VIEWPOINT

FROM OCCLUDING CONTOUR

by

W. Brent Seales
and
Charles R. Dyer

Computer Sciences Technical Report #990

December 1990

Viewpoint from Occluding Contour

W. Brent Seales
Charles R. Dyer

Computer Sciences Department
University of Wisconsin
Madison, Wisconsin 53706

Abstract

This paper addresses the problem of how to constrain the viewpoints of a 3D model that can produce a given set of projected occluding contour features. Our approach is new in that it relies on a precomputed occluding contour representation for polyhedral models that makes T-junctions, contour terminals, and the change in contour topology explicit. These features are organized into a structure encoding inter-feature relationships and dynamic feature changes over viewpoint. Viewpoint constraints generated from this shape representation are used as an index for the problem of recognizing 3D objects from a single monocular image or a set of images gathered over time. The indexing strategies use occluding contour features such as T-junctions to narrow viewpoint by hypothesizing model-image correspondences based on measured contour features and their relationships. An hypothesized correspondence includes a highly constrained viewpoint estimate since occluding contour features are defined by the small region of viewpoints where they can occur. The implementation results demonstrate that a strategy for indexing and matching using simple T-junctions can efficiently solve for a viewpoint that is close to the exact viewpoint and consistently accounts for the occluding contour features.

Support of the University of Wisconsin Graduate School under Project No. 910288 and the National Science Foundation under Grant Nos. IRI-8802436 and DCR-8521228 is gratefully acknowledged.

1. Introduction

This paper addresses the problem of how to constrain the viewpoints of a 3D model that can produce a given set of occluding contour features. Our approach is new in that it relies on precomputed features of the occluding contour for polyhedral models, features that are organized into a structure making inter-feature relationships and dynamic feature changes explicit. Viewpoint constraints generated from this shape representation are used as an index for the problem of recognizing 3D objects from a single monocular image or a set of images gathered over time. It is assumed that: (1) precise geometric information about the models is known; (2) features such as edges and occluding contours can be detected from the image set; (3) information such as surface normals or texture is not available.

The problem of constraining viewpoint for a particular model is a subproblem of model-based 3D object recognition. In general, model-based approaches select a model **M** and the corresponding model transformation **T** that best matches the image data. For each model, the best transformation **T** selected from all possible model transformations is computed. Recognition is the selection of the "best" model-transformation pair, that is the pair (**M**, **T**) with the highest degree of match. Thus a fundamental problem is *viewpoint determination*, the computation of the model transformation that best matches the image for a given model.

Three well-known approaches to the viewpoint determination problem are iterative viewpoint revision, interpretation tree methods, and transformation space for the accumulation of local evidence. Iterative methods revise an initial estimation of viewpoint to bring a model into correspondence with the image [Lowe87, Ponc89]. Interpretation tree methods use a constrained search through a tree of model-image correspondences to solve for feature correspondence and viewpoint [Grim90]. The transformation space approach, which is based on model-image feature correspondences, accumulates local evidence for a model transformation computed from a measured feature such as the vertex pair [Thom87].

One important weakness in each of these approaches is the difficulty in finding a starting viewpoint, one that is known to be relatively close to the best viewpoint solution for a particular model. Iterative methods must start close enough to a solution in order to guarantee convergence to a global solution. Interpretation trees tend to be large, and the lack of global viewpoint constraints yields a search space that is exponential. Transformation spaces are also large, and without constrained viewpoint regions the entire space of transformations must be quantized and represented in order to find global peaks. In each approach, a constrained starting region greatly reduces the work to follow.

Current researchers have used two different methods to hypothesize a viewpoint starting region for a model: perceptual organization and the aspect graph. The principles of perceptual organization attempt to globally constrain the possible starting points for a model by precomputing views that produce persistent, nonaccidental feature sets [Lowe87a]. The aspect graph gives a natural division of all viewpoints into a set of starting points, one starting point for each viewpoint region corresponding to a node of the graph. A global solution is the best solution across all nodes [Bowy89].

Our method uses the occluding contour to constrain the viewing transformation. It is known that the shape of the occluding contour provides strong information for 3D object recognition [Koen84, Rich88]. Information about depth ordering and viewpoint can be inferred from T-junctions and their relative orientations, and the shape of the occluding contour is directly related to the 3D surface generating it. For example, Fig. 1(a) shows that the T-junctions and occluding contour segments alone of a projected 3D object provide enough information to recognize it.

The relationship between sets of occluding contour features generated by arbitrary non-convex 3D shape strongly constrains the possible viewpoints that can generate those features. For example, the projected 3D object in Fig. 2 is rich with visual cues such as T-junctions, ending contours, concave and convex arcs, and inflections. These features independently and in relationship to each other constrain the possible views that can generate the given projection. This notion is captured in part by the aspect graph. The aspect graph is a graph where nodes represent regions of viewpoint space, and arcs represent adjacencies in viewpoint space. The division of viewpoint space into nodes is done so as to preserve *constant aspect* within each node. Two views are in the same aspect when the features projected from those views are qualitatively the same. Constant aspect is a region of viewpoints where the projected features from every viewpoint in the region are qualitatively the same.

Figure 1. (a) The projection of selected junctions and corners of a 3D model. (b) The complete model. This figure is adapted from [Bied85].

Although the visual information captured in the aspect graph is complete and perceptually salient, the organization of this information prevents its effective use. There are several reasons for this. First, the lack of structure in the aspect graph makes it difficult to match a set of features to an aspect node given only partial information. Second, the definition of aspect used to partition the model can potentially depend on features that are never observable in an image. Third, the changes in specific features and feature sets across adjacent aspects is not explicit.

This paper describes a new approach to the starting viewpoint problem using shape features of the occluding contour. In particular, the inflections and junctions that arise from self-occlusion and non-convexity are used as features to determine a constrained viewpoint region for a model. These contour features are precomputed and, unlike the aspect graph, are organized into a structure that makes inter-feature relationships and dynamic feature changes explicit. We relate precomputed model geometry of the occluding contour that is dependent on viewpoint to features in image data. This relationship globally constrains viewpoint and provides a direct

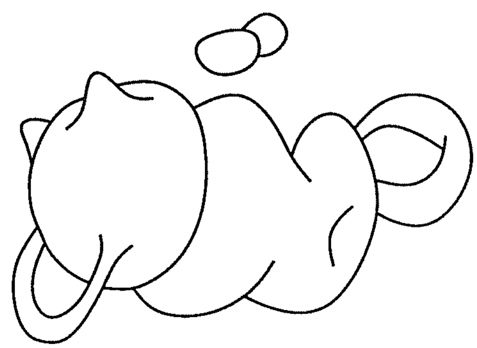


Figure 2. The projection of a 3D non-convex object is rich in visual cues that strongly constrain viewpoint. This figure was taken from [Koen87].

model-to-data correspondence for features of the occluding contour. Included in this framework is a representation of the dynamic evolution of the occluding contour as a function of viewpoint. This dynamic information gives the potential for using image changes over time for viewpoint determination.

