Viewpoint From Occluding Contour

W. Brent Seales*
Computer Science Dept
University of Kentucky
Lexington, Kentucky 40536
seales@ms.uky.edu

Charles R. Dyer
Computer Science Dept
University of Wisconsin
Madison, Wisconsin 53706
dyer@cs.wisc.edu

Abstract

In this paper we present the geometry and the algorithms for organizing a viewer-centered representation of the occluding contour of polyhedra. The contour is computed from a polyhedral boundary model as it would appear under orthographic projection into the image plane from every viewpoint on the view sphere. Using this representation, we show how to derive constraints on regions in viewpoint space from the relationship between detected image features and our precomputed contour model. Such constraints are based on both qualitative (viewpoint extent) and quantitative (angle measurements and relative geometry) information that has been precomputed about how the contour appears in the image plane as a set of projected curves and T-junctions from self-occlusion. The results we show from an experimental system demonstrate that features of the occluding contour can be computed in a model-based framework, and and their geometry constrains the viewpoints from which a model will project to a set of occluding contour features in an image.

*Address until September 1992: INRIA Sophia-Antipolis, 2004 Route des Lucioles, B.P. 109 06561 Valbonne Cedex, France Internet: seales@sophia.inria.fr

1 Introduction

The occluding contour is a prominent part of what is seen in the image of a projected three-dimensional shape. For example, human observers can obtain information about 3D surface shape and orientation from the contour alone [Koe90]. This paper reports our work toward recovering a viewpoint estimate of a model from the occluding contour in a model-based vision system. First, we examine an approach to modeling the features of the occluding contour of polyhedra. The definitions and the geometry of the *rim appearance representation* are summarized here [Sea91,SD91]. This representation provides an explicit model of the geometry of the contour for all viewpoints. Second, we describe how the structure and the features in this representation are organized for use in a model-based system that recovers viewpoint estimates given a set of occluding contours in a single image. The precomputed model of how the contour can appear provides a set of constraints in viewpoint since the contour itself is directly dependent upon viewing direction. Although features of the occluding contour are strongly tied to viewpoint, they have been largely ignored because of difficulty in representation.

The occluding contour is the projection in the image of the points on a smooth surface where the visual direction is tangent. Much of the geometry associated with the occluding contour and its pre-image in $\mathbb{R}^3$ (the *rim*) has been studied for the purposes of surface recovery and segmentation [CB90,VF91]. In contrast, our approach is strongly model-based. Rather than recovering 3D surface parameters from image contours, we recover information about a model's viewpoint using the viewer-centered *rim appearance representation.* This representation is organized into a structure that makes features like T-junctions more explicit. By matching these precomputed occlusion features, which are strongly related to viewpoint, to detected image features, a set of viewpoints is recovered. This recovered viewpoint region comes from the constraints that have been precomputed about how the occluding contour can appear in the image plane. We assume that geometric information about the models is known, occluding contours can be detected in the image [VF91], and information such as surface normals or texture is not available.

The problem of constraining viewpoint for a particular model is often considered a subproblem of model-based 3D object recognition [Low87]. In general, the model-based approach is to select a model $\mathbf{M}$ and the corresponding viewpoint $\mathbf{V}$ that will produce a projection that best matches the image data. For each model, the best viewpoint $\mathbf{V}$ is selected from a space of all possible viewpoints. Recognition is the selection of the best model, viewpoint pair, i.e., the pair $(\mathbf{M}, \mathbf{V})$ with the highest degree of match. Thus a fundamental problem under this paradigm is *viewpoint determination*, i.e., the computation of the model viewpoint that best matches the image for a given model. This is essentially equivalent to *pose determination*, where an object's pose describes the transformation that relates the object position to the camera position in world coordinates.

Our approach can be distinguished from previous methods in two respects: (1) the kinds of features that are explicitly represented, and (2) the type of model-image correspondences that are made. First, the appearance of the occluding contour, including the formation and persistence of T-junctions, is represented. There is strong information about viewpoint available in the occluding contour, but the difficulty in adequately and explicitly describing it has prevented its use in the past. Second, the shape and topology of the occluding contour is usually stable over a range of viewpoints but is not generated by a fixed, intrinsic feature of the shape.

For example, a T-junction produced by the projection of a smooth shape persists over an open set in the space of viewpoints, yet the 3D points that project to the T-junction are not fixed over that set. Consequently, a correspondence between a T-junction detected in an image and a model T-junction is different from a point-point correspondence; a T-junction correspondence defines a connected set of viewpoints where the T-junction can occur rather than a single transformation to bring the model features into exact correspondence with the image. This notion of correspondence is more qualitative, producing a small, constrained region of viewpoints rather than a single viewpoint that generates the image features. This implies a two-step procedure for viewpoint determination: (1) finding a constrained region of viewpoints, and, if necessary, (2) find a single viewpoint within the region as a solution.

There are certain advantages in making use of features of the occluding contour for viewpoint determination. The occluding contour provides strong information for viewpoint constraints and for 3D object recognition [Koe90,RDW88], although until now there has been little work that incorporates this information into a model-based approach. The depth ordering of surfaces relative to the viewer and qualitative pose information can be inferred from T-junctions and their relative orientations. The curvature of the occluding contour is directly related to the 3D surface generating it [Koe90,KD91]. 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, a pair of T-junctions independently and in relationship to each other constrain the possible views that can generate them.

We use the T-junctions that arise from self-occlusion and non-convexity as features in order to determine a region of viewpoints that matches the image. These contour features are precomputed and, unlike the aspect graph [KvD79,PD86,GM90], are organized into a structure that makes inter-feature relationships and dynamic feature changes explicit. When the precomputed geometry of the occluding contour, geometry that is dependent on viewpoint, is matched to image features, this relationship globally constrains viewpoint because of the direct dependence of the features of the contour on viewing direction.

The next section presents a summary of the rim appearance representation. We outline the geometry and the algorithm for constructing this representation. Section 3 gives the details of the algorithm we have developed for organizing and searching the rim appearance representation in order to recover a set of viewpoints that bound where a model can project onto a set of occluding contour image features. The implementation results show that the size of this viewer-centered representation is modest, and that tightly-constrained viewpoint regions can be recovered from detected image features such as T-junctions.

2 The Occluding Contour

One of the primary components of 3D model-based computer vision is the representation of the salient, observable features of objects. The occluding contour, which is generated by the projection of points on a surface where the viewing direction is tangent, is one of the primary features of the appearance of an object. The projection mapping generates occluding contours, and the opacity of solid shape causes the creation of T-junctions in the image plane. The arrangement of these contours and T-junctions is directly related to the 3D properties of the shape and provides strong information for recognition [Koe84].

Object-centered object models do not explicitly represent the properties of the occluding contour since the occluding contour is not generated by any specific set of object features. Viewer-centered models of 3D shape are better suited to encode the viewpoint-dependent properties of the occluding contour.

The set of points in $\mathbb{R}^3$ on a smooth shape that projects to the occluding contour changes smoothly with viewpoint, and hence the changing contour in the image plane is a complicated function of shape, viewpoint and projection. Features such as T-junctions and curvature extrema on the contour are interesting in that they persist over large portions of the space of viewpoints despite the continuous change in the 3D surface points generating them. Our approach avoids difficult numerical problems by relying on the linearity of polyhedra [FH86]. We define here the sets of 3D points called the *rim*, the *visible rim* and the *occluded rim*. These definitions for smooth surfaces motivate our approximation of them for the polyhedral model.

Let $p \in \mathbf{S}$ be a point on a smooth, oriented surface in $\mathbb{R}^3$. Let $\mathbf{V}$ be a viewpoint on the unit sphere. A point p is defined to be *visible* when the normal direction at p is turned toward the viewer, i.e., when $\mathbf{V} \cdot \mathbf{n}_p > 0$. This is a local definition of visibility for p that does not take into account global occlusion that may obscure p.

The rim is defined by the local visibility condition, $\mathbf{V} \cdot \mathbf{n}_p = 0$, and the contour generated by the projection of the rim contains points that are potentially, but not necessarily, visible. Projection causes global occlusion, obscuring some of the rim points. For any viewpoint the set of rim points is only potentially visible, and so the rim is divided into two sets: the visible rim and the occluded rim.

