ASPECT GRAPHS: STATE-OF-THE-ART AND APPLICATIONS IN DIGITAL PHOTOGRAMMETRY*

David W. Eggert and Kevin W. Bowyer
Department of Computer Science and Engineering
University of South Florida
Tampa, Florida 33620 USA
eggertd or kwb@csee.usf.edu
ISPRS Commission V

Charles R. Dyer
Department of Computer Science
University of Wisconsin
Madison, Wisconsin 53706 USA
dyer@cs.wisc.edu
ISPRS Commission V

The study of the viewer–centered object representation known as the *aspect graph* has recently been an active area of research in computer vision. The aspect graph is desirable because it provides a complete enumeration of all possible distinct views of an object, given a particular model for viewpoint space and a definition of "distinct". This paper presents a history of the evolution of the aspect graph, culminating with the current state of the art in algorithms and implementations for automatically constructing an aspect graph. The use of the aspect graph in possible applications in computer vision and computer graphics is described. Finally, current limitations of the representation are discussed and a potential solution involving the scale space concept is presented.

Key Words: Computer Vision, Computer Graphics, Viewer-centered Representation, Aspect Graph, Survey, Scale Space.

1. INTRODUCTION

The origin of the aspect graph concept** has several independent roots. It is most often credited to Koenderink and van Doorn (Koenderink and Van Doorn, 1976, 1979) who initially referred to it as the *visual potential* of an object. Somewhat earlier, Minsky described a concept very similar to the aspect graph (Minsky, 1975), sketching an example in terms of a *frame system* that depicted the different visual possibilities for a cube. Somewhat later, Chakravarty and Freeman (Chakravarty and Freeman, 1982) employed a similar concept, under the term *characteristic views*, in a study involving recognition of polyhedra. Since then, several other viewer–centered representations similar to the aspect graph have also been proposed.

Each of these authors recognized the potential value of a representation that summarizes all of the possible distinct views of an object. Also, researchers in the fields of computer vision (Rosenfeld, 1987) and psychophysics (Palmer, Rosch and Chase, 1981; Perrett, *et al.*, 1989) have been gathering evidence that humans may use a set of "important" aspects to achieve fast recognition of unknown objects, although it is unclear whether a human's definition of "aspect" and "important" coincide with what will be described here.

Unfortunately, none of the first researchers was able to describe an algorithm to automatically compute such a representation for any specific class of objects. As stated by Koenderink and van Doorn, "A general decomposition of $E_3 - B$ (B refers to the space occupied by a solid object) into cells that provide a stable global aspect of δB is by no means trivial to carry out." (Koenderink and van Doorn, 1976, page 57). This simple fact delayed research on aspect graphs for several years. However, now due to intensive research in recent years there exist a number of different algorithms and even implementations to produce this representation.

The remainder of this paper is organized as follows. Section 2 presents a more rigorous and detailed definition of the aspect graph. Section 3 describes the approach used in computing an approximate aspect graph. Sections 4 and 5 outline the considerations involved and subsequent results in computing the exact aspect graph of polyhedral objects and curved objects, respectively. Section 6 discusses a generalization of the aspect graph concept for objects having articulated connections between rigid parts. Section 7 describes some of the possible applications for aspect graphs in both computer vision and computer graphics. In Section 8 some possible deficiencies in the current conception of the aspect graph representation are mentioned, and a potential solution, the *scale space aspect graph*, is presented. Finally, Section 9 briefly presents some topics of continuing research.

2. DEFINING THE ASPECT GRAPH

The commonly agreed upon elements of the definition of an aspect graph representation are generally that:

- there is a node for each *general view* of the object as seen from some maximal connected cell of *viewpoint space*, and
- there is an arc for each possible transition, called a *visual event*, between two neighboring general views.

The not so commonly agreed upon elements, needed to complete this definition, are the model of viewpoint space, and what is meant by a *general view*. These and other factors discussed in the next sections can be used to classify the various algorithms developed to date, as shown in Figure 1.

* This work was supported at the University of South Florida by Air Force Office of Scientific Research grant AFOSR-89-0036, National Science Foundation grant IRI-8817776 and a grant from the Florida High Technology and Industry Council, and at the University of Wisconsin by National Science Foundation grant IRI-8802436 and the University of Wisconsin Graduate School.

