

Computing Exact Aspect Graphs of Curved Objects: Parametric Surfaces*

Jean Ponce
Beckman Institute
University of Illinois
Urbana, IL 61801

David J. Kriegman
Center for Systems Science
Yale University
New Haven, CT 06520-1968

Abstract

This paper introduces a new approach for computing the exact aspect graph of curved objects observed under orthographic projection. Curves corresponding to various visual events partition the Gaussian sphere into regions where the image structure is stable. A catalogue of these events for piecewise-smooth objects is available from singularity theory. For a solid bounded by rational parametric patches and their intersection curves, it is shown that each visual event is characterized by a system of n polynomials in $n + 1$ variables whose solutions can be found by numerical curve tracing methods. Combining these methods with ray tracing, it is possible to characterize the stable image structure within each region. Results from a preliminary implementation are presented.

Introduction

Recognizing a polyhedron from its monocular image using an object-centered representation is a non-trivial, but feasible task [14, 21], because most observable image features are the projections of object features: The image contours are the projections of edges, while their junctions are the projections of vertices, plus the occasional t-junctions that occur when a face partially occludes an edge.

In contrast, most features observable in the image of a curved object are viewpoint-dependent and cannot be traced back to particular object features: The image contours of a smooth object are the projections of limb points, i.e., regular surface points where the viewing direction is tangent to the surface; they join at t-junctions, and may also terminate at cusp points having the additional property that the viewing direction is an asymptotic direction of the tangent plane [18].

*Part of this research was conducted while the authors were with the Robotics Laboratory of Stanford University, where support was provided in part by the Air Force Office of Scientific Research under contract F33615-85-C-5106 and by the Advanced Research Projects Agency of the Department of Defense under Knowledge Based Vision ADS subcontract S1093 S-1.

These simple remarks suggest investigating viewer-centered representations for recognizing curved objects from their monocular image contours. This paper is a first step in this direction, and it addresses the problem of constructing the aspect graph representation [19] of complex curved objects. Combining this qualitative, viewer-centered representation with more quantitative, object-centered representations [25] to obtain efficient control structures of the recognition process is the long-term goal of our research, and we will return to these issues in the conclusion.

Informally, the aspect graph of an object enumerates all possible configurations of its image features as a function of viewpoint. Formally, it is known from singularity theory [1, 17] that image structure is in general stable with respect to viewpoint: perturbing the camera position in a small ball around the original viewpoint will not change the contour topology. However, from some viewpoints, almost any infinitesimal change will alter this topology. It follows that the range of all possible viewpoints can be partitioned into maximal regions where the structure of the contours, or aspect, is stable. The change in the aspect at the boundary between the regions is named a visual event. The maximal regions and their boundaries can be organized into an aspect graph, whose nodes represent the regions with their associated aspects and whose arcs correspond to the visual event boundaries between adjacent regions.

Since their introduction by Koenderink and Van Doorn [19] more than ten years ago, aspect graphs have been the object of very active research. We believe that viewer-centered representations like aspect graphs are crucial for curved objects, but, ironically, most of the past research in this field has focussed on polyhedra, maybe because the contour generators of these objects, being viewpoint-independent, are relatively simple. Indeed, approximate aspect graphs of polyhedra have been successfully used in recognition tasks [10, 13, 15], while several algorithms have recently been proposed for computing the exact aspect graph of these objects (see, for example, [9, 23, 31, 33]).

That the necessary theoretical tools for building exact aspect graphs of curved objects were firmly established results from singularity and catastrophe theories [1, 17] was already recognized by Koenderink and Van

Doorn [19], and has, since then, been reaffirmed by others [4, 27]. However, algorithms implementing these tools have, until very recently, remained elusive. This paper is the second in a series on the construction of exact aspect graphs of piecewise-smooth objects, based on the catalogue of possible visual events established by Rieger [27]. Previously, we had considered solids of revolution whose generating curve is algebraic [20] (see also [6] for a different approach to the same problem). Here, we present an algorithm for computing the aspect graph of solids bounded by rational parametric patches and their intersection curves, observed under orthographic projection. To the best of our knowledge, this is the first algorithm ever proposed for computing the exact aspect graph of a piecewise-smooth, curved object which is not a solid of revolution.