Rim: $\mathbf{V} \cdot \mathbf{n}_p = 0$
p is on the rim if the viewpoint vector $\mathbf{V}$ is tangent to $\mathbf{S}$ at p
Visible Rim: $\mathbf{V} \cdot \mathbf{n}_p = 0$; $\nexists q \in \mathbf{S}$ s.t. $q = t\mathbf{V} + (1-t)p, 0 < t < 1$
p is on the visible rim if p is both on the rim and visible from $\mathbf{V}$
Occluded Rim: $\mathbf{V} \cdot \mathbf{n}_p = 0$; $\exists q \in \mathbf{S}$ s.t. $q = t\mathbf{V} + (1-t)p, 0 < t < 1$
p is on the occluded rim if p is on the rim and is not visible from $\mathbf{V}$

2.1 The Rim Appearance

The rim appearance representation can be formally defined as a surface in the cross-product space of viewpoint $\times$ image plane. The space has been termed *aspect space* [Pla88]. Consider a surface patch $\mathbf{S} \subset \mathbb{R}^{\mathbf{3}}$. The rim points on $\mathbf{S}$, dependent on the viewing direction, are expressed by the set $\mathbf{R} = \{\mathbf{n}_p \cdot \mathbf{V} = 0\}$ where $\mathbf{n}_p$ is the oriented normal at $p \in \mathbf{S}$ and $\mathbf{V}$ is a vector on the view sphere. The orthographic projection $\Pi_{(orth,V)}$ of $\mathbf{R}$ maps the rim points on $\mathbf{S}$ to the image plane. The projected rim for the complete space of viewpoints for the surface patch $\mathbf{S}$ forms a hypersurface in $\mathbb{R}^4$ described by $\Pi_{(orth,\mathbf{V})}\{\mathbf{n}_p \cdot \mathbf{V} = 0\}$. We call this hypersurface the *rim surface.*

By restricting the space of viewpoints to a great circle on the view sphere, we obtain a rim surface in $\mathbb{R}^3$, illustrated in Figure 1. The selected viewpoints $\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3$, and $\mathbf{V}_4$ on the left correspond to the contours along the viewpoint axis θ on the right. The surface that is swept through the aspect space ($\mathbb{R}^3$) can be intuitively thought of as a surface formed by occluding contours that are stacked together at each viewpoint along the path. The rim surface for a 1D parametrized curve of viewpoints on the view sphere is a surface in $\mathbb{R}^3$. In general, the rim surface can be expressed as a hyper-surface in $\mathbb{R}^{n+2}$ for n equal to the number of projective degrees of freedom.

There has been recent research in formulating an explicit expression for the rim surface.

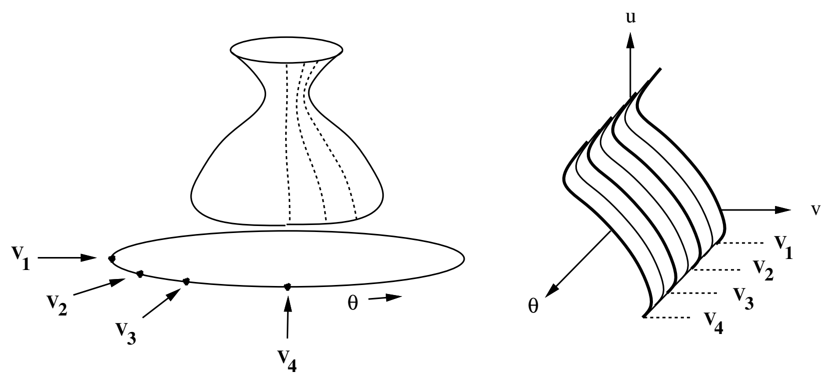


Figure 1: A rim surface in $\mathbb{R}^3$ is generated by restricting viewpoint to a 1D path on the view sphere.

In particular, the locus of points that lie on the folds and creases of the multi-dimensional rim surface are of interest since they are formed by the self-occlusion of the model. Kriegman and Ponce [KP90] have used parametric patches under a weak perspective viewing model in order to formulate equations for the occluding contour of surfaces of revolution. The equations that describe the self-occlusion of the occluding boundary of a torus under orthographic projection is an 8th degree polynomial containing over 170 terms. Others have computed the singularities of the rim surface for the purposes of computing the aspect graph for surfaces of revolution [EB89,KP89] and for parametric surfaces [SJ89]. All of the work with smooth model representations suffers from difficult numerical problems.

Our approach is to compute a simplified, approximate rim surface that is the result of the piecewise-linear occluding contour of polyhedra. The rim appearance representation is an exact representation of the rim surface that is generated by a polyhedral model. Polyhedra eliminate the continuous property of the rim because the set of rim edges changes discretely as viewpoint changes. The rim surface generated by polyhedral edges affords a local way of expressing the singularities from occlusion as piecewise interactions related to individual edges. Each edge can be described as a bounded volume, the boundaries of which are algebraic surfaces and curves with a geometric interpretation resulting from the apparent intersection in the image plane of pairs and triples of unconnected edges.

2.2 The Rim Appearance: Polyhedra

We assume that the polyhedral model is an approximation of a smooth object. Our intent is to study and represent the rim surface that a polyhedral model will generate under the orthographic projection model. This rim surface is completely dependent upon the object-centered model that generates it, and so for polyhedral models, the rim surface is only an approximation of the true rim surface that would be generated by an actual smooth 3D object. The polyhedron, which is linear, affords a simple, local way to explicitly describe the exact behavior of the rim surface. Local pieces of the rim surface are generated by faces and edges of the polyhedron. The piecewise collection of these local patches of the rim surface gives the exact rim surface for the polyhedron. Our assumption is that if the polyhedron is a good approximation of the smooth object, the exact rim surface that we can compute for the polyhedron will also be a good approximation of the model's true rim surface.

We define the occluding contour for polyhedra as the set of edges where, for each edge, exactly one of the two faces that meet at the edge is back-facing. This geometric condition is easily specified as a dot product in terms of the viewpoint $\mathbf{V}$ and the surface normals $\mathbf{n}_i$ of the faces $\mathbf{f}_i$ of the polyhedron. For an edge e formed by f_1 and f_2, e is on the rim when

$$\begin{array}{c}\mathbf{V}\cdot\mathbf{n}_1 > 0\\ \mathbf{V}\cdot\mathbf{n}_2 < 0\end{array} \quad \text{or} \quad \begin{array}{c}\mathbf{V}\cdot\mathbf{n}_1 < 0\\ \mathbf{V}\cdot\mathbf{n}_2 > 0\end{array} \tag{1}$$

These are local conditions that are required for an edge to be on the rim.

Before going farther, it is helpful to reformulate the *rim surface* that was presented for smooth surfaces earlier in terms of the edges of polyhedra. Under the orthographic projection model, a point (x_0, y_0, z_0) in $\mathbb{R}^3$ is projected from the direction (θ, ϕ) with the