We use polyhedral object models, and the behavior of the occluding contours of these models is precomputed and organized for searching and matching. Although the occluding contour of a polyhedral model does not change *continuously* with viewpoint, it changes discretely and provides an analog of the smooth occluding contour in the polyhedral domain. Definitions of the *visible rim* and the *occluding contour* are given in Section 2, along with a description of the *rim appearance representation*, a complete encoding of the structure of the occluding contour of a polyhedral model [Seal90]. Section 3 specifies how features of the occluding contour appear in the image plane, and identifies feature properties that serve to constrain viewpoint. The indexing and search issues involving the occluding contour representation are presented in Section 4. Section 5 presents implementation results from a prototype system that constructs occluding contour information for polyhedra and then matches this information with projected contour features. Results include information about the number and the persistence of precomputed model contour features, as well as a matching algorithm that consistently accounts for the occluding contour features in images. These results and future directions are summarized in Section 6.

2. Representing Rim Features

The occluding contour produced by a 3D opaque shape is generated by the projection process and is directly dependent upon viewpoint. This dependence is the primary property that gives the contour such strong viewpoint-constraining power. A correspondence between a contour feature and a section of a 3D shape that projects to that contour includes a viewpoint constraint. The dependence upon viewpoint is also the cause of major representational difficulties since, in general, changes in the contour occur even for infinitesimal changes in viewpoint. First we clarify the terminology for the rim and the contour, and then present the important aspects of a novel method for computing and representing the features of the polyhedral occluding contour.

2.1. The Rim and the Occluding Contour

The terminology for describing the points on a shape related to the projected contours in an image vary widely. Marr used the terms contour generator, contour, and silhouette [Marr77]. Others have used terms such as limbs [Egge89, Nalw88, Mali87] and the rim [Basr88, Koen87]. To avoid confusion, we define precisely the meaning of the sets of 3D points we call the *rim*, the *visible rim* and the *occluded rim*, and the sets of 2D points in the image plane called the *contour*, the *occluding contour* and the *occluded contour*. Let $p \in S$ be a point on a smooth, oriented, compact surface in $\mathbb{R}^3$. The rim is a set of points on S defined as follows:

Rim:	p is on the rim if the viewpoint vector **V** is tangent to S at p.
Visible rim:	p is on the visible rim if p is on the rim, and p is visible from **V**.
Occluded rim:	p is on the occluded rim if p is on the rim, and p is not visible from **V**.

The occluding contour is a set of points in the image plane generated from S under projection. The contour points are related to the rim as follows:

Contour:	The contour is the projection of the rim.
Occluding contour:	The occluding contour is the projection of the visible rim.
Occluded contour:	The occluded contour is the projection of the occluded rim.

Observe that this definition of the rim is local, and that the contour generated by the projection of the rim contains points that are potentially, but not necessarily, visible. Note also that the entire surface S can be divided into patches of points that are either potentially visible or not visible. The rim is the transition, or the boundary, between patches [Koen90]. The projection of the rim is the contour that would be generated by transparent shapes.

The projection of opaque shapes causes global occlusion, removing some of the rim points from view. For any viewpoint the set of rim points is only potentially visible, and so the rim can be divided into two sets: the visible rim and the occluded rim. Clearly this induces two sets of points in the image under projection. We define the occluding contour to be the projection of the visible rim. The *occluded contour* is the projection of the occluded rim. Fig. 3 illustrates the distinction between these sets of points.

Given these definitions we can construct the analog of the rim for arbitrary polyhedra. We restrict a polyhedron to form a set of closed, oriented 3D volumes. Each edge is the intersection of exactly two faces, and each face is oriented, its orientation given by the ordering of the edges

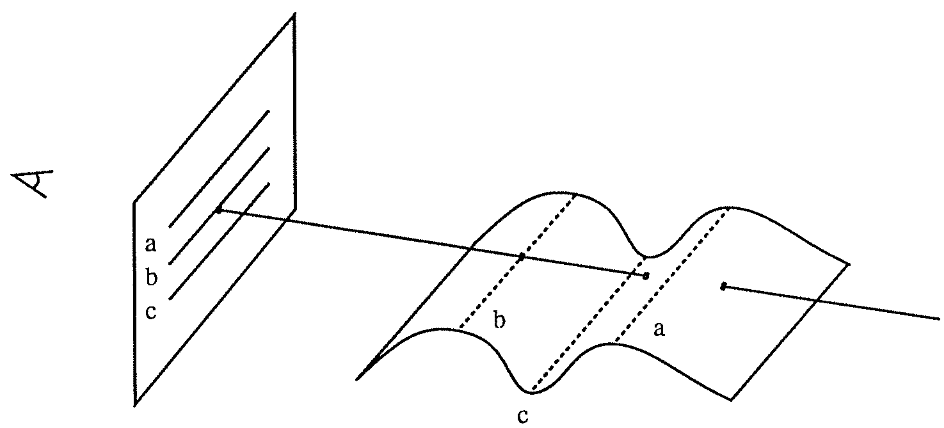


Figure 3. The rim, shown as dotted lines, is the set of points where the viewpoint **V** is tangent to the surface. The rim points labeled a and b are visible, while the rim points labeled c are occluded. The rim sets generate a corresponding contour under projection. The rim is the boundary on the surface between points that are potentially visible and occluded.