Mathematical details, omitted here for the sake of conciseness, can be found in [26].

Approach

A rational parametric patch has ratios of polynomials as coordinates, i.e., it is defined by:

$$\mathbf{x}(u, v) = \frac{1}{\sum_{i,j} d_{ij} u^i v^j} \sum_{i,j} u^i v^j \mathbf{x}_{ij}, \quad (u, v) \in I \times J, \tag{1}$$

where the d_{ij} 's are scalar coefficients, the \mathbf{x}_{ij} 's are vectors of coefficients, and I, J are intervals of \mathcal{R} . This representation includes nearly all geometric models used in computer aided geometric design and computer vision, such as polyhedra, CSG models, bicubic patches, generalized cylinders, and superquadrics [25]. In addition, Sederberg and his colleagues [11, 28] have shown that elimination theory can be used to compute the polynomial implicit equation of a rational patch and construct an exact representation for the intersection curve of two patches in the form of a polynomial implicit equation $I(u, v) = 0$ in the parameter space of one of the patches.

So, the edges of a solid bounded by rational parametric patches and their intersection curves can be characterized by a polynomial equation. As remarked earlier, the limbs are characterized by the fact that the viewing direction \mathbf{v} is tangent to the surface, i.e., $\mathbf{n}(u, v) \cdot \mathbf{v} = 0$, where \mathbf{n} is the surface normal. Once again, this can be rewritten as a polynomial equation in u and v , and it follows that all image contours can be characterized by implicit polynomial equations. More generally, it will be shown in the next sections that visual events can be represented by systems of polynomial equations, and this will allow us to use global numerical techniques such as continuation [22] to characterize these events. The proposed procedure for constructing an aspect graph is:

- Characterize the curves on the unit sphere which correspond to the visual events of the transparent object.
- Build a graph of the regions delineated by these curves and characterize within each region the corresponding aspect of the transparent object.

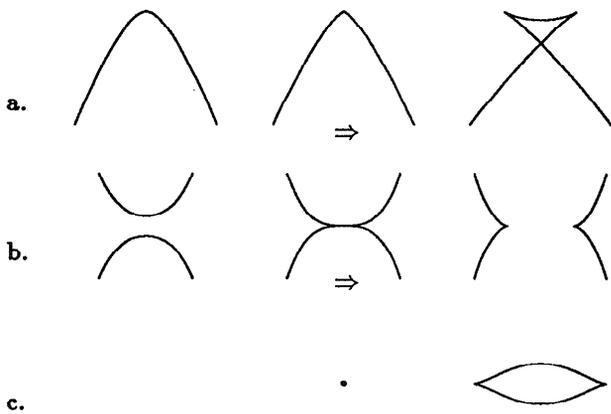


Figure 1: Local singularities of the visual map: (a) swallowtail; (b) beak-to-beak; (c) lip.

- Remove hidden curve branches of the opaque object from the aspects and merge adjacent regions with identical opaque aspects.

As will be shown in a moment, each step of this algorithm corresponds de facto to characterizing the topology of curves defined in high dimension spaces by systems of polynomial equations. This is the key to the success of our algorithm, because it gives us access to the arsenal of computational tools available for characterizing algebraic curves. In principle, this can be done by using symbolic methods such as cylindrical algebraic decomposition [5]. However, it is well known that these algorithms suffer from “intermediate expression swell,” i.e., the time and space required for the computation of intermediate expressions may become so large as to make the desired computations impossible [2]. Instead, we propose using the numerical method of continuation [22] and combining it with a new numerical curve tracing algorithm and ray tracing techniques [16, 35].

A catalogue of visual events

From singularity and catastrophe theories [1], it is known that, from most viewpoints, image contours are piecewise-smooth curves whose only singularities are cusps and t-junctions. They are stable with respect to the viewing direction: perturbing the camera position in a small ball around the original viewpoint will not change their topology. From some viewpoints, however, almost any perturbation of the viewing direction will alter the contour topology.

