Recovering three dimensional shape from a single image of curved objects

Jitendra Malik

Computer Science Division, EECS
University of California, Berkeley

Abstract

We propose an algorithm to recover three-dimensional shape i.e. surface orientation and relative depth from a single segmented image of scenes containing opaque curved objects bounded by *piecewise smooth* surfaces. It is assumed that the surfaces have no markings or texture and that the reflectance map E = R(n) is known. Solutions for simpler versions of this problem have been presented by Sugihara for polyhedra, and by Horn et al for smooth surface patches.

We first analyze the constraints from the line drawing and the image brightness values on the faces, edges and vertices in the scene. For a face this is done by Horn's image irradiance equation. To set up the other constraints we need to know line *labels-which* can be found using the algorithm developed in [Malik '85]. At a limb, the direction of the surface normal is known directly. We develop a variational formulation of the constraints at an edge-both from the known direction of the image curve corresponding to the edge and the shading. The associated Euler-Lagrange equation completely captures the local information. At a vertex, the constraints are modelled by a set of non-linear equations. An algorithm has been developed to solve this system of constraints,

1 Introduction

In this paper, we study the problem of recovering the three-dimensional shape of the visible surfaces in a scene from a single two-dimensional image. We restrict our attention to scenes composed of opaque solid objects bounded by piecewise smooth surfaces with no markings or texture on them. By three-dimensional shape we mean a map of surface normal vectors—or equivalently relative depth—along the lines of Marr's 2 1/2D sketch, Horn's needle diagram, or Barrow and Tennenbaum's intrinsic images.

The two sources of information about 3-D shape in a segmented image of the class of scenes permitted by us are (a) the line drawing, and (b) the pixel brightness values. Both have been the subject of considerable research in computational vision. Work on line drawing interpretation of scenes containing curved objects has been most successful in qualitative characterization, e.g., in terms of line labels [8] or sign of gaussian curvature [6]. Attempts [1] have also been made to obtain numerical surface orientation information by making additional assumptions.

Use of pixel brightness values in a single smooth surface patch has been the theme of the shape-from-shading work of Horn and his colleagues [5], [2]. Here the goal has been to solve the image irradiance equation-an equation relating surface orientation to brightness-by supplying the surface normal direction along the boundary of the patch. It is usually assumed that the reflectance map *R(n),* which specifies the radiance of a surface patch as a function of its orientation, is known. For the canonical case of lambertian surfaces illuminated by a point light source, this implies knowing the light source direction. Three-dimensional shape recovery is possible if the patch happens to be bounded by limbs (called occluding contours by some authors) which makes the boundary surface normal calculation easy. No use is made of other kinds of lines e.g. projections of edges(tangent plane discontinuities). Consequently the approach cannot be directly employed on images of general piecewise smooth curved objects.

The question which can then be posed is—are there any schemes which try to exploit all the information in a segmented image, i.e., both line drawing and shading constraints, in order to recover quantitative 3-D shape. This has been done for the special case of polyhedra by Sugihara [12]. For scenes containing curved objects, the problem is significantly harder and Sugihara's approach does not have a natural generalization. To the best of our knowledge there was no method for 3-D shape recovery for the class of curved objects bounded by piecewise smooth surfaces.

In this paper, we develop such a method. The scheme builds upon, and exploits, past work on line drawing interpretation—specifically the line labelling work of Malik [8], [9],—and the work of Horn and his colleagues on shape-from-shading. A more detailed account of this scheme can be found in [10].

2 Constraints from a line drawing

In order to study the constraints imposed by a line drawing on the shape of the scene, it is convenient first to obtain a qualitative characterization of each line. This is done by associating with it a label from the set {+,-,«-,-►,♦-«-,-*->}. The first four of these labels denote *edges* which correspond in the scene to the intersections of surfaces with distinct tangent planes. These labels have the usual significance as in the Huffman-Clowes label set for polyhedra. The {♦-«-,-*-►} labels denote l*imbs* which correspond to curves along which the surface curves smoothly around to occlude itself. As one walks in the direction of the twin arrows the surface lies to the right. See Figure 1 for an example.