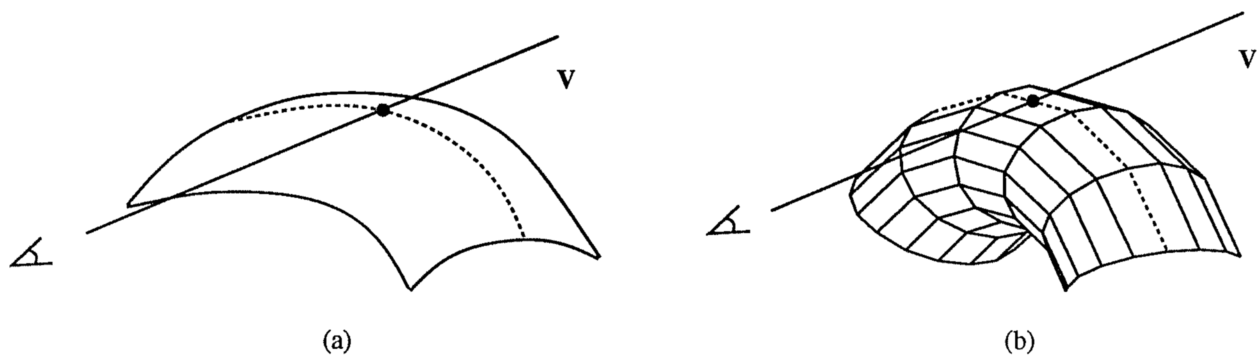


Figure 4. (a) The points where the viewpoint **V** is tangent to the surface are rim points. The rim is the boundary between potentially visible surface points and invisible surface points. (b) The polyhedral analog: rim edges on the boundary between potentially visible and invisible faces.

bounding it. An edge is on the rim when exactly one of the two faces adjacent to it is turned toward the viewing direction. This is analogous to the definition of the smooth rim in that the rim edges are the transition edges between parts of the model that are potentially visible, and parts that are not visible. In Fig. 4(b), the rim edges are exactly those edges bounding the transition between sets of faces that are potentially visible and faces that are not visible.

2.2. The Rim Appearance Representation

The rim appearance representation is the encoding of all the visual events affecting the appearance of the rim [Seal90]. A visual event includes T-junctions and triple-edge junctions, and the degenerate events such as the alignment of an edge and a vertex. The important contribution of this representation is the fact that the exact appearance of the contour is computed in closed form, with its features stored explicitly. By *contour feature* we mean the junctions, splits and merges of the contour, as well as other interesting contour shape properties such as high curvature points. The rim appearance information is encoded as an exact set of volumes in aspect space, a multi-dimensional space formed from the image plane $\times$ viewpoint space. Each edge corresponds to a volume in aspect space, the boundaries of which are algebraic surfaces and curves with a geometric interpretation that results from the apparent intersection in the image plane of pairs and triples of unconnected edges. See [Plan88] for the details of aspect space.

There are three important characteristics of the precomputed rim appearance that make it useful for solving the starting viewpoint problem:

- Visual event extent (persistence) in viewpoint space can be computed exactly
- The interaction between contours (as opposed to only individual edge interaction) can be represented by making interaction adjacencies across viewpoint explicit
- Changes in feature relationships are encoded as a function of viewpoint

The viewpoint extent of a visual event is the bounded region of viewpoint space over which the event is visible. For example, a T-junction between two edges persists over a patch of viewing directions on the view sphere. The boundaries of this patch arise from three separate sets of constraints: the planes defining the viewpoints where the edges are on the rim; the planes defined by the endpoints of the edges; and the triple-edge interactions that cause global occlusion. The persistence of a feature, measured by the size of the space of viewpoints over which it occurs, is a direct indicator of its likelihood of being observed.

Related to individual event extent is the adjacency between event patches. Adjacency information permits the extent of a contour feature to be computed even though the edges forming that feature change. For example, edges that are connected spatially and that form separate but adjacent T-junction patches are associated together as the continuation of the same contour-contour interaction. This continuation guides a search to refine an initial guess at a contour-contour interaction match, since a subsequent set of edges on the contour-contour interaction can be immediately selected for matching using this adjacency information.

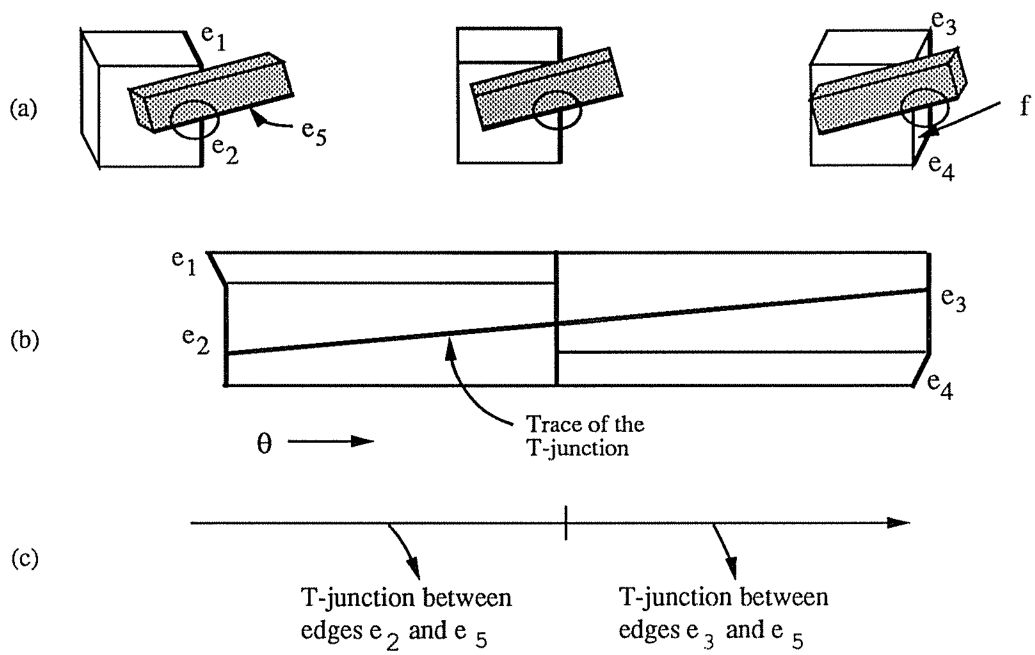


Figure 5. (a) The edges forming the circled T-junction change as viewpoint changes. (b) The viewpoint extent of the T-junction for e_2 and e_5 is adjacent to the extent of the T-junction for e_3 and e_5. (c) The contour-contour interaction is a linked set of adjacent intervals in viewpoint space. Dynamic change in feature values can be computed at the boundary.

Dynamic feature relationships are available by computing the change in features across adjacency boundaries in viewpoint space. This precomputed change gives a prediction of the change in the relationship of the contour features that guides the refinement of a contour-contour match through viewpoint. In addition, this dynamic change information creates a framework that will be used to incorporate dynamic image information gathered as viewpoint changes over time.