A catalogue of these “visual events”, or “singularities”, has been established by Kergosien [17] for transparent generic smooth surfaces observed under orthographic projection. It has been extended by Rieger to piecewise-smooth [27] surfaces, so it applies to both edge and limb projections. As shown in [1], each visual event in this catalogue occurs when the viewing direction has a high order contact with the observed surface along certain characteristic curves.¹ When contact occurs at a single point on the surface, the singularity is

¹Recently, Rieger’s catalogue has been extended to

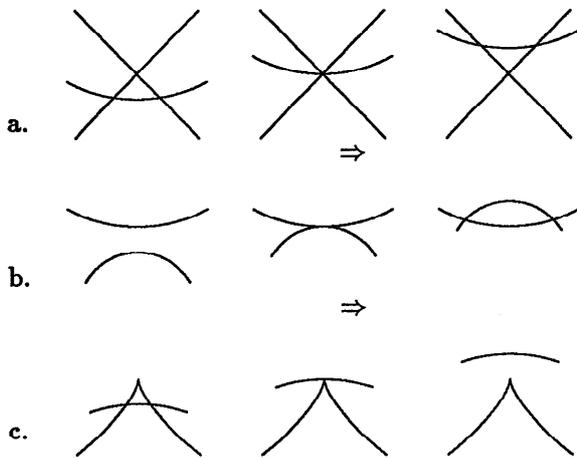


Figure 2: Multilocal singularities of the visual map: (a) triple point; (b) tangent crossing; (c) cusp crossing.

said to be local; when it occurs at different points, it is said to be multilocal.

In the next two sections, we study the characteristic curves associated with local and multilocal events in more detail. For rational parametric patches, it is shown that both types of events are characterized by systems of n polynomial equations in $n + 1$ unknowns.

Local singularities

Let us first consider edge projections. Their local visual events are cusps [27, 32], where the viewing direction is tangent to the edge. In this case, the characteristic curve is the edge itself, and the viewing direction corresponding to each edge point is its tangent [26].

From catastrophe theory, it is known that occluding contours may exhibit three types of local singularities: swallowtails, beak-to-beak, and lip transitions [17], as shown in figures 1,5,6,7. During a swallowtail transition, a smooth image contour forms a singularity and then breaks off into two cusps and a t-junction. In a beak-to-beak transition, two distinct portions of the occluding contour meet at a point in the image; after meeting, the contour splits and forms two cusps; the interconnectivity of the limbs changes. Finally, a lip transition occurs when, out of nowhere, a closed curve is formed with the introduction of two cusps.

As shown by Arnol'd [1], swallowtails occur on flecnodal curves, while both beak-to-beak and lip transitions occur on parabolic curves; in all cases, the corresponding viewing directions are given by asymptotic directions along these curves. Flecnodal points are inflections of asymptotic curves, while parabolic points are zeros of the Gaussian curvature [34]. Implicit equations for both types of curves are given in [26]; the corresponding viewing directions are computed by solving the asymptotic direction equation [34].

some non-generic surfaces, namely developable surfaces, by Sripradisvarakul and Jain [30]. Further research is needed to ascertain if new characteristic curves are necessary to characterize the additional visual events of these surfaces.

Multilocal singularities

We have just seen that local singularities can be characterized by one equation in two unknowns which is the implicit equation of a surface curve. Let us now turn to multilocal singularities, which occur when two or more surface points project onto the same contour point. As shown by Kergosien [17] and Rieger [27], there are three types of multilocal visual events (figure 2). Three contour segments can intersect at a triple point. For an opaque object, only two branches are seen on one side of the transition while three branches are visible on the other side. Secondly, a tangent crossing occurs when two contours meet at a point and share a common tangent. Finally, a cusp crossing occurs when the projection of an occluding contour cusp meets another contour.

Multilocal events are not characterized by a single surface curve, but instead by curves in high dimension spaces, or equivalently by families of surface curves. For example, a triple point is formed when three surface points are aligned and, in addition, the surface normals at the three points are all orthogonal to the common line supporting these points. By sweeping this line while maintaining three-point contact, a family of three curves is drawn on the surface.

The equations for the multilocal singularities are derived in [26]. The most complex singularity is the triple point, defined by five equations in six unknowns. Tangent crossings and cusp crossings are each characterized by three equations in four unknowns. The viewing direction corresponding to a multilocal event is given by $\mathbf{x}_2 - \mathbf{x}_1$, where \mathbf{x}_1 and \mathbf{x}_2 are two of the points forming the singularity.

