An Occlusion-Based Representation of Shape for Viewpoint Recovery

W. Brent Seales*
INRIA Sophia-Antipolis,
2004, Route des Lucioles
Sophia-Antipolis, France
seales@sophia.inria.fr

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 and using a viewer-centered representation of the occluding contour of polyhedra. The representation is computed from a polyhedral model under orthographic projection for all viewing directions. Using this representation, we derive constraints on viewpoint correspondences between image features and model contours. Our results show that the occluding contour, computed in a model-based framework, can be used to strongly constrain the viewpoints where a 3D model matches the occluding contour features of the image.

Keywords: occlusion, occluding contour, rim appearance representation, viewer-centered

1 Introduction

In this paper, we present an approach to modeling the occluding contour of polyhedra. The *rim appearance representation* encodes how the rim projects to the image to form the occluding contour for all viewpoints [11]. This 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 surface where the visual direction is tangent. Our approach deals with the occluding contour produced by polyhedra, and is strongly model-based. Rather than reconstructing 3D surface parameters from image contours [2, 12], 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 of occlusion (such as T-junctions) more explicit. By relating these precomputed occlusion features, which are strongly related to viewpoint, to image features, a set of viewpoints is recovered.

*Permanent address: Computer Science Dept, University of Kentucky, Lexington, Kentucky 40536

The problem of constraining viewpoint for a particular model is often considered a subproblem of model-based 3D object recognition [9]. 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, 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. Second, the shape and topology of the occluding contour is usually stable over a range of viewpoints but is not generated by a fixed part of the shape.

The next section presents the features of the rim appearance representation. Section 3 shows experimental results and gives the details of the algorithm we have developed for organizing 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 the moderate size of the rim appearance representation and the constrained viewpoint regions that can be recovered from image features such as T-junctions. Section 4 summarizes our results.

2 Representing the Occluding Contour

The projection mapping creates occluding contours, generated by points on a surface where the viewing direction is tangent, and opaque shapes cause 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 [7]. We avoid the numerical problems of smooth surfaces by using the edges of polyhedra [4]. 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 for an edge is specified as $\mathbf{V} \cdot \mathbf{n}_1 > 0$, $\mathbf{V} \cdot \mathbf{n}_2 < 0$ where $\mathbf{V}$ is the viewpoint and $\mathbf{n}_i$ are the surface normals of the faces of the polyhedron. This local condition for an edge to be on the rim is augmented with geometric conditions that describe the global occlusion that occurs under projection. Global occlusion is specified in terms of edge-edge-edge events (EEE-events) in the image plane [10, 5]. The EEE-event is the apparent intersection of three (not necessarily adjacent) edges.

The rim appearance representation is formally defined as a surface in the crossproduct space of viewpoint $\times$ image plane. The space has been termed aspect space [10].

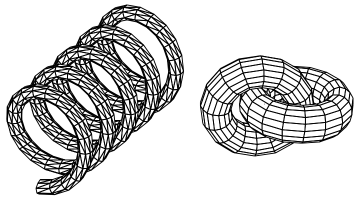
There has been recent research in formulating explicit expressions for the locus of points that lie on the folds and creases of the multi-dimensional rim surface [8]. Work with smooth model representations suffers from difficult numerical problems. The rim surface generated by polyhedral edges affords a local way of expressing the singularities from occlusion as piecewise interactions related to individual edges.

The steps below outline an algorithm to construct the rim appearance representation:

1. Compute a structure for each edge that delimits the viewpoints where the edge is on the rim
2. Find EE-events between edges on 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- and EEE-events [10].
   (b) Organize the final BREP for each edge according to its global visibility.
3. Connect the individual BREP's together spatially and in viewpoint. Spatial connections come from edge connectivity information; viewpoint connections arise from visual events.

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 $O(n^4)$ space and $O(n^5)$ time for a polyhedron with n faces. These bounds are the same as those for constructing the asp, an intermediate representation for computing the aspect graph [10]. 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 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.

Table 1 shows visual event data for two polyhedral models. There is approximately a 75% reduction in the number of visual events to be 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.



	Spring	Tori
Faces	1800	512
Total Events	39954	5135
Rim Events	8790	1294
Total Size (Kbytes)	868	104
Rim Size (Kbytes)	220	30

Table 1: The number of visual events is reduced by 75% when considering only the rim edges.

3 Viewpoint from Occluding Contour

Our approach to the problem of recovering a set of viewpoints from the detected occluding contour uses the geometry of contour features organized in the rim appearance representation.