Fig. 5 shows three views of a polyhedral scene. We assume that projection is orthographic, and in Fig. 5 we allow only one degree of freedom in viewpoint direction. The T-junction formed by edges e_2 and e_5 in the left view in (a) disappears in the middle view where e_2 leaves the rim. The left interval along the viewpoint axis in (c) shows this extent. The T-junction between e_5 and e_3 begins at this boundary, and the boundary coincides exactly to the viewpoint where e_2 leaves the rim and face f appears. From the adjacency of these two intervals over viewpoint the two distinct T-junctions can be associated together as the same contour-contour interaction, despite the change in the edges that generate the interaction. Finally, the change in the properties of the T-junction such as the angle formed in the image by the projected junction is easily obtained across boundaries where edges either join or leave the rim.

3. Contour Feature Geometry and Organization

The features of the occluding contour that are precomputed for a particular model are defined by viewing direction, self-occlusion from projection, and the curvature of the surface at the rim. Thus the association of a feature in an image to a model contour feature implicitly includes a set of constraints on viewpoint. These constraints are derived from the geometry that produces the contour feature under projection.

We focus here on the following important issues: the geometry of contour features, the organization of these features, and the viewpoint constraints resulting from model-image correspondences. The problem of model-image correspondence is treated as a *contour-level* correspondence rather than an exact edge-edge correspondence. This is based on the organization of contour features across viewpoint, described here and in Section 4.

Smooth opaque shapes without surface discontinuities generate only T-junctions, smooth contours, and contour terminals in the image plane [Mali87]. We assume that the polyhedral model is an approximation of a smooth shape so that there are no true surface discontinuity edges. In general surface discontinuities can be treated without difficulty, but we restrict the

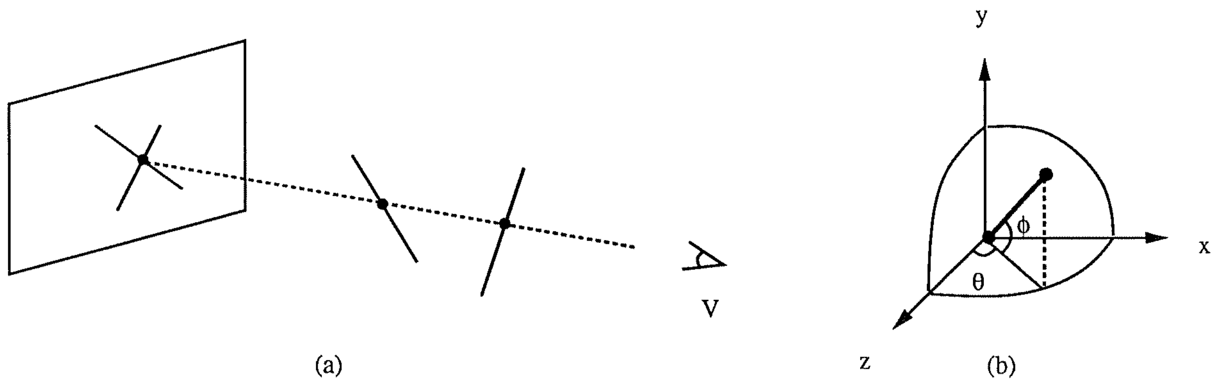


Figure 6. (a) Two edges project to a T-junction for a limited range of viewing directions. (b) The orthographic model represents a viewing direction as a unit vector (θ, ϕ) on the unit sphere.

discussion here to the rim. In this case the only persistent junctions in the smooth occluding contour are T-junctions and contour terminals. Triple-contour (or higher order) intersections can occur, but only from a 1D or 0D set of viewpoints. Any perturbation of viewpoint within an open disc on the view sphere will cause the triple-contour feature to disappear. Accordingly, the features we describe in detail here are the T-junction, caused by the intersection in the image plane of two non-adjacent edges, and the contour terminal, caused by concave edges.

3.1. T-junctions

The geometry of the T-junctions formed by non-convex polyhedra is made simpler by the fact that the edges forming the T-junctions are linear. The geometry of the interaction of two edges forming a T-junction is illustrated in Fig. 6. The image coordinates of a projected T-junction for two edges is represented directly as a function of the viewing direction **V** and the endpoints of the segments [Plan88]. As shown in the figure, a viewpoint is modeled as a point (θ, ϕ) on the unit sphere. The projective transformation is rotation by (θ, ϕ) and then orthographic projection in the direction of the z-axis.

Fig. 6(a) shows only edges, although under the definition of the polyhedral rim, edges forming a T-junction are constrained by the visibility of the faces that meet to form them. Thus a particular edge can only form T-junctions for viewpoints where that edge is on the rim.

A T-junction is observed under projection when the viewpoint is within a patch of viewpoints on the unit sphere. The boundaries of the patch are defined by either sections of great circles or polynomial curves generated by triple-edge events [Plan91]. Fig. 7(a) illustrates the patch of viewpoints on the unit sphere where a T-junction occurs in the image plane. The equations defining the boundaries of the patch are functions of the endpoints of the edges, the normal to the faces meeting at each edge, and the global occlusion that modifies the visibility of the T-junction in the form of the triple-edge event [Seal90].

An entire set of T-junctions can be produced by a non-convex shape at a single viewpoint. The relationships of T-junctions to each other is of interest because of the added viewpoint constraint that they provide. These relationships can be computed using the geometry of the T-

Figure 7. (a) A T-junction will occur between the two edges from any viewpoint within the patch on the view sphere. (b) Two T-junctions co-occur in the image plane when the viewpoint is in the intersection of the patches for each T-junction on the view sphere.

junction patch boundaries. Two or more T-junctions that can arise from the same viewpoint region define a highly constrained region of viewpoints, i.e., the region defined by the intersection of the patches for each individual T-junction. Two T-junctions occur from the same viewpoint, or *co-occur*, when their respective patches overlap on the view sphere. The region common to both patches is the set of viewpoints where: (1) the edges are on the rim; and (2) the two pairs of edges will each project to a T-junction in the image plane. Fig. 7(b) shows two patchs of viewpoints on the view sphere formed by two pairs of edges, and the intersection of those two patches where the T-junctions co-occur.