For curved objects the label can change along a line as can be seen on edge AB in Figure 2 (section 3.2). We therefore need to distinguish between two different senses of line labelling. A *dense labelling* is a function which maps the set of *all* points on curves in the drawing into the set of labels. The dense labelling problem is to find all the dense labellings of a drawing which can correspond to a projection of some scene. Currently no algorithm is known for finding the dense labelling of a line drawing of curved objects using only the information available in a line drawing. In the context of

Figure 2, it means that there is no algorithm for exactly locating X, the point of transition from convex to occluding.

Alternatively, we could restrict our attention to sufficiently small neighborhoods of the junctions of the line drawing. For each line segment (between junctions) we now have to specify only two labels-one at each end. Of the 6^{2n} combinatorially possible label assignments to the *n* lines in a drawing only a small subset correspond to physically possible scenes. We refer to these as *legal sparse labelltngs.* The determination of all legal sparse labellings of a particular line drawing is the sparse labelling problem. Noto that the 6et of legal sparse labellings is always a finite set (usually small).

The sparse labelling problem has been tackled successfully- by Hurfman[4] and Clowes [3] for trihedral objects, by Mackworth [7] and Sugihara [12] for arbitrary polyhedra, and by Malik [8], [9] for curved objects.

Line labelling is important precisely because lines with different labels correspond to different types of constraints. At limbs one can determine the surface orientation uniquely. It can be shown [l], [5] that n, the unit surface normal, lies in the image plane and is in the direction of $\mathbf{N_c}$ — the outward pointing unit vector in the image plane drawn perpendicular to the projection of the limb. See Figure 1.

At an edge, the constraint is weaker. Let e be the unit tangent vector to the edge at a point, and let n_1 and n_2 be the unit surface normals to the tangent planes to the two faces f_1 and f_2 at the point. Let e be oriented such that when one walks on the edge in the direction of $\hat{\mathbf{e}}$ „ the face f_1 is to the left (see Figure 1). Now e is perpendicular to n_1 because e lies in the tangent plane to the fac‹f_1, milarly e is perpendicular to n_2. Therefore e is parallel to Similarly. We do not know the vector e, but from a line drawing we can determine its orthographic projection into the image plane. We thus have the constraint $(\mathbf{n_1} \times \mathbf{n_2})_{\mathbf{proj}} = \lambda \hat{\mathbf{e}}_{\mathbf{proj}}$. Here the notation vproj is used for the orthographic projection of v into the image plane, λ is a positive scalar if the edge is convex, negative if the edge is concave. Note that this constraint is equally valid for occluding convex edges, where one of the surface normals corresponds to a hidden face.

For later use, it is convenient to develop an alternative version of the orientation constraint. Let N_c be a unit vector in the image plane perpendicular to e_{proj}. As that $(\mathbf{n_1} \times \mathbf{n_2})_{\mathbf{proj}} \cdot \mathbf{N_c} = 0$. As N_c has no component in the z direction, this is equivalent to saying that $(\mathbf{n_1} \times \mathbf{n_2}) \cdot \mathbf{N_c} = 0$, or using the vector triple product notation $[\mathbf{n_1}\, \mathbf{n_2}\, \mathbf{N_c}] = 0$.

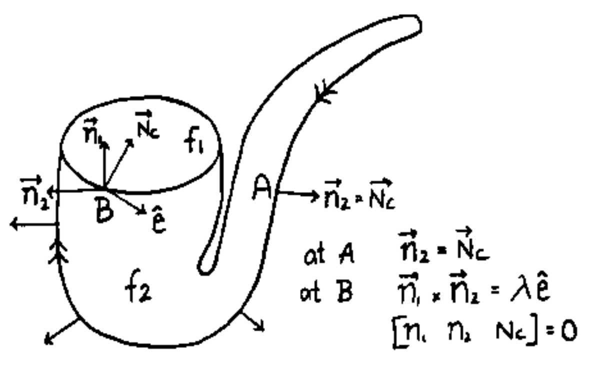

Figure 1: Orientation constraints at limbs and edges

3 Image constraints on 3-D shape

If the dense labelling of a line drawing is known, the problem of three-dimensional shape recovery can be formulated as that of finding a piecewise smooth function n(x,y) which minimizes the following functional