Tracing the visual events

We saw in the previous section that any visual event of a solid bounded by smooth surface patches and their intersection curves is a curve Γ defined in \mathbb{R}^{n+1} by a system of n equations in $n + 1$ variables:

$$\begin{cases} P_1(x_0, x_1, \dots, x_n) = 0 \\ \dots \\ P_n(x_0, x_1, \dots, x_n) = 0 \end{cases} \quad (2)$$

As shown in [26], these equations are polynomials in the partial derivatives of the considered patches. By clearing if necessary the appropriate denominators, it is obvious in the case of rational parametric patches that each of the P_i 's is a polynomial in the patches' parametric coordinates.

The curve Γ can therefore be defined implicitly by a system of algebraic equations. To characterize its topology, it is possible, in principle, to use the symbolic method of cylindrical algebraic decomposition [5]. This may, however, prove impossible in practice [2]. Alternatively, the curve can be traced numerically, using a prediction-correction scheme [7]. The main difficulties are finding a sample point on each real component of the curve to initiate the marching and dealing with the curve singularities where the tangent is not defined [3].

We propose a new algorithm to overcome these difficulties. It is divided into four steps: (1) first compute

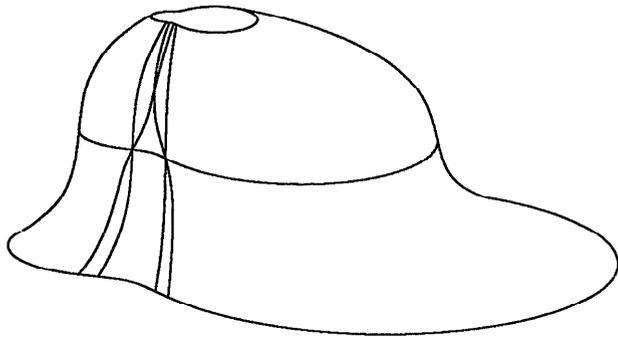


Figure 3: The parabolic and flecnodal curves on the surface of an SHGC.

all extremal points in some direction, say x_0 ; this includes all singular points; (2) compute all intersections of the curve with the hyperplanes orthogonal to the x_0 axis at the extremal points; (3) for each interval of the x_0 axis delimited by these hyperplanes, intersect the curve and the hyperplane passing through the mid-point of the interval; (4) march numerically from these intersections to the adjacent hyperplanes.

As shown in [26], steps (1) to (3) involve solving systems of polynomial equations. This is done by using the global numerical method of continuation [22] for solving systems of n polynomial equations in n variables. Continuation is itself a form of curve tracing, and it can find all solutions (counting their multiplicities) of systems having up to a few thousand roots [22].

Step (4) is similar to the marching step in [3, 7], and it only involves the inversion of a linear system: Since the mid-point of each interval corresponds to branches of the curve where the tangent's component along the x_0 axis is never zero, a Taylor expansion of the curve in the x_0 direction can be used to guide the marching. This step ends when the bounds of the interval are reached. At this time, the current branch is connected to the extremal points found earlier, producing a graph representation of the curve in terms of smooth branches joined at singularities (an s-graph in Arnon's terminology [2]).

Note that, for each interval, a sample point is obtained for each branch of the curve within this interval. In addition, marching through singularities is trivially avoided by marching only within intervals where the curve is extrema-free and therefore not singular. The algorithm is detailed in [26].

The remaining steps of the algorithm

We have shown how to trace the curves which correspond to the visual events on the object's surface. At each point, it is possible to compute the corresponding viewing direction and therefore to trace the corresponding visual event on the viewing sphere. We now sketch the remaining steps of the aspect graph construction: building a graph of stable view regions, constructing the corresponding transparent aspects, and merging adjacent identical opaque aspects.

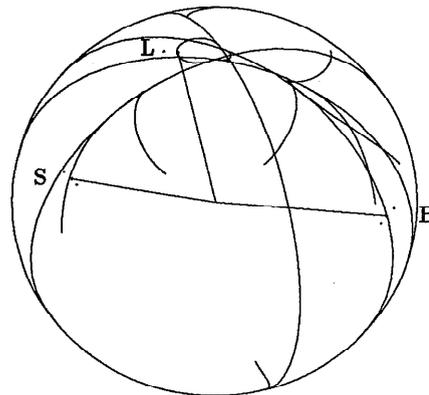


Figure 4: The curves on the unit sphere that correspond to the local events of an SHGC. Three sample directions (marked 'S', 'B', and 'L') are used to characterize the local events in the next three figures.