The model-image correspondence for a T-junction is essentially an hypothesis that assumes that two model edges create the image T-junction. The model edges can be aligned to the image junction for any of the viewing directions within the viewpoint patch created by the geometry of the model edges. After choosing a viewpoint, an appropriate rotation about the optical axis and a translation in the image plane brings the model edges into correspondence with the image junction. Because a T-junction is *oriented*, with the stem of the T-junction being on the unoccluded side, the rotation about the optical axis for alignment is well-defined. Fig. 8 shows two model edges that are to be aligned with an image T-junction. The edges are rotated by the viewpoint (θ, ϕ) and then projected to the image plane. The rotation α about the optical axis orients the edges appropriately. The translation τ in the image plane completes the alignment.

There are two important points to be made here. First, a model-image correspondence implicitly constrains viewpoint since a T-junction defines a small patch of viewpoints. The exact viewpoint within a T-junction patch that best matches the stem and occluding edge of the image junction is difficult to compute, however. In fact, there may be multiple viewpoints within a patch that can produce an angle that exactly matches the image angle. Despite this, the

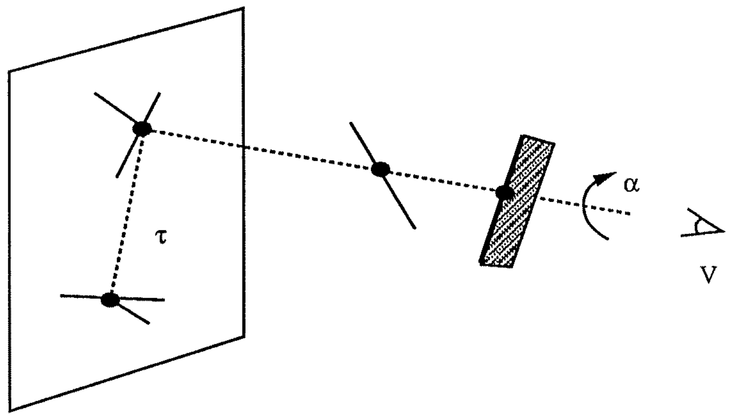


Figure 8. The T-junction formed by a pair of model edges is aligned with an image T-junction by a rotation α about the optical axis and an image plane translation τ to align the junction point.

variation in the angle of the T-junction formed by two edges depends on the size of the viewpoint patch where it occurs [Burn90]. The set of angles formed by a T-junction persisting over a small region of viewpoint space can be approximated by the value of the angle at a single viewpoint within the region.

3.2. Contour Terminals

The geometry of contour terminals is a key part of the change in the topology of the contour across viewpoint. Contours end in polyhedra at a *concave* edge, an edge where the measure of the exterior angle between the faces meeting at the edge is less than π. Concave edges can be excluded from the rim since a concave edge can never occlude anything behind it. When concave edges are removed from the rim set, the rim can form a set of unconnected edges. For example, Fig. 9 shows a simple polyhedron from a viewpoint where the set of rim edges are not connected in $\mathbb{R}^3$. The discontinuity occurs at the dashed concave edges, shown in Fig. 9(c) The broken contour that results under projection, shown in Fig. 9(b), consists of two sections. Each

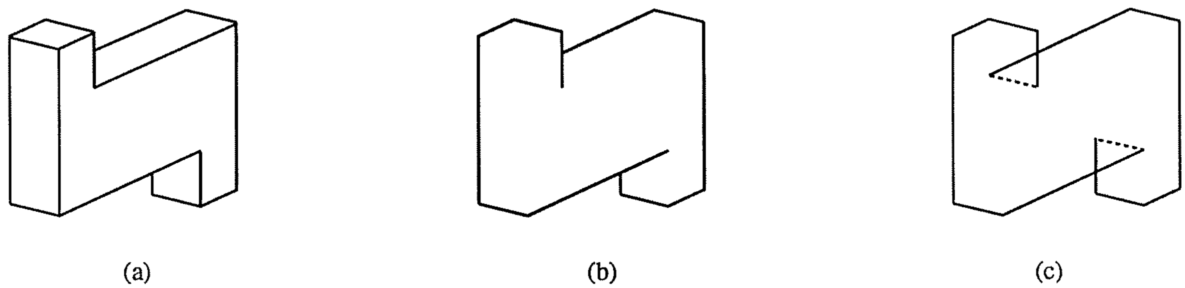


Figure 9. (a) A polyhedral model. (b) The occluding contour is broken by a combination of T-junctions and contour terminals. (c) The dotted edges are concave, and cause the contour terminal.

section corresponds to part of the disjoint rim on the model. A concave edge causes the contour to end at either a T-junction from self-occlusion or at a contour terminal.

The viewpoints where a contour terminal disappears (or a T-junction disappears) form visibility boundaries on the view sphere. These boundaries divide the qualitative appearance of the contour so that the topology of the projected contour is different for each region. This induced viewpoint space division is similar to the idea of constant aspect from the aspect graph. Fig. 10 shows the qualitative difference in the projected contour for a polyhedral model. The curved approximation illustrates the qualitative difference in the contour resulting from visible contour terminals and T-junctions.

The set of viewpoints where the contour breaks at terminals and T-junctions is important because those viewpoints bound regions where the contour topology changes. Over regions of constant contour topology, information such as high curvature points can be computed, and the relationship of those points to the projected T-junctions and contour terminals can be computed. This qualitative description of the contour can provide a coarse description of the contour in

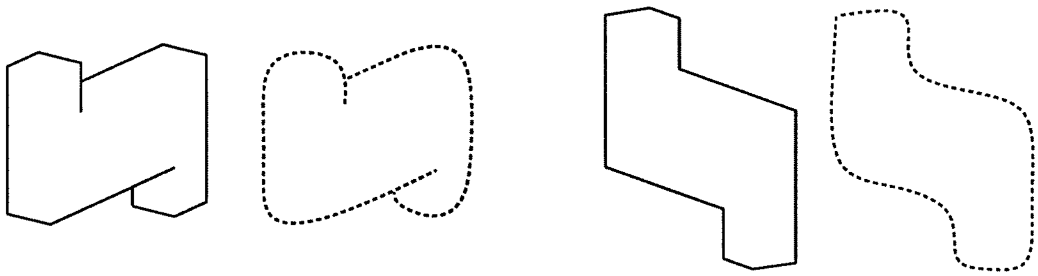


Figure 10. These two views show the qualitative difference between the contours as a result of contour terminals and T-junctions. Viewing regions are computed exactly where the contour changes qualitatively as a result of visible T-junctions and contour terminals.

terms of qualitative connectivity and curvature information. At the same time, exact contour appearance is still maintained in the polyhedral edges and their interaction.