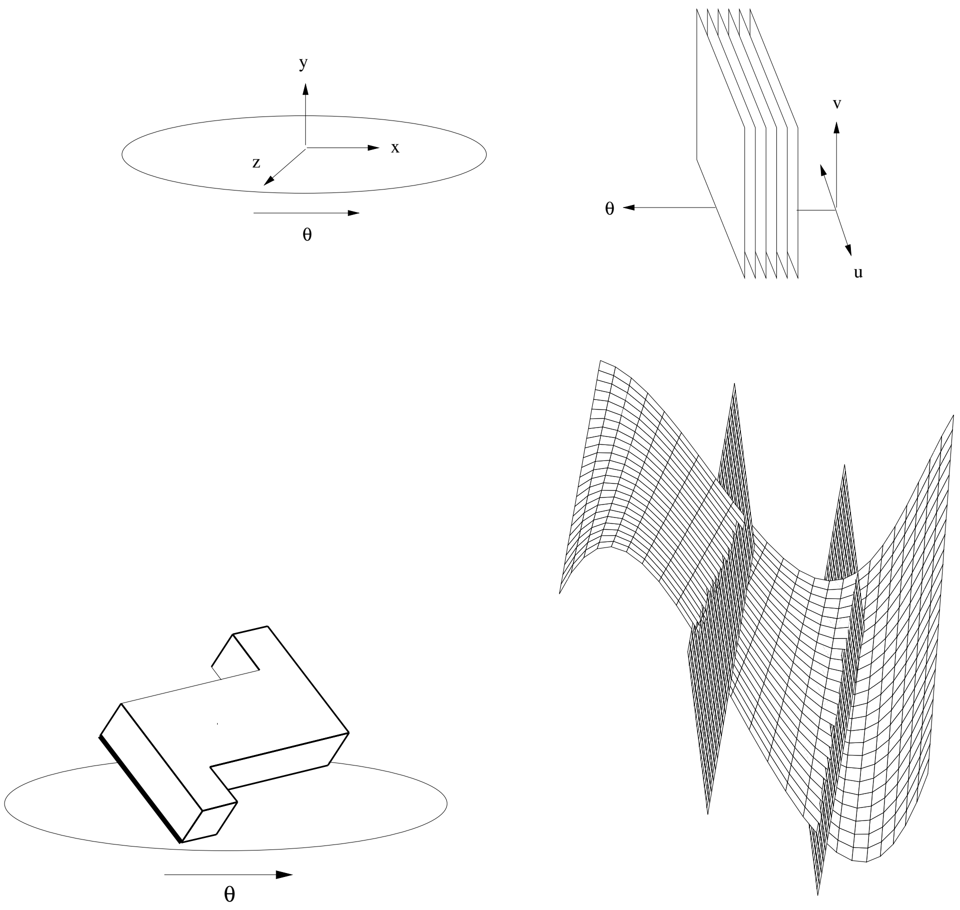


Figure 2: (Top) The viewpoint parameter θ is a rotation of the object in the x-z plane. The aspect space for this one-dimensional viewpoint space is a three-dimensional space with θ shown as the horizontal axis. The image planes (upper right) are formed from the projected scene as a function of θ. (Bottom) The model edge (in bold) forms a hypersurface in aspect space. The grid (lower right) shows the appearance of the edge as a function of viewpoint, where the horizontal viewpoint axis θ coincides with the circular viewpath around the object on the left. The two black planes mark the values of θ where the edge either joins or leaves the rim. The section of the rim surface between the two planes corresponds to the viewpoints where the edge is not on the rim (both faces for the edge are visible.

following transformation:

$$\begin{bmatrix} x_0, & y_0, & z_0 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{2}$$

The coordinates (u, v) of the projected point in the image plane become

$$\begin{aligned} u &= x_0 \cos\theta - z_0 \sin\theta \\ v &= x_0 \sin\theta \sin\phi + y_0 \cos\phi + z_0 \cos\theta \sin\phi \end{aligned} \tag{3}$$

A single point under this transformation traces a hypersurface in $\mathbb{R}^4$, described by the set of points

$$\Pi(\vec{X}) \tag{4}$$

where Π is the orthographic projection of the 3D point $\vec{X}$ to the image space (u, v) from Equation 2. That is, in the coordinates of (u, v, θ, ϕ), a 3D point corresponds to a surface formed from the image coordinates (u, v) of the point as a function of the viewpoint parameters (θ, ϕ).

Thus the rim surface for an edge is parametrized as a hypersurface in the space (u, v, θ, ϕ) that connects the trace of the edge's two endpoints. Specifically, the hypersurface

$$\mathbf{S} = \beta\Pi(\vec{X}_1) + (1 - \beta)\Pi(\vec{X}_2) \tag{5}$$

is a hypersurface in $\mathbb{R}^4$ for $\beta \in [0..1]$ where $\vec{X}_1$ and $\vec{X}_2$ are the endpoints of a 3D edge. To illustrate a rim surface for a single edge, we have plotted these surfaces for the edges in the polyhedron shown in Figure 2. These rim surfaces lie in $\mathbb{R}^3$ because one parameter of the viewing transformation has been fixed (ϕ = constant). The coordinate system is illustrated by the top portion of Figure 2, where the left side shows the viewpoint parameter θ along a great circle, and the right side shows the vertical image planes as a function of the horizontal θ axis. The lower portion of Figure 2 shows the rim surface for an edge of the model for a fixed value of ϕ ($\phi = 0$) with θ varying from 0 to 2π. The linear rulings for each value along the θ-axis on the surface correspond to the projected edge segment at that viewpoint.

Now we can visualize the rim conditions in Equation 1 as a set of boundaries on the rim surface of an edge. Figure 2 shows the planar boundaries positioned at $\theta = 2.95$ and $\theta = 4.39$ where the faces that meet at the edge turn away from the viewer. In general, these boundaries are hyperplanes in the space (u, v, θ, ϕ).

The complete rim surface for the model is a piecewise collection of the rim surfaces for individual rim edges. Self-occlusion alters the rim surface, so we must augment the local definition of the rim surface using geometric conditions that describe the global occlusion that occurs under projection.

2.2.1 Visual Events

One important result that has emerged from the work on the aspect graph is that all changes in self-occlusion with respect to viewpoint in polyhedra can be specified in terms of edge-edge-edge events (EEE-events) in the image plane [PD90,GM90]. The EEE-event is the apparent intersection of three (not necessarily adjacent) edges. A degenerate case of the EEE-event is the edge-vertex event (EV-event), where two of the edges actually meet at a vertex. The geometry and the equations for finding these events is the basis for constructing the rim appearance representation. In object coordinates, a segment $\mathbf{E}_1$ has two endpoints that determine its location in $\mathbb{R}^3$. The projection matrix $\mathbf{\Pi}$ is applied to $\mathbf{E}_1$ in order to generate the image segment e_1. Finding EE-events for two model segments $\mathbf{E}_i$ and $\mathbf{E}_j$ means solving for the set of all $\mathbf{\Pi}$ such that $e_i \cap e_j \neq \emptyset$, i.e., finding the set of all viewpoints that cause the two segments to intersect in the image plane. It has been shown that the set of viewpoints that satisfy the conditions for an EE-event (the apparent intersection of two edges) and an EEE-event are computable as a boundary representation in a higher-dimensional space [PD90,Pla88].

In order to visualize this, Figure 3 shows the rim surfaces in $\mathbb{R}^3$ for two edges from the model (shown in bold) that form a T-junction. The geometry of this interaction is shown in the intersection of their rim surfaces. The equation of this curve of intersection

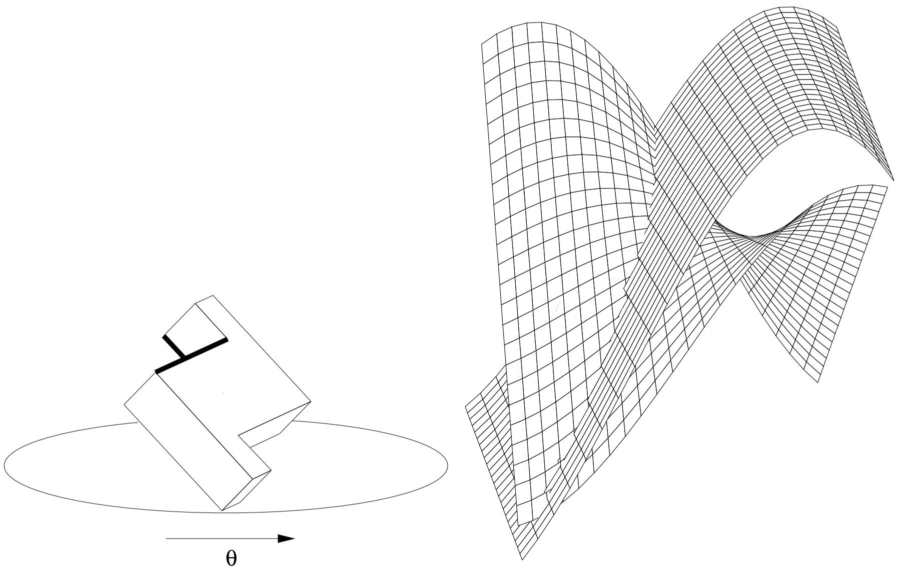


Figure 3: The T-junction formed by a pair of model edges is computed as an intersection in aspect space.

is given by the equations for the two rim surfaces for edge $\mathbf{E}_1$ from X_1 to X_2 and edge $\mathbf{E}_2$ from X_3 to X_4:

$$\begin{gathered} \beta_1 \Pi(\vec{X}_1) + (1 - \beta_1)\Pi(\vec{X}_2) \\ \beta_2 \Pi(\vec{X}_3) + (1 - \beta_2 \Pi(\vec{X}_4) \end{gathered} \tag{6}$$

where

$$\beta_1 = \frac{\mathbf{V} \cdot ((\vec{X}_3 - \vec{X}_1) \times (\vec{X}_4 - \vec{X}_3))}{\mathbf{V} \cdot ((\vec{X}_2 - \vec{X}_1) \times (\vec{X}_4 - \vec{X}_3))} \tag{7}$$

with the viewpoint $\mathbf{V}$ expressed in Cartesian coordinates [Pla88]. This curve of intersection is specified in the boundary representation of the rim surface and is exactly the image location of the T-junction between the two edges. Figure 3 shows the curve where the two rim surfaces intersect for a fixed value of the viewpoint parameter ϕ. Note that the extent along the viewpoint axis of the curve of intersection corresponds to the viewpoints where the T-junction begins and ends. The interested reader should refer to [PD90,Sea91] for more detailed work related to the form of these and higher-order interactions that affect the rim surface of polyhedra.