To compute the graph of regions, we need to find all intersections of the visual event curves on the unit sphere. This can be done, once again, by solving a set of polynomial equations. Then, the intersections and extremal points can be sorted, as before, along some direction, and sample points for each smooth branch can be constructed. A sample point for the region delimited by two consecutive branches on one interval is simply obtained as the mid-point of the branches' sample points. The output of this step is a set of regions, curves, and intersection points, plus their connectivity information.

In addition, the sample point associated with each region is a representative viewing direction for this region. The contours associated with this view can once again be computed by curve tracing since they are the projections of edges, defined by $I(u, v) = 0$, and limbs, defined by $\mathbf{n} \cdot \mathbf{v} = 0$. It follows that each aspect can be represented by an s-graph of smooth curve branches separated by the contour singularities, i.e., cusps and t-junctions.

It is known that, for a stable view, the visibility of contours only changes at cusps and t-junctions [18]. It follows that the visibility of contours does not change along a given branch of the s-graph, and it can be determined by ray tracing [35] at the sample point of each branch. As shown in [16], ray tracing rational parametric patches amounts to solving a system of polynomials, which can be done by using continuation. Once the visibility of the contour branches has been determined, it is a simple matter to eliminate the invisible branches and merge identical adjacent aspects.

Implementation and results

The curve tracing algorithm has been implemented and used, among other things, to compute the intersection curves and the silhouette of algebraic surfaces. As a first step in the implementation of the aspect graph algorithm, local visual events of a straight homogeneous generalized cylinder (SHGC, see [24, 29]) have been computed. SHGC's are obtained by scaling a (not necessarily circular) reference cross-section along

a straight axis. They are a convenient choice for experimentation since they can be represented by rational parametric patches [25], and proven methods are available for computing and drawing their contours [24]. In the examples presented in this section, we have used an SHGC whose scaling function is a cubic curve and whose reference cross-section is a complex non-convex curve.

Figure 3 shows a line-drawing of this object and the corresponding parabolic lines and flecnodal curves. This line-drawing has been computed by the methods described in [24]. The flecnodal and parabolic curves have been computed by the curve tracing algorithm described earlier. Figure 4 shows the corresponding curves on the unit sphere. Note that some of these curves are not closed, and therefore do not delimit regions on the unit sphere. This should be expected since the SHGC's surface has a finite extent and the multilocal events have not been traced.

Finally, figures 5, 6, and 7 show examples of swallowtail, beak-to-beak, and lip transitions in the neighborhood of sample directions lying on visual event curves of the unit sphere. These directions and the adjacent stable views are shown as arrows and dots, and marked 'S', 'B', and 'L' in figure 4. In order to show the contour structure (e.g., cusps and t-junctions), the hidden lines have not been removed in these three figures.

Discussion and future research

We have presented a new algorithm for computing the exact aspect graph of curved objects and demonstrated a preliminary implementation. This algorithm is quite general since, as noted before, rational patches and their intersection curves subsume most representations used in computer aided geometric design and computer vision. Unlike possible alternative approaches based on cylindrical algebraic decomposition [2, 5], our algorithm is also practical, combining well-established numerical techniques such as continuation [22] and ray tracing [16, 35] with a new curve tracing technique which respects curve singularities.

Our immediate goal is to complete the implementation of the aspect graph algorithm. As noted in the introduction, future research will be dedicated to actually using the aspect graph representation in recognition tasks. In [25], we have demonstrated the recovery of the position and orientation of curved three-dimensional objects from monocular contours by using a purely quantitative process that fits an object-centered representation to image contours. What is missing is a control structure for guiding this process. We believe that the qualitative, viewer-centered aspect graph representation can be used to guide the search for matching image and model features and yield efficient control structures analogous to the interpretation trees used in the polyhedral world [8, 12, 14].

Acknowledgements: We thank Kevin Bowyer, John F. Canny, Martial Hébert, Seth Hutchinson, and Brigitte Ponce for useful discussions and comments.