$$\lambda_1 \iint_{I-B} (E - R(\mathbf{n}))^2\, dx\, dy + \lambda_2 \int_E [\mathbf{n}_1\, \mathbf{n}_2\, \mathbf{N}_c]^2\, ds + \lambda_2 \int_L (\mathbf{n} - \mathbf{N}_c)^2\, ds + \lambda_3 \iint_{I-B} (\mathbf{n}_x^2 + \mathbf{n}_y^2)\, dx\, dy$$

Here *I* is the entire image, *E* the edges, *L* the limbs, and *B* the set of all curves (edges and limbs). In the two line integrals *s* is the arc length parameter along the curve. $n_1(5)$ and $n_2(5)$ are the two limits of the discontinuous function n(x,y) when a point on an edge is approached on its two faces.

The first term measures the error in satisfying the image irradiance equation, and the second and third terms measure the errors in trying to satisfy the line drawing constraints. The last term is a 'regularization term' [11], [13] intended to select a particularly smooth solution.

Note that we have not ensured that the position constraints corresponding to the line labelling are satisfied nor that the resulting *n(x,y)* is integrable. How this can be done is discussed in [10].

One could find the Euler equation and develop a numerical scheme for minimizing this functional-however that would be a futile exercise as there is no available algorithm for determining the dense labelling in advance. Of course, we could seek to recover the dense labelling as part of the process of finding the shape *n(x,y).* This leads to a difficult global minimization problem, which is discussed in [10].

In this paper we develop an alternative approach. While we do not know how to precompute the dense labelling, the set of legal *sparse* labellings of the drawing can in fact be precomputed using the line labelling algorithm developed in Malik [8]. To exploit this knowledge, we need to formulate the image constraints in a slightly different way.

It is convenient to partition the constraints into the following three classes: (a) constraints in the interior of an im?ge area, (b) constraints in the neighborhood of an image curve, (c) constraints in the neighborhood of a junction. Surface continuity requirements provide the cross-coupling among the solutions of these constraint sets. In the next three subsections we analyze each of these and develop schemes for local shape recovery.

3.1 Shape constraints inside an area

The only source of constraint is the shading and we wish to find the 'smoothest' surface consistent with the pixel brightness values. This is exactly the shape from shading problem which has been studied by Horn and his colleagues and we can adopt their analyses and algorithms unchanged. Specifically, we use the relaxation scheme suggested by Drooks and Horn [2], section 4.2.

The Horn-Brooks relaxation scheme, just like the other algorithms for this problem, requires knowledge of the direction of the surface normal along the boundary of the patch. Empirical evidence suggests that knowing the surface normal along a significant fraction of the boundary is adequate in practice.

3.2 Shape constraints along an image curve

At a limb, the surface normal is uniquely determined by the direction of the projected curve. So the interesting case is that of an edge, where there are both shading and line drawing constraints. We want to use these to determine the surface normals $\mathbf{n}_1(s)$ and $\mathbf{n}_2(s)$ the two faces f_1 and f_2 as function of the arc length parameter s along the edge. To do this, we have to measure the brightness values $E_1(s)$, $E_2(s)$ on the two sides of the edge in the line drawing.

To determine n_1 and n_2 one has to solve for 4 independent parameters. (While there are 3 components of n, the unit length constraint means that only two are independent.) If the edge is connect (convex or concave), there are two equations due to shading: $E_1 = R(\mathbf{n}_1)$ and E_2- The direction of the edge gives us one more equation $[\mathbf{n}_1\ \mathbf{n}_2\ \mathbf{N}_c] = 0$, still leaving us one equation short. It is not possible to solve for n_1 n_2 by just looking at the neighborhood of a point on an edge, just as in the standard shape-from-shading problem it is not possible to compute n locally. We need to make use of boundary conditions, and maximize smoothness-points which are physically close should have similar surface normals . To do this, we incorporate a ' regularization term' $\int(\mathbf{n}_1'(s))^2 + (\mathbf{n}_2'(s))^2\,ds$ where $\mathbf{n}_1'(s)$, $\mathbf{n}_2'(s)$ are the derivatives of $n_1(s)$ and $n_2(s)$ with respect to arc length. We thus have the following composite functional