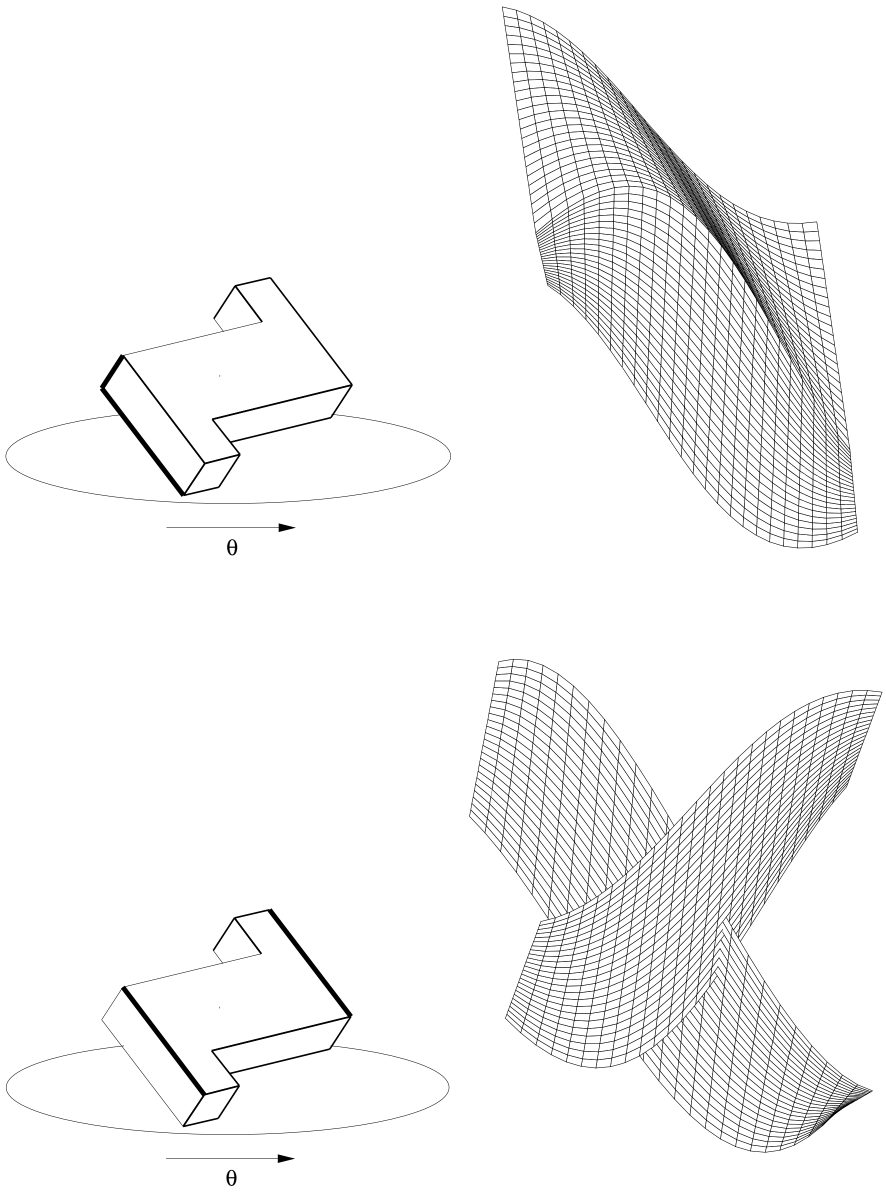


Figure 4: (Top) Edges that are spatially adjacent (i.e., share a vertex) meet at a crease in aspect space. The equation of the crease is the trace of the common vertex. (Bottom) The two bold edges are connected over viewpoint. As one edge leaves the rim the other joins the rim. At the transition viewpoint the two rim surfaces are adjacent in aspect space.

2.2.2 A Construction Algorithm

The complete rim appearance representation (the exact rim surface for a polyhedron) is a collection of the rim surfaces for the edges of the polyhedron. This collection is maintained as a boundary representation (BREP) in the space (u, v, θ, ϕ). The steps below outline our algorithm to construct the rim appearance representation:

1. Compute a bounded structure for each edge that delimits the viewpoints where the edge is locally on the rim.
2. Find EE-events between edges that are part of the rim.
   (a) Modify the structure for each rim edge that is occluded by another. This builds a boundary representation (BREP) with algebraic boundaries specified by the equations for the EV-events and EEE-events [Pla88].
   (b) Organize the final BREP for each edge according to its global visibility. The global visibility test must be made to determine if the edge is fully visible or completely occluded.
3. Connect the individual BREP's together spatially and in viewpoint. Spatial connections are specified by the edge connectivity information in the polyhedral model. BREP's for two distinct edges are connected in viewpoint space at those viewpoints where one edge leaves the rim and the other edge becomes part of the rim, and at visual events.

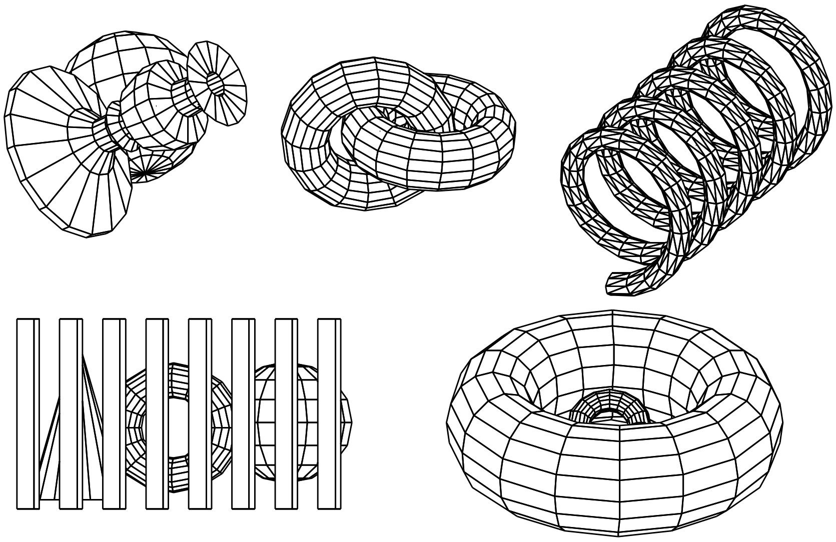
The plots of the rim surfaces shown in Figure 4 illustrate two of the connections that are computed by Step 3 of our algorithm for constructing the BREP of the rim appearance representation. First, two edges that are spatially adjacent in the model (i.e., share a vertex) form adjacent pieces of the rim (hyper)surface. The top half of Figure 4 shows the rim surfaces for two edges that are adjacent in the model. The discontinuous crease in the surface is the trace of the vertex that the two edges share. There is also connectivity in viewpoint space where one edge leaves the rim at the point along the viewpoint axis where another edge joins the rim. The bottom half of Figure 4 shows two edges in the model that form this kind of adjacency along the viewpoint axis. The BREP for the rim surface includes this connectivity information so that the boundary where the first edge leaves the rim and the second one enters the rim (and the edges spatially coincide) is made explicit.

2.2.3 Construction Results

The complexity of constructing the rim appearance representation is bounded by the number of visual events that affect the appearance of the rim. In the worst case, the rim appearance representation can be constructed in space $O(n^4)$ for a polyhedron with n faces. Since the algorithm must compute the potential intersection in the image plane of each rim edge with every other, the construction time is bounded by $O(n^5)$.

These complexity bounds are the same as those for constructing the asp, an intermediate representation for computing the aspect graph [Pla88]. As with the asp, pathological polyhedra such as picket fences and grids can achieve the worst-case behavior. The rim appearance representation, however, has a much better average case behavior because of the elimination of many of the visual events that occur in polyhedra but are not related to the rim or the occluding contour.