4. Indexing and Viewpoint Refinement Using Contour Features

A viewpoint that is close to the starting point is very valuable for other techniques that can use the starting point to solve exactly for viewpoint [Lowe87]. The organization of the contour information for a polyhedral model is the basis for our method to find a starting viewpoint. Given a set of contour features that have been extracted from an image, we want to determine the viewing transformation of a model that will be close to the best alignment of the object with the data. One benefit in encoding contour information is the ability to index to starting points based on shape features alone, without solving for specific point-point or edge-edge correspondences. Further, the persistence of events and the explicit relationships between features help refine an initial guess and constrain viewpoint.

4.1. Detecting Occlusion Features

The reliable detection of contour features is a difficult problem and is an area of active research [Vail89, Thom85]. It is important to study the way in which shape constrains viewpoint in the ideal case so that the methods discovered can be adopted and adapted in practice. It is also important to maintain a framework that can incorporate methods to provide robustness. Accordingly, we have studied this representation using idealized data, while maintaining a framework to incorporate constraints from motion information that will give better contour detection and localization.

Specifically, the encoding of the dynamic *change* in features as a function of viewpoint directly represents shape changes observed in low-level motion-based methods. These methods can more robustly compute contour information and provide shape change information over time. The added robustness of contour feature detection from motion and the dynamic constraints from motion sequences will be incorporated into this framework.

4.2. Indexing

Explicit contour features provide several powerful ways to index to a starting viewpoint. Precomputed contour features can be organized for indexing based on any measurable characteristic that distinguishes between them. For example, T-junctions create a specific orientation and angle. Multiple features together give relative orientation constraints and also provide scale information. The contour topology itself can give a coarse estimate of the potential matching sets of viewpoints. The implementation results we present in Section 5 make use of T-junction orientation and angle for indexing. In addition, we describe a method using feature pairs to index and solve for scale.

As stated previously, the angle formed by a T-junction between two edges is relatively stable over a small range of views. Since contour-contour interactions are represented as piecewise edge-edge interactions, the angle for a contour-contour interaction can also be represented piecewise, where each edge pair has a fixed, precomputed angle. Then, a starting point for a match between a contour-contour event and an image T-junction is the part of the event that matches the angle of the T-junction within some range of values. To complement this quick indexing method, a fast verification of an hypothesized match can be accomplished by storing a small number of contour segments with the model contour-contour event. A model-image correspondence with matching angles and a small number of verified contour sections is fast to compute. The probability of a large number of coincidental matches is low. Our initial results, described in Section 5, show good performance for idealized data sets.

The well-defined geometry of co-occurring T-junctions provides a strong constraint on viewpoint. For images where more than one T-junction resulting from self-occlusion can be reliably detected, the relative orientation of a pair of T-junctions can be used as an index. A pair of features gives more accuracy for indexing. In addition, scale can be recovered from the measured distance between the two T-junctions. The geometric relationships between multiple features are stable for small changes in viewpoint. Thus, the precomputed values of these relationships can be efficiently organized as an index to a starting position in the model.

4.3. Match Refinement

Once an initial correspondence is made between a model contour feature and an image feature, adjacency information across viewpoint guides match refinement so that the initial indexed match can follow the contour feature through viewpoint space. Specifically, the initial index step hypothesizes a candidate correspondence between an edge-edge interaction in the

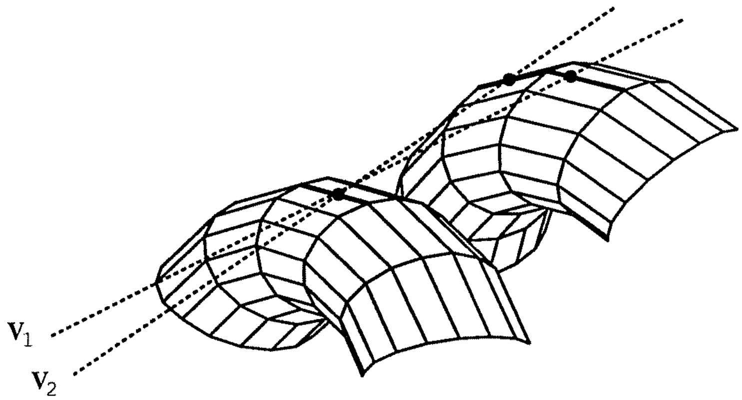


Figure 11. The two different edge-edge events that appear at V_1 and V_2 are associated together as part of a single contour-contour event. This is computed using edge connectivity and visual event adjacency information.

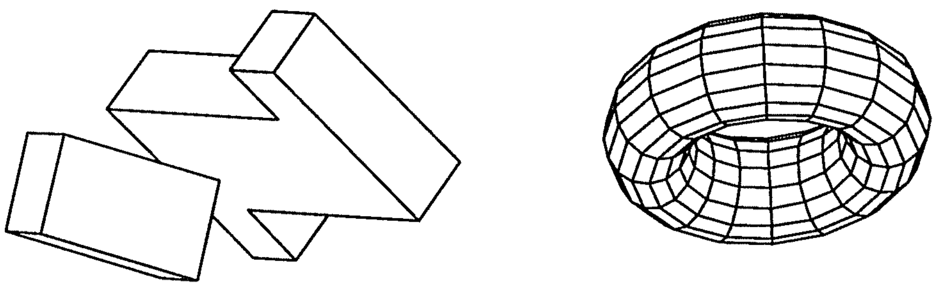
model and an image feature. This hypothesis is really an hypothesis of a contour-contour interaction, and the edge-edge interactions at nearby viewpoints are highly likely match candidates. The nearby edge-edge interactions are precomputed in the rim appearance representation. Thus the hypothesized correspondence between a contour-contour event and a T-junction is refined by following the precomputed event in the model over viewpoint. If the final refined match is below some threshold, it is abandoned for another contour-contour event generated from the initial indexing step.

Fig. 11 gives an example of a contour-contour event that is followed in the model from one viewpoint to the next. An index step that hypothesizes the edge-edge interaction at V_1 is able to move to the edge-edge interaction at V_2 to attempt to find a better match.

The result of this process is the computation of a viewpoint that is close to the best possible viewpoint for a particular model. A complete model-image correspondence is not found, and the best viewpoint is not found. Rather, the result is a viewpoint that accounts for the observed occluding contour in a consistent way. Consistent is determined by the constraints in the way the 3D model can project to the image plane.