$$\int \lambda_1\{(E_1 - R(\mathbf{n}_1))^2 + (E_2 - R(\mathbf{n}_2))^2\} + \lambda_2[\mathbf{n}_1\ \mathbf{n}_2\ \mathbf{N}_c]^2 + \lambda_3\{(\mathbf{n}_1')^2 + (\mathbf{n}_2')^2\} + \mu_1(\mathbf{n}_1^2 - 1) + \mu_2(\mathbf{n}_2^2 - 1)\,ds \qquad (1)$$

$\mu_1(s)$, $\mu_2(s)$ are Lagrange multipliers intended to ensure that the normals $n_1(s)$ and $\mathbf{n}_2(s)$ have unit length.

Minimizing this functional is an exercise in the calculus of variations. We have to find the Euler equations, eliminate the Lagrange multipliers and come up with a suitable finite difference scheme. These calculations [10] result in the following scheme for computing $\mathbf{n}_1^{i+1}$ given knowledge of $\mathbf{n}_1^{i-1}$, $\mathbf{n}_1^i$ and $\mathbf{n}_2^i$:

$$\begin{aligned}
\mathbf{q}_1^i &= \mathbf{n}_1^{i-1} - (\mathbf{n}_1^{i-1}\cdot\mathbf{n}_1^i)\mathbf{n}_1^i \\
\mathbf{p}_1^i &= -\mathbf{q}_1^i - \alpha(\Delta s)^2(E - R(\mathbf{n}_1^i))\{R'_{\mathbf{n}_1} - (R'_{\mathbf{n}_1}\cdot\mathbf{n}_1^i)\mathbf{n}_1^i\} + \\
&\quad \beta(\Delta s)^2[\mathbf{n}_1^i\ \mathbf{n}_2^i\ \mathbf{N}_c^i]\{(\mathbf{n}_2^i \times \mathbf{N}_c^i) - [\mathbf{n}_1^i\ \mathbf{n}_2^i\ \mathbf{N}_c^i]\mathbf{n}_1^i\} \qquad (2)
\end{aligned}$$

To compute $\mathbf{n}_2^{i+1}$, we use

$$\begin{aligned}
\mathbf{q}_2^i &= \mathbf{n}_2^{i-1} - (\mathbf{n}_2^{i-1}\cdot\mathbf{n}_2^i)\mathbf{n}_2^i \\
\mathbf{p}_2^i &= -\mathbf{q}_2^i - \alpha(\Delta s)^2(E - R(\mathbf{n}_2^i))\{R'_{\mathbf{n}_2} - (R'_{\mathbf{n}_2}\cdot\mathbf{n}_2^i)\mathbf{n}_2^i\} - \\
&\quad \beta(\Delta s)^2[\mathbf{n}_1^i\ \mathbf{n}_2^i\ \mathbf{N}_c^i]\{(\mathbf{n}_1^i \times \mathbf{N}_c^i) + [\mathbf{n}_1^i\ \mathbf{n}_2^i\ \mathbf{N}_c^i]\mathbf{n}_2^i\} \qquad (3) \\
\mathbf{n}_2^{i+1} &= \mathbf{p}_2^i + \mathbf{n}_2^i\sqrt{1 - (\mathbf{p}_2^i)^2}
\end{aligned}$$

Here $\alpha = \lambda_1/\lambda_3$ and $\beta = \lambda_2/\lambda_3$. If $\mathbf{n}_1^0$, $\mathbf{n}_1^1$, $\mathbf{n}_2^0$, $\mathbf{n}_2^1$ are known, these schemes tell us how to 'grow' the solution. The iteration can in fact be started just given the values of n_1, n_2 at the initial point. As $\mathbf{n}_1'$ and n_3 are both equal to 0 at the initial point (natural boundary conditions), we assume $\mathbf{n}_1^1 = \mathbf{n}_1^0$ and $\mathbf{n}_2^1 = \mathbf{n}_2^0$.