To support this claim, Table 1 shows visual event data for several polyhedral models. Column 3 shows the total number of events, and column 4 shows the number of these events that are actually rim events. The rightmost two columns report storage sizes. As the numbers in columns 3 and 4 show, there is approximately a 75% reduction in the number of visual events considered in the rim appearance representation. For example, for the spring model, the average number of visual events per face decreased from 22.2 to 4.9 in the rim appearance representation. These results indicate that the rim appearance representation saves significant time and space over the asp (and hence the aspect graph as well because the aspect graph contains nodes and transitions for *all* changes in the projected line drawing [PD90]) while preserving the completeness of the exact appearance of the occluding contour at all viewpoints.



Model	No. Faces	Total No. Events	No. Rim Events	Total Size (Kbytes)	Rim Size (Kbytes)
Candlestick and sphere	416	5270	1264	120	31
Interlocked tori	512	5135	1294	104	30
Spring	1800	39954	8790	868	220
Large grid	449	9776	2738	216	68
Torus inside torus	512	4395	1010	104	25

Table 1: The total number of events and the number of events affecting the rim were computed for the polyhedral models above. The fourth column shows that the number of visual events computed for each polyhedron is reduced by 75% when considering only the rim edges. The visual events were generated under orthographic projection with one degree of freedom in viewpoint.

3 Viewpoint from Occluding Contour

We present our approach to the problem of recovering a set of viewpoints from the detected occluding contour by first describing the geometry of contour features, the representational organization of these features, and the viewpoint constraints resulting from model-image correspondences. First, we describe how 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 as a function of viewpoint. Second, the steps of the algorithm for searching this representation and making viewpoint hypotheses is outlined and then described in more detail. Finally, we present implementation results and make several observations about the methods.

3.1 Contour Geometry and Organization

Smooth opaque shapes without surface discontinuities generate only T-junctions, smooth occluding contours, and contour terminals in the image plane [Mal87]. We assume here that the polyhedral model is an approximation of a smooth shape. It can then be assumed that none of the polyhedral edges are true surface discontinuities. In general, surface discontinuities can be treated without difficulty, but for simplicity the discussion here is restricted to smooth surfaces. In this case the only stable, or transverse, 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. Consequently, any perturbation of viewpoint within an open disc on the view sphere will cause a triple-contour feature to disappear. Accordingly, we describe in detail here the geometry of the T-junction, caused by the intersection in the image plane of two non-adjacent rim edges. The T-junction is a persistent feature that extends over a 2D set of viewpoints on the view sphere.

3.1.1 The T-junction

The EE-event is represented in the rim appearance representation as part of the boundary of a BREP (hypersurface) in a multi-dimensional space[1] for a rim edge. A set of EE-event hypersurfaces that are adjacent in aspect space (viewpoint $\times$ image plane) establishes a piecewise representation for a single T-junction between two occluding contours. This piecewise-connected set of hypersurfaces approximates a single contour-contour event (CC-event).

[1]Aspect space and its properties are fully developed in [PD90].

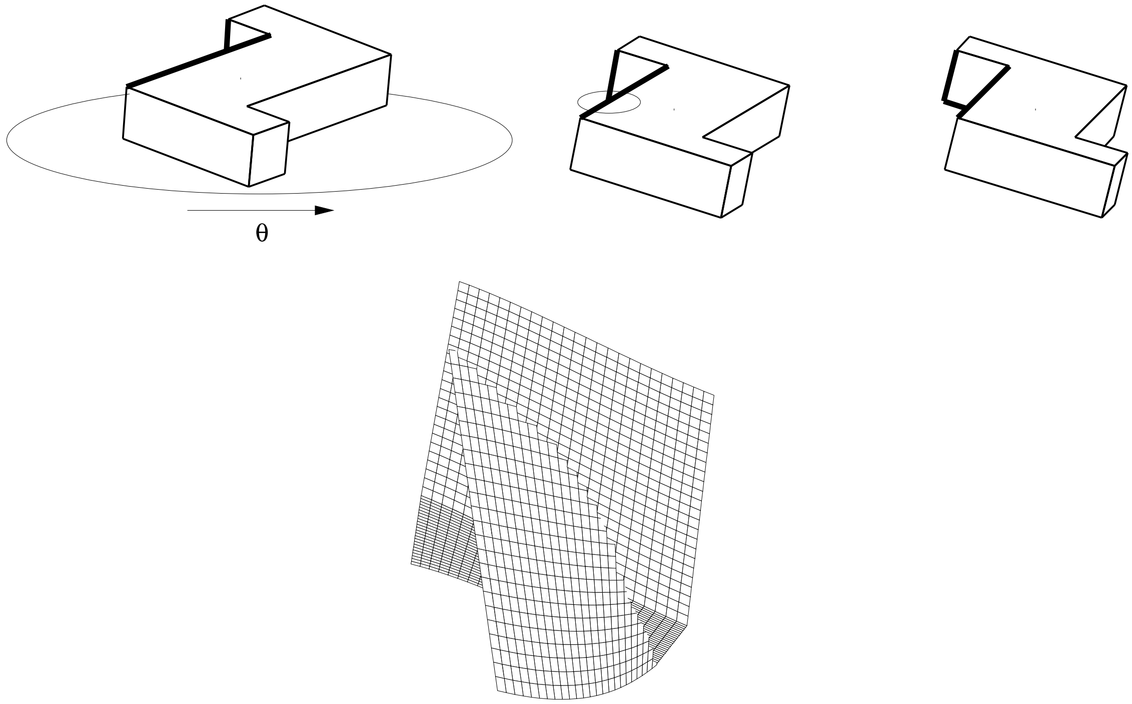


Figure 5: The T-junction between the edges in the leftmost view continues as a T-junction between the edges in the rightmost view. At the transition viewpoint the T-junction curve intersects the crease between the two edges that share a vertex.

The rim surface for a polyhedron is a piecewise collection of individual rim surfaces for the edges. The connectivities of the pieces of the rim surface can be exploited in order to build a larger structure that combines the individual pieces. A *contour-level* structure is a set of pieces of the rim surface that are put together because of various types of adjacencies in space. For example, Figure 5 shows a T-junction that continues to persist across a set of views even though the edges that form the T-junction change. The viewpoint where the edges forming this T-junction change (circled) is the point where the T-junction curve intersects with the vertex-adjacency crease. We link these two T-junctions together as part of the same “virtual” T-junction in order to recapture the same behavior that exists for smooth surfaces.

The CC-event is the T-junction that is produced by the projection of two smooth surfaces. The difficulty in explicitly representing T-junctions for smooth surfaces is the numerical complexity of the locus of surface points that define the viewing directions where T-junctions occur [KP90]. For polyhedra, we assemble the EE-events into piecewise CC-event structures in order to provide an approximation of the smooth T-junction geometry.

There are three important geometric quantities for a T-junction that we represent: a way to compute the image location of the apparent intersection point, the T-junction orientation, and the viewpoint extent of the T-junction. These properties are directly computable from the geometry of the two edges that form a T-junction; the piecewise connection of EE-events represents the change in the T-junction as the rim changes. The orientation and location in the image plane are immediately available from the geometry of the edges that form it.

The geometry of the interaction of two rim edges that form a T-junction is described in part by a region of visibility on the view sphere, as shown in Figure 6(c). Figure 6(a) shows for simplicity only two isolated 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. By the definition of the polyhedral rim, an edge only forms T-junctions for viewpoints where that edge is on the rim. The image coordinates of a projected T-junction for two edges are represented directly as a function of the viewing direction $\mathbf{V}$ and the endpoints of the segments [Pla88]. As shown in Figure 6(b), a viewpoint is modeled as a point (θ, ϕ) on the unit sphere. The orthographic projective transformation for this sphere of viewpoints is rotation by (θ, ϕ) and then orthographic projection in the direction of the z-axis.

