-D autoregressive process.**

I. Introduction

Image segmentation based on texture is a popular topic in image processing. For the beginners, recent survey papers by Van Gool *et al.* [1] and Rosenfeld [2] (Sections C and F) provide a good starting point for the literature search. The method used in the present study is based on a two-dimensional (2-D) autoregressive model which has been used for texture analysis by Tou [3], Jain, [4], [5], Kashyap and his colleague [6]-[8], Haralick [9], Therrien [10], and Therrien *et al.* [11]. Cross and Jain [12] and many other authors have also used related Markov random field models. This type of approach assumes that the image is a random field of intensities, denoted by $X(h, k)$, as the brightness of the image in the (h, k)th coordinate in the image plane. Small segments of this image plane are assumed stationary and representable by an autoregressive model

$$X(h, k) = \sum_{(i,j)\in P} a_{ij}X(h - i, k - j) + e(h, k) \qquad (1.1)$$

in which P is a set of coordinates excluding $(0, 0)$ a_{ij}'s are parameters usually referred to as the autoregressive coefficients, and $e(h, k)$ is a white noise process. The choice of P is not easy. It may be confined in one quadrant, in a half plane, or extended over the whole plane. Discussions of the choices of P and their theoretical and practical implications can be found in [5], [11], Chellappa and Kashyap [13], and Tjøstheim [14]. It has been shown by many authors through visual comparison of synthesized and original textures that (1.1) is quite adequate for texture representation (see [6], [13], Kashyap *et al.* [15]).

In this correspondence, we will use (1.1) to segment those images that seem difficult to separate from human visual. Examples of this type of image have been synthesized in Julesz [16] by rearranging random dots, in Ashjari [17] by using first-order autoregressive line processes, and in Diaconis and Freedman [18] by using high-order stochastic structures. They conclude that human vision is not equipped to differentiate random fields with certain types of latent structures. However, to separate them can be very important in practice, such as determining the wave patterns on the sea surface. In this correspondence, synthetic aperture radar (SAR) ocean surface images are used as examples. We would like to extract from them the wave patterns indistinguishable by human inspection. Besides some of the images to be presented later in this correspondence, Keyte and Macklin [19] have also demonstrated that similar image appearances may contain different wave spectra. Readers who are interested in seeing more ocean radar images can find them in Allan [20]. To adapt the commonly used random field segmentation techniques such as those mentioned in [8] and [10], we encounter the following difficulties.

1) Generally, the ocean surface is quite variable, and consequently, it is very difficult to find a training sample for the prototype areas.

2) A small portion of the ocean surface usually contains a very large number of pixels. For example, in SIR-B SAR imagery, a typical picture covering an area of 250 mi × 250 mi contains 5000 × 5000 pixels. If each segment is a 64 × 64 matrix, then the whole image contains roughly 610 segments. To group them by cluster analysis can be a tremendous computational burden.

3) Most segmentation procedures have a threshold problem, i.e., when to claim statistical significance. Because multiple comparisons are not mutually independent, a theoretical threshold is almost impossible to find.

All these difficulties are usually handled by human interpretation. For example, prototypes may be determined visually by examining the 2-D clusters formed from a display of any two parameters in model fitting. The best threshold is then chosen to be the value which produces segments that agree with the operator's perspective. Since human interpretation seems unavoidable in image segmentation, we propose a method to first modify the image itself so that a human viewer can segment it more easily. The emphasis is on the wave form in the image, but nonwave structures are also considered.

II. Method and Justification

The first step in our image enhancement (modification) is to use model (1.1) for each segment. The P in (1.1) is chosen to be a subset of the quarter plane following Tjøstheim [14] and the num-

Manuscript received February 4, 1987; revised December 23, 1987. Recommended for acceptance by A. K. Jain.

M. C. K. Yang is with the Department of Statistics, University of Florida, Gainesville, FL 32611.

C.-C. Yang is with the Naval Research Laboratory, Washington, DC.

IEEE Log Number 8826292.

0162-8828/89/0600-0655$01.00 © 1989 IEEE