5. Implementation Results

A prototype system is being developed to study the issues involved with a model-based, explicit representation of occluding contour features. We have implemented algorithms that compute the exact appearance of the occluding contour of polyhedra under orthographic projection. The results we report here are twofold. First, the task of computing and storing the contour information for medium-sized polyhedral models is shown to be very manageable. Second, the computation of starting viewpoints that are close to the true viewpoint for polyhedra given a set of synthetic projected shape information is successfully demonstrated.



Model	Faces	Edges	Vertices	Visible EE-events	Average Event Persistence
S-shape	16	36	24	142	15%
Torus	256	512	256	1210	2.5%

Figure 12. Size information for the two models above is shown in the table. Event persistence is the percentage of the entire area of the view sphere over which the event occurs.

Fig. 12 shows two example polyhedral models. The table in Fig. 12 indicates model size, the number of edge-edge interactions computed for each model, and the average persistence of each edge-edge interaction as a percentage of the total area of the view sphere. The number of edge interactions produced by a model is a function of several variables including the size of the model, the amount of non-convexity, and the sharpness of the angles between adjacent faces of the model.

There are two important observations from the table in Fig. 12. First, the EE-event information is small enough to be efficiently computed and stored for a model with a moderate number of edges. The number of EE-events is $O(\mathbf{n}^2)$ where $\mathbf{n}$ is the number of edges. Rim constraints and non-degenerate polyhedral models give an average complexity that is much smaller than this worst-case size. Second, the average persistence of EE-events decreases with an increased number of polyhedral faces. These observations illustrate the size/accuracy tradeoff that exists between large models that provide a very close approximation to a smooth surface and small models that provide a coarser approximation.

Figs. 13 and 14 show the computational results of determining a starting viewpoint given a synthetic image generated from a randomly-selected viewing direction. The image contours were generated using orthographic projection with fixed (known) scale. Each image was represented as a set of individual edges, with known T-junctions. An approximate solution for a model-image match is a viewing direction (θ, ϕ), a rotation α about the optical axis, and a translation τ in the image plane.

The circled T-junction in Figs. 13 and 14 was used as the feature to index into the precomputed representation of the occluding contour. The measured angle of the T-junction was used as the primary index into the set of T-junctions, with an initial match being any angle within 10

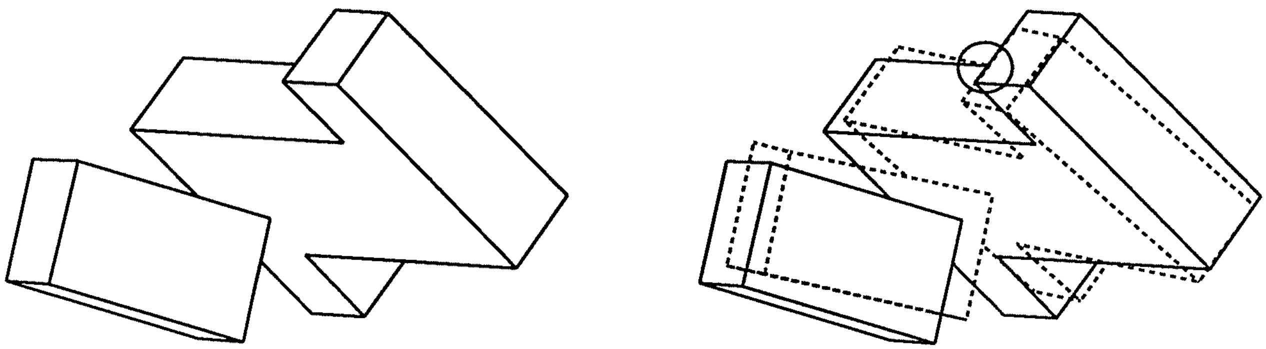


Figure 13. Starting viewpoint determination for a polyhedral model consisting of two separate shapes. The circled T-junction was used as the feature for indexing into the exact representation of the occluding contour for the model.

degrees of the measured value. The relative position of the occluding contour to the oriented T-junction was used to further constrain potential correspondences. A measure of the degree of correspondence was determined by matching other predicted T-junctions and rim features as they appear from the viewpoint that produced the matching T-junction.

An exact solution for viewpoint is not found. An approximate solution is found that can be used as the starting point for other methods that solve exactly for viewpoint [Lowe87]. Furthermore, a contour correspondence is found, not an exact edge-to-edge correspondence. The two model edges aligned with the T-junction in Fig. 14 are not the same model edges that originally projected to that T-junction. Because of the symmetry of the torus there are many close solutions that can be found. A viewpoint is found that is within a fixed distance of the best viewpoint and accurately accounts for the available contour information. For our set of test models, all solutions were within $\frac{\pi}{4}$ radians of the known solution.

Ten correspondences were made by the algorithm for the torus before the solution shown in Fig. 14 was found. The first matches were rejected after measuring the degree of match of the predicted occluding contour edges to the image. A rejected match was revised by following the contour-contour constraints while still satisfying the measured angle constraint. Only 3 matches were required for the view shown of the polyhedron in Fig. 13.

An interesting problem occurs with "coarse" polyhedral models such as the one shown in Fig. 13. A coarse model is one that has large angles between adjacent faces so that it does not closely approximate a smooth surface. Sharp angles cause individual T-junctions to persist over larger ranges of viewpoint space and hence a projected angle value varies over a wide range. This variation causes the initial indexing step based on angle only to perform badly. Additional contour information incorporated into the index step can solve this problem and will help the indexing method tolerate noise.

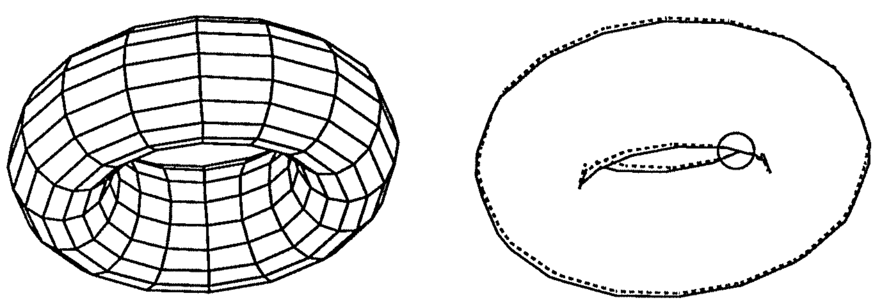


Figure 14. Successful starting viewpoint determination for a model of a torus using oriented T-junction and relative contour features. The projected model is superimposed on the image in the figure on the right. The orientation of the T-junction and the relative position of the occluding contour was used as an index for matching.