Smooth opaque shapes without surface discontinuities generate only T-junctions, smooth occluding contours, and contour terminals in the image plane [7]. We assume here that the polyhedral model is an approximation of a smooth shape, and so we assume that none of the polyhedral edges are true surface discontinuities, although they can be treated in this framework without difficulty. Thus the only stable junctions in the smooth occluding contour are T-junctions and contour terminals. Triple-contour (or higher order) intersections can occur, but are not stable. We focus on the T-junction, a persistent feature that extends over a 2D set of viewpoints on the view sphere.

3.1 The T-junction

All T-junctions made by the projected model are represented in the rim appearance representation. A set of EE-events establishes a piecewise representation for a single T-junction between two occluding contours. There are three geometric quantities for a T-junction that we represent: image coordinates, orientation, and viewpoint extent. These properties are directly computable from the geometry of the two edges that form a T-junction. There are two conditions that govern the viewpoint extent of an EE-event between two edges: each edge must be on the rim and must project onto one another in the image, and the T-junction must be globally visible.

A T-junction is in fact an interaction between two parts of the occluding boundary. For polyhedra, the edges forming a T-junction can change while the T-junction itself still arises from the same two parts of the surface. Thus we store

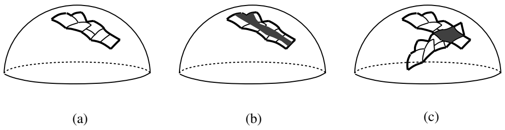


Figure 1: A CC-event is a piecewise collection of EE-events. (a) The persistence constraint (b) The geometric constraint (c) Co-occurring feature constraints

a contour-contour event as a collection of edge-edge events that together give rise to the same underlying T-junction. 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. 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. 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. Co-occurring feature constraints are constraints between pairs of CC-events. An hypothesized correspondence involving one CC-event must also include

These constraints (CC-event persistence, geometric, and co-occurring) are illustrated in Fig. 1. The CC-event boundary is the boundary of the union of EE-events (Fig. 1(a)). Viewpoints within this bounded region will produce a T-junction in the image. Fig. 1(b) shows that only a subset of viewpoints satisfy specific geometric measurements, such as relative orientation and angle of a T-junction. Fig. 1(c) shows viewpoints where two CC-events co-occur are computed exactly as an intersection of regions on the view sphere.

Edges from an EE-event can be aligned with an image T-junction for any of the viewing directions within the bounded region where the CC-event occurs. Because a T-junction is oriented, for a specific viewpoint, the rotation about the optical axis and translation in the image to bring the CC-event into correspondence with the image T-junction is well-defined. A rotation α about the optical axis orients the edges, and a translation τ in the image plane completes the alignment. Note that the viewpoints producing a single EE-event that best matches an image T-junction are difficult to compute. But since the variation in the geometry of a T-junction formed by two edges depends on the size of the viewpoint patch where it occurs [1], the geometry of a T-junction that persists over a small region of viewpoint space changes very little.

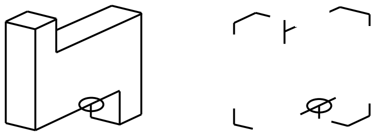


Figure 2: The geometric relationship of high-curvature contour fragments to an EE-event reduces the number of CC-events that match a T-junction.

3.2 Algorithm for Viewpoint Determination

We have implemented the following algorithm using T-junctions and contour fragments in order to solve for a set of viewpoints matching an image:

---

1. Detect and measure in the image an occluding contour feature, e.g., a T-junction.
2. Select candidate CC-events from the rim appearance representation that match the measured properties of the image T-junction. Since we assume that a T-junction is oriented and approximated by a single viewpoint, this correspondence completely determines the model-to-image transformation.
3. Verify and rank the candidate matches by comparing other occluding contour features with the model using the transformation computed in step 2.
4. For each of the verified correspondences, compute the maximal viewpoint region that matches the image contours. This step refines the initial viewpoint region from step 2 by searching neighboring EE-events in the rim appearance representation.

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

The implementation results we present use T-junction orientation and angle information. Contour-contour interactions are selected from the rim appearance representation for the model based on precomputed orientation and angle geometry. The angle for a CC-event is represented piecewise, where each edge pair has a fixed, precomputed angle.

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

	Spring	Tori
Faces	1800	512
Computation Time (Secs)	528	130
Visible Patches	33029	6008
Average Patch Area	2.3	5.0

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

(step 3). 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. Fig. 2 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.