It should be pointed out that the analysis in this section has been for connect edges—if the edge is occluding, we do not know the brightness values for the occluded face and thus have one less constraint. For curved objects, an edge can change its label from convex to occluding between junctions e.g. at point X in Figure 2. However we know [8] that such points correspond to

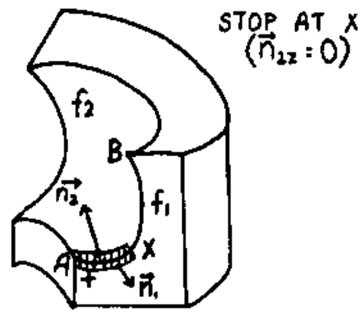

Figure 2: Dealing with edge label transitions

invisible limbs, and therefore can be detected by checking if n, is sufficiently small. In Figure 2, $\mathbf{n}_{2z} = 0$ **at X**. The iterations in (2) and (3) should be stopped at this point.

When n_2 (s)is known along the edge, it is possible to recover $n_1(s)$ even when n_1 unknown at initial point. Use Newton's method to compute n_1 at initial point. This can be done because we now have two constraints on n_1 viz. $E_1 = R(\mathbf{n}_1)$ and $[\mathbf{n}_1\ \mathbf{n}_2\ \mathbf{N}_c] = 0$. As these constraints are non-linear, there is no guarantee of a unique solution. However it can be proved that for a reflectance map arising from a lambertian surface illuminated with a point light source, there can only be one or two solutions. We use the iterative scheme in (2) above to propagate each of the

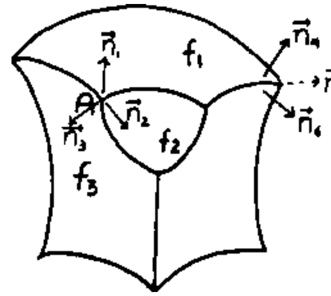

Figure 3: Shape recovery at a vertex

solutions, and pick the one which corresponds to the smaller error as measured by the functional in (1).

3.3 Shape constraints at a vertex

At a vertex we have (a) shading constraints from the brightness values on the faces which meet at the vertex, and (b) line drawing constraints from the projections of the edges incident at a vertex. Under certain conditions, there are enough constraints to uniquely determine the surface normals on each of the faces purely from the local information.

Consider vertex A in Figure 3 and enumerate the constraints on n_1, n_2, and n_3. There are 6 independent parameters (2 for each surface normal) and 6 independent equations (3 from brightness values on the faces, 3 from the directions of the projections of the edges). There are as many independent equations as unknowns. As the equations are nonlinear, we are not guaranteed a unique solution. However, pruning using our additional knowledge about the three edges being convex or concave typically leaves us with only one solution.

It can be seen that the method described above works for any vertex with $n \geq 3$ faces if and only if they are all visible. If any of the faces at the vertex is hidden, the constraint due

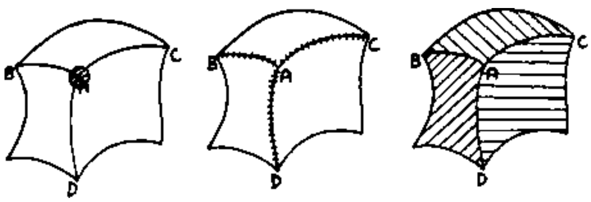

Figure 4: Sequence of steps in global shape recovery

to the brightness value on that face is missing, and there are fewer equations than unknowns. However if there is exactly one hidden face, and the surface normal on one of the faces at the vertex is known (perhaps by propagation of information on one of the faces/edges that it lies on), then the number of independent equations again becomes equal to the number of unknowns and one can solve for the rest of the surface normals. In Figure 3, once n_4 is known, n_5 and n_6 can be determined.