There are two conditions that govern the observability of an EE-event between two rim edges. First, the local rim conditions must be satisfied: each edge must be on the rim and must project onto one another in the image plane. Second, the T-junction must be globally visible. Given only the local conditions, an EE-event may be occluded by another part of the surface. The rim appearance representation (BREP) encodes both the local

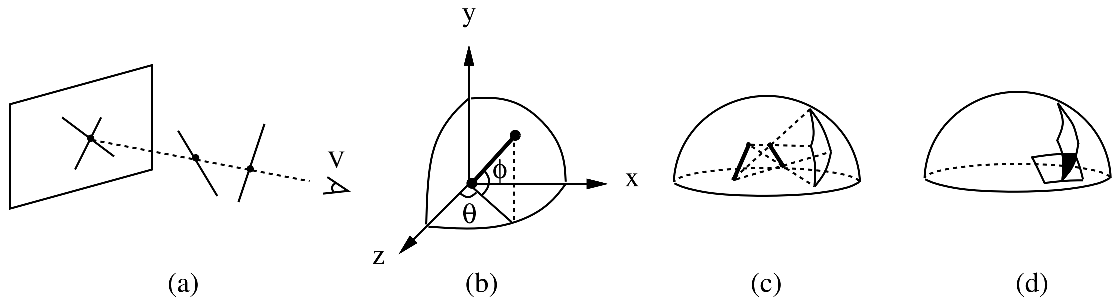


Figure 6: (a) The T-junction formed by a pair of model edges. Both edges are restricted to the rim. (b) The orthographic viewing model. (c) A T-junction persists for a bounded viewing region. (d) Two T-junctions co-occur at the viewpoints where their patches overlap on the viewing sphere.

and global geometry of the T-junctions for a model, including the adjacencies between EE-events.

A set of T-junctions (rather than just one T-junction) can be produced when a non-convex shape is projected from a single viewpoint. The geometric relationships between simultaneously visible T-junctions is of interest because of the additional viewpoint constraints that they provide. These relationships are computed in our rim appearance model using the geometry of the T-junction patch boundaries. Two or more T-junctions that can simultaneously occur in an image define a strictly higher constrained region of viewpoints, i.e., the region defined by the intersection of the patches for each individual T-junction. Conversely, two T-junctions co-occur at the same viewpoint when their respective patches form a non-empty intersection on the view sphere. The region of intersection is the set of viewpoints where the edges are on the rim and the two pairs of edges will each project to a T-junction in the image. Figure 6(d) 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.

3.1.2 The Contour-Contour Event

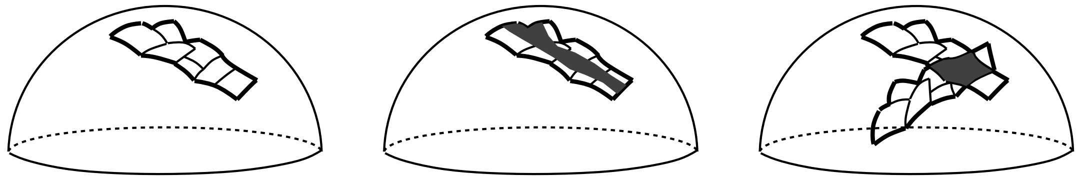
The presentation above explains the well-defined geometric structure for the set of EE-event hyper-surfaces in aspect space that collectively approximate a CC-event. A CC-event is computed directly from a polyhedral model for the purpose of predicting how the occluding contour of the model will appear in the image as a function of viewpoint. An hypothesized correspondence between a CC-event computed from a model and a T-junction in the image produces a set of viewpoint constraints. These constraints are of three forms: CC-event persistence constraints, geometric constraints, and co-occurring feature constraints.

The CC-event persistence constraint eliminates viewpoints where the CC-event cannot occur. That is, if it is hypothesized that a particular image T-junction has been produced by a specific model CC-event, then the viewpoint that created the image projection must lie within the set of viewpoints that produce the CC-event. This set of viewpoints for a CC-event in the model is well-defined with exact boundaries.

Geometric constraints are derived from an assessment of how well the measured characteristics of an image T-junction matches a CC-event in the model. For example, the stem and the T of a detected T-junction meet together at a particular angle. The viewpoints bounding the CC-event are further constrained by boundaries around the viewpoints that will project to a T-junction that is approximately equal to the measured angle of the detected T-junction.

Co-occurring feature constraints are constraints between pairs of CC-events. The set of viewpoints where two CC-events co-occur is in general much more restrictive than the set of viewpoints where a single CC-event occurs. Co-occurring constraints are simply persistence constraints involving a pair of CC-events. Specific geometric conditions describing the relative position and orientation of two CC-events can also constrain the set of viewpoints that may produce a pair of image T-junction features.

These three types of constraints (CC-event persistence, geometric, and co-occurring)



(a) (b) (c)

Figure 7: A CC-event is a piecewise collection of EE-events. (a) The persistence constraint: the outermost boundary (in bold) defines the CC-event region and delimits viewpoints where a single T-junction is visible. (b) The geometric constraint: the shaded viewpoints within the CC-event region are those viewpoints where the T-junction satisfies a detected geometric condition in the image such as a particular measured angle or orientation. (c) Co-occurring feature constraints: the shaded viewpoints are those viewpoints where two spatially-distinct CC-events co-occur in the image.

are illustrated in Figure 7. The figure shows a set of EE-events on the view sphere that are joined together to form a single CC-event. In Figure 7(a), the outer boundary of the union of EE-event boundaries is the CC-event boundary. Viewpoints within this bounded region will produce a T-junction in the image. A subset of the viewpoints within this CC-event region will satisfy specific geometric requirements from the observed image feature. Figure 7(b) illustrates how a subset of viewpoints satisfy specific geometric measurements such as relative orientation and angle of a T-junction. Viewpoints where two CC-events co-occur are computed exactly as an intersection of viewpoint regions on the view sphere. The shaded region in Figure 7(c) is the set of viewpoints where two model T-junctions will co-occur.

For a given viewpoint (θ, ϕ), the rotation about the optical axis and translation in the image to bring a CC-event into correspondence with an image T-junction is well-defined. This is a result of the fact that a T-junction is *oriented*, with the stem of the T-junction being on the unoccluded side of the other edge. Figure 8 shows two model edges being aligned with an image T-junction. The alignment transformation applied to the two edges of an EE-event is of the form

$$R_y(\theta)R_x(\phi)R_z(\alpha) + \tau \tag{8}$$

This is exactly the (θ, ϕ) rotation (given) and a rotation α about the z-axis. The translation τ translates the vertex of the model T-junction to the coordinates of the vertex of the image T-junction.

Because we assume that the orientation of a T-junction has been detected, the rotation α can be computed directly by measuring the angle between the image segments and the 3D model segments at the viewpoint (θ, ϕ). Specifically, let e_1, e_2 be image segments that form a T-junction, and $\mathbf{E}_1, \mathbf{E}_2$ model edges of the EE-event. We can assume without loss of generality that e_1 occludes e_2, and $\mathbf{E}_1$ is in front of $\mathbf{E}_2$. A single viewpoint (θ, ϕ) within the EE-event region determines the rotations $R_y(\theta)$ and $R_x(\phi)$ for the orthgraphic projection mapping that creates the image of the EE-event. The rotation matrix $R_z(\alpha)$ about the z-axis is given by

$$\mathrm{R_z}(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

where

$$\cos(\alpha) = \frac{e_1 \cdot \mathbf{E}_1}{||e_1|| ||\mathbf{E}_1||} \tag{10}$$

This is the angle between the projected model edge $\mathbf{E}_1$ and the image edge e_1 at the viewpoint (θ, ϕ). The direction of α is determined by the relative orientation of $\mathbf{E}_2$ with respect to e_2.

To summarize, a model-image correspondence constrains viewpoint since a CC-event exists over a bounded set of viewpoints. There are also geometric constraints such as the measured angle of a detected T-junction. Since a CC-event is represented as a piecewise set of adjacent EE-event hypersurfaces, the exact geometry for each piece of the CC-event is available.

Finally, it should be noted here that the viewpoints within a single EE-event that best match the stem and occluding edge of the image junction are difficult to compute.

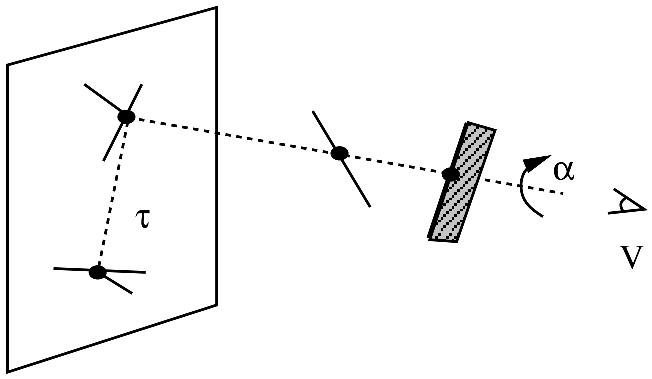


Figure 8: The T-junction formed by a pair of model edges is aligned with an image T-junction by a rotation α about the optical axis and an image plane translation τ to align the junction point.