Once the initial correspondences are made between the model and an image feature, the persistence, geometric and co-occurring constraints described earlier 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. This search has been implemented as a fixed-distance search over the set of adjacent EE-events within the CC-event. 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.

3.3 Implementation Results

First, we have implemented the algorithms to compute the rim appearance representation under orthographic projection [11]. The task of computing and storing the contour information for medium-sized polyhedral models is shown to be very manageable. Second, we have implemented the algorithm to recover a viewpoint region estimate from an image that has been rendered using a raytracer and then segmented using standard edge detectors.

The two 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.

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 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, accurate models and smaller, coarser models.

Fig. 3 shows the computational results of computing a viewpoint region given an image generated by raytracing a model under perspective projection. The original image (Fig. 3(a)) was the input to a Canny edge detector [3], which extracted the linked image contours (Fig. 3(b)). The polygonal approximation of these contours (Fig. 3(c)) was obtained using methods developed at INRIA [6]. The occluding contours were extracted from the set of polygonal edges by hand (Fig. 3(d)). We are working on algorithms to automatically detect occluding boundaries using stereo and motion [12].

Each image was represented as a set of individual edges, with T-junctions segmented by hand. Although the image was generated using perspective projection, the recovered viewpoint solution for a model-image match is a viewing direction (θ, ϕ) on the view sphere under othographic projection. At the center of the solution EE-event region, we solve for a rotation α about the optical axis, and a translation τ in the image plane to align the model with the image.

Fig. 3(f) shows the input image edges as solid lines, and the projected model at the solution viewpoint as dashed lines. The leftmost T-junction of the torus was used as the central feature. 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. The solution viewpoint displayed is the center of the final solution viewpoint region. For the torus shown, ten edge-edge correspondences within the CC-event for the circled T-junction were made by the algorithm before the solution shown in Fig. 3(e),(f) was found. The selected CC-event correspondence was revised after measuring the degree of match of the predicted occluding contour edges to the image.

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 approach relies on the precomputed occluding 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,

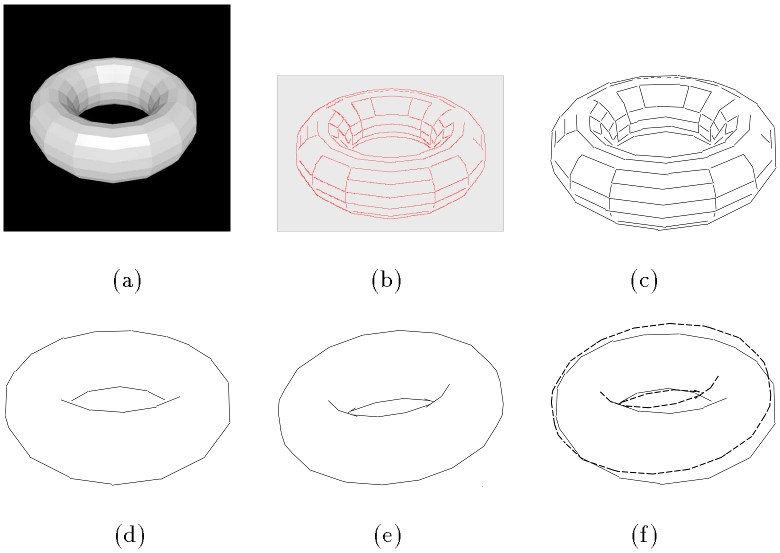


Figure 3: (a) The original raytraced image. (b) Linked edge points from the Deriche-Canny detector. (c) Polygonal approximation of the linked edge points. (d) Hand-segmented occluding contours. (e) The recovered view, projected under orthographic projection. (f) The recovered view of the model (dotted) superimposed onto the extracted occluding contours.

of the rim for polyhedra.

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.

Our current work is focusing on the robust segmentation of the occluding boundary, and on integrating our representation of the features of occlusion with other shape characteristics for model-based vision.

References

[1] 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.

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

[3] R. Deriche. Using Canny's Criteria to Derive an Optimal Edge Detector Recursively Implemented. In *The International Journal of Computer Vision*, volume 2, pages 15–20, April 1987.

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

[5] 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.

[6] G. Giraudon. Chaînage efficace de contour. Rapport de Recherche 605, INRIA, February 1987.

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

[8] 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.

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

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

[11] W. B. Seales and C. R. Dyer. Viewpoint from occluding contour. *CVGIP: Image Understanding*, 55(2):198–211, March 1992.

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