There are several interesting observations that can be made about the indexing and matching process. First, convex shapes provide no occlusion-based information, although the interaction between several convex polyhedra gives occlusion cues that constrain viewpoint. Second, symmetry causes a larger number of matches to occur since there is a locus of viewpoints that create identical contours. Notice, however, that *any* such match is a valid starting point. Finally, the initial analysis of this approach has not incorporated other shape-based information that is almost always present in real 3D modeling situations. This omission was intentional in order to study the representation of the occluding contour alone. Clearly, the addition of other shape information, as well as texture, color, and surface markings, will provide a larger set of constraints for viewpoint determination.

6. Conclusions and Future Work

Features of the occluding contour contain strong constraints for viewpoint determination. These constraints can be used to find a starting viewpoint for a model that is close to the true viewpoint that matches a set of projected contour features without solving for exact point-point or edge-edge correspondence. Our novel approach relies on the precomputed relationship of contour features produced by polyhedra under orthographic projection. These precomputed features include contour T-junctions, contour terminals at concave edges and the relative arrangement of sections of the contour. The basis for the computation of this information is the construction, at all viewpoints, of the rim for polyhedra, defined as the analog of the rim for smooth shape. Precomputed contour information is organized over viewpoint based on edge connectivity in the model and feature adjacencies across viewpoint.

Our indexing strategies use occluding contour features to narrow viewpoint by hypothesizing model-image correspondences based on measured contour features and their relationships.

An hypothesized correspondence includes a constrained viewpoint estimate since occluding contour features are defined by small regions of viewpoints where they can occur.

The implementation results demonstrate that a strategy for indexing and matching using simple T-junctions can efficiently solve for a viewpoint that is close to the true viewpoint, and consistently accounts for the occluding contour features.

There are two problems that we will address in future work. First, development of an indexing structure that includes precomputed contour information such as extremal points and inter-feature relationships will improve the reliability and efficiency of the indexing procedure. Second, the problem of detecting occluding contour features and the sensitivity of this approach to noise and incomplete data will be studied by using dynamic information over time. The contour representation framework can incorporate constraints from the *changes* in contour with respect to viewpoint. We are currently studying robust contour detection methods in image sequences and the model-based application of the derived contour dynamics.

References

[Basr88] Basri, R. and S. Ullman, The alignment of objects with smooth surfaces, *Proc. 2nd Int. Conf. Computer Vision*, 1988, 482-488.

[Bied85] Biederman, I., Human image understanding: Recent research and a theory, *Computer Vision, Graphics, and Image Processing* **32**, 1985, 29-73.

[Bowy89] Bowyer, K., D. Eggert, J. Stewman, and L. Stark, Developing the aspect graph representation for use in image understanding, *Proc. Image Understanding Workshop*, 1989, 831-849.

[Burn90] Burns, J. B., R. Weiss, and E. M. Riseman, View variation of point set and line segment features, *Proc. Image Understanding Workshop*, 1990.

[Egge89] Eggert, D. and K. Bowyer, Computing the orthographic projection aspect graph of solids of revolution, *Proc. IEEE Workshop on Interpretation of 3D Scenes*, 1989, 102-108.

[Grim90] Grimson, W. E. L., The combinatorics of object recognition in cluttered environments using constrained search, *Artificial Intelligence* **44**, 1990, 121-165.

[Koen84] Koenderink, J. J., What does the occluding contour tell us about solid shape?, *Perception* **13**, 1984, 321-330.

[Koen87] Koenderink, J. J., An internal representation for solid shape based on the topological properties of the apparent contour, in *Image Understanding 1985-86*, W. Richards and S. Ullman, ed., Ablex, Norwood, N.J., 1987.

[Koen90] Koenderink, J. J., *Solid Shape,* The MIT Press, Cambridge, Massachusetts, 1990.

[Lowe87] Lowe, D. G., Three-dimensional object recognition from single two-dimensional images, *Artificial Intell.* **31**, 1987, 355-395.

[Lowe87a] Lowe, D. G., The viewpoint consistency constraint, *Int. J. Computer Vision* **1**, 1987, 57-72.

[Mali87] Malik, J., Interpreting line drawings of curved objects, *Int. J. Computer Vision* **1**, 1987, 73-103.

[Marr77] Marr, D., Analysis of occluding contour, *Proc. Royal Society London* **197**, 1977, 441-475.

[Nalw88] Nalwa, V. S., Line-drawing interpretation: A mathematical framework, *Int. J. Computer Vision* **2**, 1988, 103-124.

[Plan88] Plantinga, W. H., The Asp: A Continuous, Viewer-centered Object Representation for Computer Vision, Ph.D. Dissertation, Computer Science Department, University of Wisconsin-Madison, 1988.

[Plan91] Plantinga, W. H. and C. R. Dyer, Visibility, occlusion, and the aspect graph, *Int. J. Computer Vision*, to appear, 1991.

[Ponc89] Ponce, J. and D. J. Kriegman, On recognizing and positioning curved 3D objects from image contours, *Proc. Image Understanding Workshop*, 1989, 461-470.

[Rich88] Richards, W., B. Dawson, and D. Whittington, Encoding contour shape by curvature extrema, in *Natural Computation*, W. Richards, ed., MIT Press, Cambridge, Mass., 1988.

[Seal90] Seales, W. B. and C. R. Dyer, Modeling the rim appearance, *Proc. 3rd Int. Conf. Computer Vision*, 1990, 698-701.

[Thom87] Thompson, D. W. and J. L. Mundy, Three-dimensional model matching from an unconstrained viewpoint, *Proc. IEEE Int. Conf. Robotics and Automation*, 1987, 208-220.

[Thom85] Thompson, W. B., K. M. Mutch, and V. A. Berzins, Dynamic occlusion analysis in optical flow fields, *IEEE Trans. Pattern Analysis and Machine Intell.* 7, 1985, 374-383.

[Vail89] Vaillant, R. and O. Faugeras, Using occluding contours for recovering shape properties of objects, *Proc. IEEE Workshop on Interpretation of 3D Scenes*, 1989, 26-32.