4 Global Shape Recovery

In sections 3.1-3.3 we studied the constraint sets for image areas, curves and junctions, and developed numerical schemes for local shape recovery. In [10], we show how these combine to form a global shape recovery algorithm. Here we will illustrate this algorithm on two examples.

The hatched parts of the figures denote what has been freshly computed in a particular iteration. In figure 4, the computation starts with determination of the surface normals to the three faces at A, using the vertex shape recovery procedure, followed by the computation of the surface normals along AB, AC and AD using the edge shape recovery procedure described by the equations (2) and (3). This generates the boundary conditions needed for

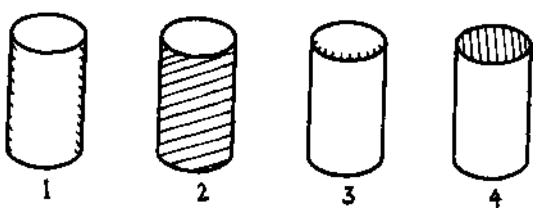

Figure 5: Sequence of steps in global shape recovery

applying the Horn-Brooks shape from shading algorithm to each face. In figure 5, the first two stages are self-evident, and the third stage makes use of the procedure for computing $n_1(s)$ given $\mathbf{n}_2(s)$ along the edge described in the last paragraph of section 3.2. Finally the Horn-Brooks shape from shading algorithm is applied to the top face.

[1]In the sense that it corresponds to the particular scene being imaged. They are all consistent with the line drawing information.

In the examples just considered, we started from the correct labelling. What is actually available to us are a set of sparse labellings among which only one is correct[1]. When we start from an incorrect sparse labelling, the 3-D shape computed will be incorrect and hence will not satisfy the line drawing and shading constraints within the margin of error permitted by the inaccuracies in the data (i.e. noise in pixel brightness values, errors in estimates of edge direction etc.). This gives us a way to prune away the shapes resulting from incorrect labellings. We refer the reader to [10] for the complete algorithm and a study of the effects of noise.

References

[1] H.G. Barrow and J.M. Tenenbaum, "Interpreting line drawings as three-dimensional surfaces,** *Artificial Intelligence* 17, pp. 75-116, Aug. 1981.

[2] M.J. Brooks and B.K.P. Horn, "Shape and Source from Shading,** A.I. Memo 820, A.I. Lab, M.I.T. 1985.

[3] M.B. Clowes, "On seeing things,** *Artificial Intelligence* 2 ,pp. 79-116, 1971.

[4] D.A. Huffman, "Impossible objects as nonsense sentences," *Machine Intelligence* 6 ,pp. 295-323, 1971.

[5] K.Ikeuchi and B.K.P. Horn, "Numerical Shape from Shading and Occluding Boundaries,** *Artificial Intelligence* 17, pp. 141-185, Aug. 1981.

[6] J.J.Koenderink," What does the occluding contour tell us about solid shape,** *Perception 13* ,pp. 321-330, 1984

[7] A.K.Mackworth, "Interpreting pictures of polyhedral scenes,** *Artificial Intelligence 4* ,pp. 121-137, 1973.

[8] J.Malik, "Interpreting Line Drawings of Curved Objects," Tech. Rept. STAN-CS-86-1099, Computer Science Dept., Stanford University, 1985.

[9] J.Malik, "Interpreting Line Drawings of Curved Objects," *International Journal of Computer Vision* 1(1), 1987.

[10] J.Malik, "Recovering three dimensional shape from a single image of curved objects,"Tech. Rept. UCB/CSD 87/340, Computer Science Division, U. C. Berkeley, Jan. 1987.

[11] T.Poggio, V.Torre, and C.Koch, "Computational vision and regularization theory," *Nature,* 317(26), Sept. 1985.

[12] K. Sugihara, "An Algebraic approach to Shape-from-Image Problems,** *Artificial Intelligence* 23 pp. 59-95, May 1984.

[13] A.N.Tikhonov and V.Y.Arsenin, *Solutions to Ill-posed Problems,* Winston, Washington, 1977.