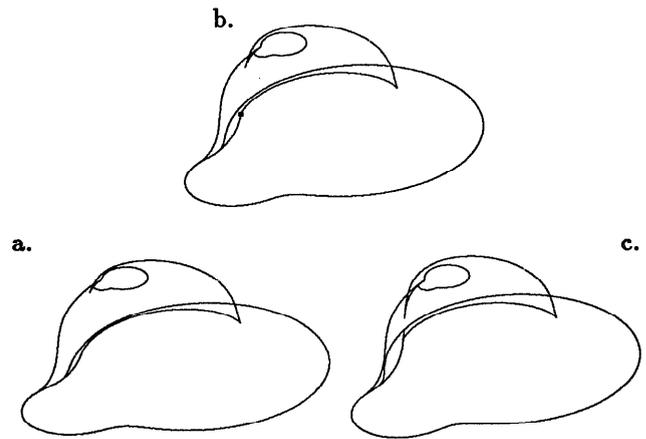


Figure 5: A swallowtail transition: A smooth contour (a) acquires a discontinuity (b) which splits into one t-junction and two cusps (c). This local event corresponds to the viewing direction marked 'S' in figure 4.

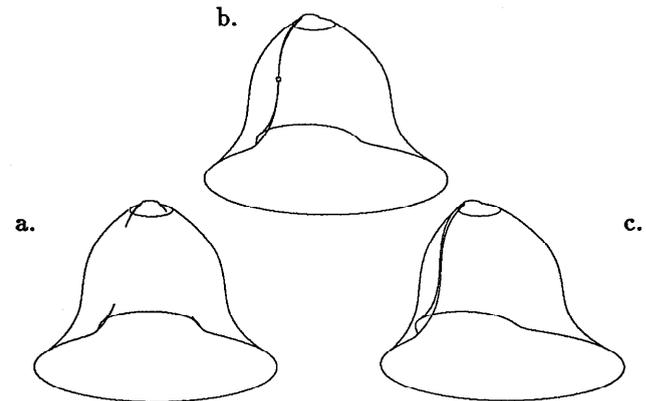


Figure 6: A beak-to-beak transition: Two disjoint pieces of contour (a) join at the transition (b), then split again in the other direction (c). This local event corresponds to the viewing direction marked 'B' in figure 4.

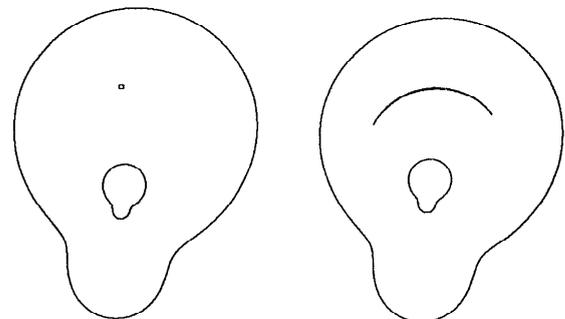


Figure 7: A lip transition: A contour appears out of nowhere. This local event corresponds to the viewing direction marked 'L' in figure 4.