In general, there is a locus of viewpoints within an EE-event patch that can produce a T-junction that exactly matches an image T-junction. The variation in the geometry of an EE-event depends on the size of the viewpoint region where it occurs [BWR90]. In other words, the solution for the model-to-image correspondence matrix for a T-junction is not unique given only the two model segments and the image T-junction. We can, however, approximate the geometry of the EE-event by chosing a single, representative viewpoint. This approximation is good since the geometry of a T-junction that persists over a small region of viewpoint space changes very little over that region. The approximation becomes worse, of course, for T-junctions that persist over large regions in viewpoint space.

3.2 An Algorithm for Feature Selection and Refinement

The organization of occluding contour information and the viewpoint constraints associated with each individual occluding contour feature is the basis for obtaining a viewpoint solution in the form of a small bounded set of viewpoints. We have implemented the following algorithm using T-junctions and contour fragments in order to solve for a set of viewpoints matching an image:

Step 1: Detect in the image an occluding contour feature, e.g., a T-junction. Measure the geometric properties of this feature such as orientation and angle sizes.

Step 2: Select candidate CC-events from the rim appearance representation that have one or more constituent EE-events with similar geometric properties to the image T-junction. This set of correspondences between a model CC-event and an image T-junction completely determines a set of transformations that aligns the model with the image (because a T-junction is oriented, and the EE-event geometry is approximated by a single viewpoint within the EE-event region).

Step 3: Verify and rank the candidate matches by comparing other occluding contour features with the model using the transformation determined in Step 2.

Step 4: For each of the verified correspondences, compute the maximal viewpoint region that matches the image contours. This step refines the initial viewpoint region associated with the match in Step 2 by searching neighboring EE-events in the rim appearance representation to find additional matches.

The problem is formulated as a search problem through the rim appearance representation for a given model. This search consists of an indexing phase (Step 2), where an initial correspondence between model and image features results in a restricted set of viewpoints, and a search phase (Step 4) where the results of the indexing phase are refined by using the piecewise structure of the CC-event in the rim appearance representation. The following sections describe Steps 2 and 4 in more detail.

3.2.1 CC-Event Selection

Contour features can be organized by characteristics such as curvature, relative orientation to other features, and persistence in viewpoint space. For example, a T-junction can be described by 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. For initial evaluation purposes, the implementation results we present make use of only T-junction orientation and angle information. Contour-contour interactions are selected from the rim appearance representation for a model based on precomputed orientation and angle geometry.

The angle formed by a T-junction between two edges is relatively stable over a small

Figure 9: The geometric relationship of high-curvature contour fragments to an EE-event reduces the number of CC-events that match a T-junction.

range of views. Since the CC-event is represented as a set of EE-events, the angle for a CC-event can be represented piecewise, where each edge pair has a fixed, precomputed angle. A CC-event and an image T-junction correspond where the measured angles of the two are similar. The set of CC-events is organized by this geometric measure so that portions of a CC-event with particular values can be selected efficiently.

In addition to the EE-event itself, a set of occluding contour fragments with high curvature values is stored. The orientation of each contour fragment is computed with respect to the coordinate system defined by the oriented EE-event. This small number of contour fragments provides an efficient template that can be used to verify the accuracy of a match between an EE-event and a detected T-junction. Precomputed contour fragments that are supported by evidence in the image increase the confidence of the match. Figure 9 shows an EE-event (circled) and the parts of the occluding contour that appear with high curvature values over the set of viewpoints where the EE-event persists. The relative geometry (with respect to the T-junction) of each of the occluding contour fragments is precomputed and stored with the EE-event.

In summary, the indexing phase of this algorithm (Step 2) selects candidate CC-events from the rim appearance representation based on the measured angle of a T-junction and the relative position and orientation of high-curvature portions of the contour with respect to the T-junction. The top candidates (Step 3) are passed on to the search component of the algorithm. Next, the search phase (Step 4) uses the candidate correspondences as starting points in order to compute a bounded region of viewpoints that best accounts for the observed image contours. The next section outlines this final phase.

3.2.2 Refining CC-Event Correspondence

Once the initial correspondences are made between the model and an image feature, the persistence, geometric and co-occurring constraints described in Section 3.1.2 are applied to guide a local search for the boundaries of the solution region. Specifically, a candidate, qualitative correspondence between a CC-event and an image T-junction is refined by searching the EE-events that together define the CC-event. The EE-events, as members of the CC-event, may not all satisfy the measured geometric properties of the image T-junction. Thus the hypothesized correspondence between a CC-event and a T-junction can be refined by searching the set of precomputed EE-events in the model through aspect space.

This search has been implemented as a fixed-distance search over the set of adjacent EE-events within the CC-event in aspect space. A fixed number of adjacent regions is examined to find a set of boundaries within the CC-event that best satisfies the persistence, geometric and co-occurring constraints imposed by the image data and the hypothesized correspondence. The CC-event with the best measured match is selected as the solution region. Note that the search is over adjacent EE-events both spatially and in viewpoint. These adjacencies, as illustrated in Figures 4 5, are stored explicitly in the rim appearance representation BREP and guide the search for the boundaries of the solution region in viewpoint space.

3.3 Implementation Results

A prototype system has been implemented in order to study the issues involved with a model-based, explicit representation of occluding contour features. We have implemented

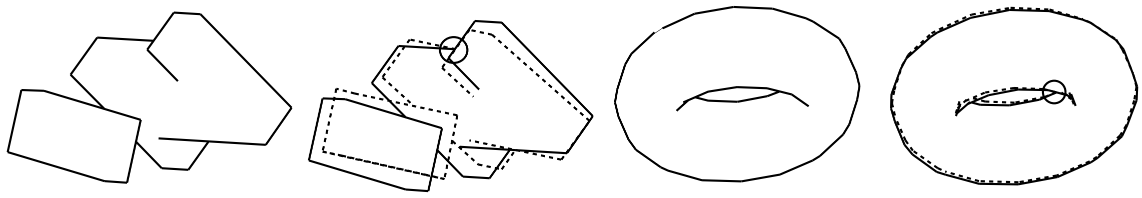
Model	Faces	Computation Time (Secs)	Number of EE-events	Average EE-event Surface Area (%)
Candlestick and sphere	416	86.9	6734	2.0
Interlocked tori	512	129.8	6008	5.0
Spring	1800	528	33029	2.3
Large grid	449	140.5	13486	1.6
Torus inside torus	512	130.3	5494	4.4

Table 2: Size and time information for the models. Average EE-event patch area is the percentage of the entire area of the view sphere over which the EE-event occurs.

the algorithms to compute the rim appearance representation under orthographic projection. The results reported 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, we describe a prototype implementation that recovers a viewpoint region estimate from a set of synthetically-generated image features. All of the viewpoints in the solution viewpoint region produce an occluding contour that matches the image data.

The models are shown above Table 1. Table 2 indicates the model size, the total number of EE-events computed for each model, and the average area of each edge-edge interaction, computed 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 Table 2. 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 $O(n^2)$ where n is the number of edges. Rim



(a) (b) (c) (d)

Figure 10: The solid lines in (a),(c) are the images used as input to the algorithm. The dashed lines in (b),(d) are the recovered views. The circled T-junctions in (b) and (d) were used as the initial feature for selecting a CC-event from the precomputed model of the occluding contour. The dashed lines are the projection of the model from the middle of the solution viewpoint region.

constraints and non-degenerate polyhedral models give an average size complexity that is much smaller than this worst-case size. Second, the average persistence of EE-events is inversely proportional to the 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.

Figure 10 shows the computational results of determining a starting viewpoint given a synthetic image generated from an arbitrary viewpoint. 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) at the center of the solution EE-event region, a rotation α about the optical axis, and a translation τ in the image plane.

The circled T-junctions in Figure 10 were hand-selected from the synthetic input and used as the primary feature for correspondence with the precomputed representation of the occluding contour. The measured angle of the T-junction was used as the primary indexing key, with an initial match being any model T-junction with an angle within 10 degrees of the measured value. The relative position of the occluding contour to the oriented T-junction was used to further constrain potential correspondences. This is similar in style to the local feature focus method [BC82]. A measure of the degree of correspondence was determined by using a least squares distance measure between predicted contour fragments and image contours.