References

- [1] V.I. Arnol'd. Singularities of systems of rays. *Russian Math. Surveys*, 38(2):87-176, 1983.
- [2] D.S. Arnon. Topologically reliable display of algebraic curves. *Computer Graphics*, 17(3):219-227, July 1983.
- [3] C.L. Bajaj, C.M. Hoffmann, R.E. Lynch, and J.E.H. Hopcroft. Tracing surface intersections. *Computer Aided Geometric Design*, 5:285-307, 1988.
- [4] J. Callahan and R. Weiss. A model for describing surface shape. In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, pages 240-245, 1985.
- [5] G.E. Collins. *Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition*, volume 33 of *Lecture Notes in Computer Science*. Springer-Verlag, New York, 1975.
- [6] D. Eggert and K. Bowyer. Computing the orthographic projection aspect graph of solids of revolution. In *IEEE workshop on interpretation of 3D scenes*, pages 102-108, Austin, TX, November 1989.
- [7] R.T. Farouki. Trimmed-surface algorithms for evaluation and interrogation of solid boundary representations. *IBM Journal of Research Development*, 31(3):314-332, may 1987.
- [8] O.D. Faugeras and M. Hébert. The representation, recognition, and locating of 3-D objects. *International Journal of Robotics Research*, 5(3):27-52, Fall 1986.
- [9] Z. Gigus and J. Malik. Computing the aspect graph for line drawings of polyhedral objects. *IEEE Trans. Pattern Anal. Machine Intell.*, 12(2):113-122, February 1990.
- [10] C. Goad. Special purpose automatic programming for 3D model-based vision. In *Proc. Image Understanding Workshop*, pages 94-104, June 1983.
- [11] R.N. Goldman and T.W. Sederberg. Some applications of resultants to problems in computational geometry. *The Visual Computer*, 1:101-107, 1985.
- [12] W.E.L. Grimson and T. Lozano-Pérez. Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. Pattern Anal. Machine Intell.*, 9(4):469-482, 1987.
- [13] M. Hébert and T. Kanade. The 3D profile method for object recognition. In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, San Francisco, CA., June 1985.
- [14] D.P. Huttenlocher and S. Ullman. Object recognition using alignment. In *International Conference on Computer Vision*, pages 102-111, London, U.K., June 1987.
- [15] K. Ikeuchi and T. Kanade. Automatic generation of object recognition programs. *Proceedings of the IEEE*, 76(8):1016-35, August 1988.
- [16] J.T. Kajiya. Ray tracing parametric patches. In *Proc. of SIGGRAPH 82*, pages 245-254, July 1982.
- [17] Y.L. Kergosien. La famille des projections orthogonales d'une surface et ses singularités. *C.R. Acad. Sc. Paris*, 292:929-932, 1981.
- [18] J.J. Koenderink. What does the occluding contour tell us about solid shape? *Perception*, 13, 1984.
- [19] J.J. Koenderink and A.J. Van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211-216, 1979.
- [20] D.J. Kriegman and J. Ponce. Computing exact aspect graphs of curved objects: solids of revolution. In *IEEE workshop on interpretation of 3D scenes*, pages 116-122, Austin, TX, November 1989.
- [21] D. Lowe. The viewpoint consistency constraint. *International Journal of Computer Vision*, 1(1):57-72, 1987.
- [22] A.P. Morgan. *Solving Polynomial Systems using Continuation for Engineering and Scientific Problems*. Prentice Hall, Englewood Cliffs, 1987.
- [23] H. Plantinga and C. Dyer. Visibility, occlusion, and the aspect graph. CS Tech. Report 736, University of Wisconsin, 1987.
- [24] J. Ponce and D. Chelberg. Finding the limbs and cusps of generalized cylinders. *International Journal of Computer Vision*, 1(3), October 1987.
- [25] J. Ponce and D.J. Kriegman. On recognizing and positioning curved 3D objects from image contours. In *IEEE Workshop on the Interpretation of 3D scenes*, pages 61-67, Austin, TX, November 1989.
- [26] J. Ponce and D.J. Kriegman. Computing exact aspect graphs of curved objects: parametric patches. Technical Report UIUCDCS-R-90-1579, University of Illinois at Urbana-Champaign, March 1990.
- [27] J.H. Rieger. On the classification of views of piecewise smooth objects. *Image and Vision Computing*, 1:91-97, 1987.
- [28] T.W. Sederberg, D.C Anderson, and R.N. Goldman. Implicit representation of parametric curves and surfaces. *Comp. Vision, Graphics, and Image Proc.*, 28:72-84, 1984.
- [29] S.A. Shafer. *Shadows and Silhouettes in Computer Vision*. Kluwer Academic Publishers, 1985.
- [30] T. Sripradisvarakul and R. Jain. Generating aspect graphs of curved objects. In *IEEE Workshop on Interpretation of 3D Scenes*, pages 109-115, Austin, Tx., December 1989.
- [31] J. Stewman and K.W. Bowyer. Creating the perspective projection aspect graph of polyhedral objects. In *International Conference on Computer Vision*, 1988.
- [32] C.T.C. Wall. Geometric properties of generic differentiable manifolds. In A. Dold and B. Eckmann, editors, *Geometry and Topology*, pages 707-774, Rio de Janeiro, 1976. Springer-Verlag.
- [33] N. Watts. Calculating the principal views of a polyhedron. Technical Report CS Tech. Report 234, Rochester University, 1987.
- [34] C.E. Weatherburn. *Differential geometry*. Cambridge University Press, 1927.
- [35] T. Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 23(6):343-349, June 1980.