An exact solution for viewpoint is not found, but rather a constrained region of viewpoints is found that accounts for the T-junction and occluding contour data. Given a tightly-constrained set of viewpoints, an exact solution can be found using, for example, an iterative method [Low87]. Furthermore, a contour correspondence is found, not an exact edge-to-edge correspondence. The two model edges aligned with the T-junction in Figure 10(d) 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. The solution viewpoint displayed is the center of the final solution viewpoint region. For our set of test models, all solution regions contained a true solution, and all viewpoints within each solution region were within $\pi/4$ radians of the exact solution.

Figure 10 shows the input image edges as solid lines, and the projected model at the center of the solution region viewpoint as dashed lines. The T-junction used as the central feature is circled. For the torus image, ten edge-edge correspondences within the CC-event for the circled T-junction were made by the algorithm before the solution shown in Figure 10(d) was found. The selected CC-event correspondence was revised after measuring the degree of match of the predicted occluding contour edges to the image. A correspondence was revised by using CC-event information in the model to obtain other edges that project to the same T-junction. This search was limited by the distance in the model from the original CC-event and the angle of the projected CC-event. Three CC-event correspondences were required to obtain the view shown of the S-shaped polyhedron in Figure 10(b).

An interesting problem occurs with "coarse" polyhedral models such as the S-shaped polyhedron shown in Figure 10(a). 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 their projected angle values will each vary over a large range. As expected, large variations for a single EE-event makes any single choice in values for that EE-event very bad. For this case, an EE-event cannot be adequately represented by a single set of angle values. This again illustrates the time-space tradeoff in representational accuracy. Additional contour feature information incorporated into the algorithm can help to solve this problem.

3.4 Remarks

We can make several observations and conclusions based on this prototype implementation. First, convex shapes provide no occlusion-based information, although the interaction between several convex polyhedra gives occlusion cues that constrain viewpoint. In the convex case, only the curvature extrema and the shape description of the occluding contour gives information about viewpoint. Second, symmetry causes a larger number of good model-image matches since a symmetrical model generates a locus of viewpoints with identical contours. In this case, any of the possible correspondences on the locus of identical views is valid. Finally, the initial analysis of this approach has not incorporated other shape-based information that is almost always present in real 3D modeling situations. We have studied the occluding contour alone with the future goal to add other shape information as well as texture, color and surface markings, to provide a larger set of constraints for viewpoint determination.

4 Summary and Conclusions

Features of the occluding contour contain strong constraints for viewpoint determination. These constraints can be used to solve exactly for a region of viewpoints where a model will project to the observed occluding contour features. Our novel approach relies on the precomputed occluding contour features produced by polyhedra under orthographic projection. These precomputed features include contour T-junctions and the relative arrangement of sections of the contour. The basis for the computation of this information is the construction, for all viewpoints, of the rim surface for polyhedra, defined as the analog of the rim surface for smooth shapes.

Occluding contour features are selected from the model given measured contour features and their relationships. Viewpoint constraints are associated with these qualitative correspondences since occluding contour features are defined by small regions of viewpoints where they can occur. The implementation results demonstrate that a model-based strategy using self-occlusion features such as T-junctions can efficiently constrain viewpoint by accounting for the occluding contour features detected in an image.

The constrained viewpoint region that is computed using the the occluding contour can be used together with other methods for exact viewpoint recovery that must assume a starting point or an initial, approximate solution. For example, in the case of iterative methods [Low87], a starting viewpoint can be obtained from a constrained viewpoint set. Parameter space methods [TM87] that can avoid a costly search of the entire space of transformations become much more efficient. The search of an interpretation tree [Gri90] is more efficient when there are global constraints on the possible solutions. A constrained viewpoint region can help aspect graph methods [BESS89,KP90,Ike87] address the problem of how to select a few aspects to test from a large number of potential aspects.

References

[BC82] R. Bolles and R. Cain. Recognizing and locating partially visible objects: The local-feature-focus method. *Int. J. Robotics Research*, 1(3):57–82, 1982.

[BESS89] K. Bowyer, D. Eggert, J. Stewman, and L. Stark. Developing the aspect graph representation for use in image understanding. *Proc. Image Understanding Workshop*, pages 831–849, 1989.

[BWR90] J. B. Burns, R. Weiss, and E. M. Riseman. View variation of point set and line segment features. *Proc. Image Understanding Workshop*, pages 650–659, 1990.

[CB90] R. Cipolla and A. Blake. The dynamic analysis of apparent contours. *Proc. 3rd Int. Conf. Computer Vision*, pages 616–623, 1990.

[EB89] D. Eggert and K. Bowyer. Computing the orthographic projection aspect graph of solids of revolution. *Proc. IEEE Workshop on Interpretation of 3D Scenes*, pages 102–108, 1989.

[FH86] O. Faugeras and M. Hebert. The representation, recognition, and locating of 3-D objects. *Int. J. Robotics Research*, 5(3):27–52, 1986.

[GM90] Z. Gigus and J. Malik. Computing the aspect graph for line drawings of polyhedral objects. *IEEE Trans. Pattern Analysis and Machine Intell.*, 12(2):113–122, 1990.

[Gri90] W. E. L. Grimson. *Object recognition by computer: The role of geometric constraints*. MIT Press, Cambridge, Mass., 1990.

[Ike87] K. Ikeuchi. Generating an interpretation tree from a CAD model for 3D object recognition in bin-picking tasks. *Int. J. Computer Vision*, pages 145–165, 1987.

[KD91] Kiriakos N. Kutulakos and Charles R. Dyer. Recovering shape by purposive viewpoint adjustment. Technical Report 1035, University of Wisconsin - Madison, August 1991.

[Koe84] J. J. Koenderink. What does the occluding contour tell us about solid shape? *Perception*, 13:321–330, 1984.

[Koe90] J. J. Koenderink. *Solid Shape*. MIT Press, Cambridge, Mass., 1990.

[KP89] D. J. Kriegman and J. Ponce. Computing exact aspect graphs of curved objects: Solids of revolution. *Proc. IEEE Workshop on Interpretation of 3D Scenes*, pages 116–122, 1989.

[KP90] D. J. Kriegman and J. Ponce. On recognizing and positioning curved 3D objects from image contours. *IEEE Trans. Pattern Analysis and Machine Intell.*, 12(12):1127–1137, 1990.

[KvD79] J. J. Koenderink and A. J. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.

[Low87] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intell.*, 31:355–395, 1987.

[Mal87] J. Malik. Interpreting line drawings of curved objects. *Int. J. Computer Vision*, 1:73–103, 1987.

[PD86] W. H. Plantinga and C. R. Dyer. An algorithm for constructing the aspect graph. *Proc. IEEE Symp. Foundations of Computer Science*, pages 123–131, 1986.

[PD90] W. H. Plantinga and C. R. Dyer. Visibility, occlusion, and the aspect graph. *Int. J. Computer Vision*, 5(2):137–160, 1990.

[Pla88] W. H. Plantinga. *The Asp: A Continuous, Viewer-centered Object Representation for Computer Vision.* PhD thesis, Computer Science Department, University of Wisconsin-Madison, 1988.

[RDW88] W. Richards, B. Dawson, and D. Whittington. *Encoding contour shape by curvature extrema*, pages 83–98. MIT Press, Cambridge, Mass., 1988.

[SD91] W. B. Seales and C. R. Dyer. Constrained viewpoint from occluding contour. *Proc. IEEE Workshop on Directions in Automated CAD-based Vision*, June 1991.

[Sea91] W. B. Seales. *Appearance models of three-dimensional shape for machine vision and graphics.* PhD thesis, Computer Science Department, University of Wisconsin-Madison, August 1991.

[SJ89] T. Sripradisvarakul and R. Jain. Generating aspect graphs for curved objects. *Proc. Workshop on Interpretation of 3D Scenes*, pages 109–115, 1989.

[TM87] D. W. Thompson and J. L. Mundy. Three-dimensional model matching from an unconstrained viewpoint. *Proc. IEEE Int. Conf. Robotics and Automation*, pages 208–220, 1987.

[VF91] R. Vaillant and O. Faugeras. Using extremal boundaries for 3D object modeling. *IEEE Trans. Pattern Analysis and Machine Intell.*, 1991.