** In this paper, the term *aspect graph* will refer generally to representations that have also been called *characteristic views, principal views, stable views, viewing data, view classes* and other similar terms.

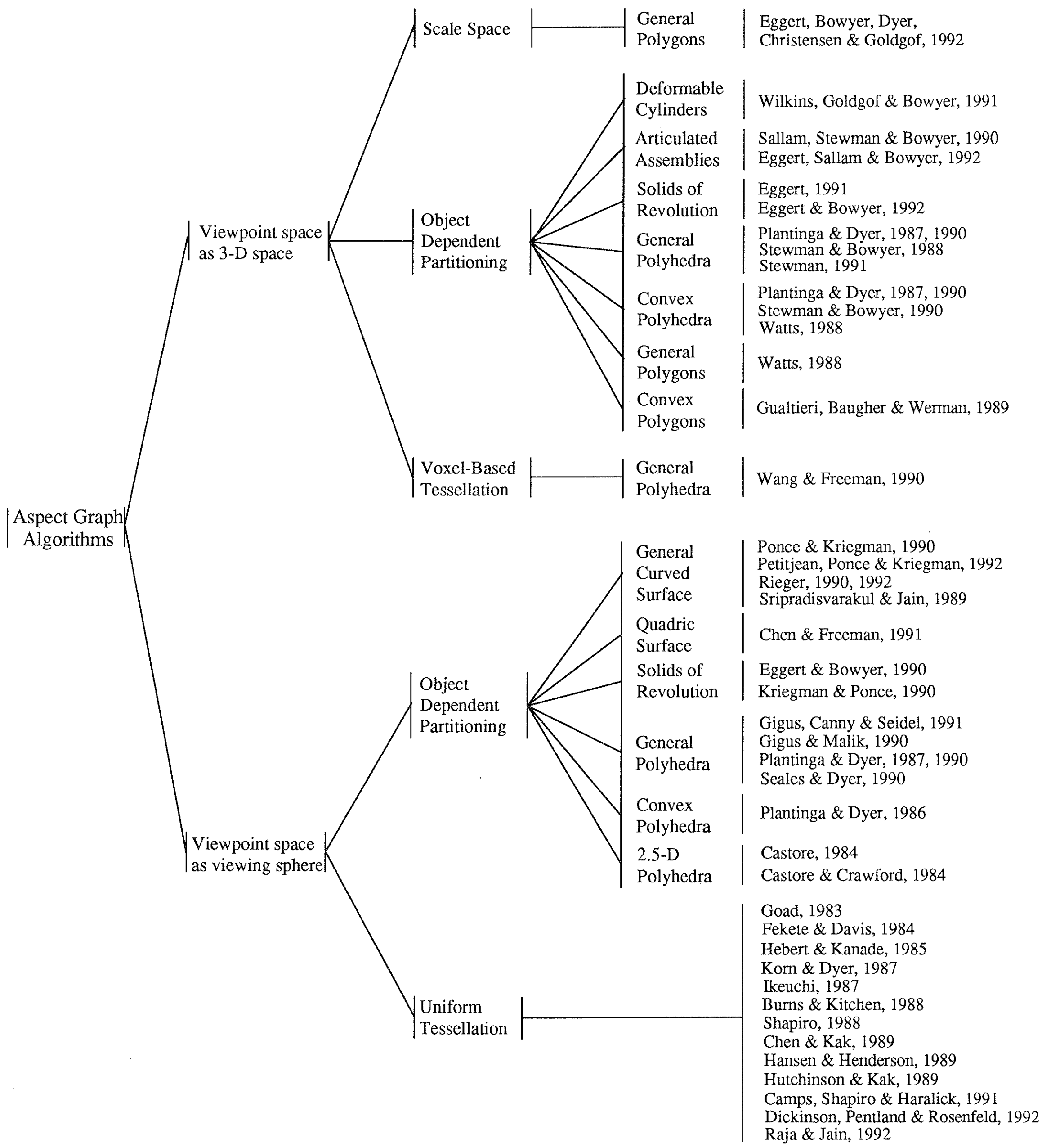


Figure 1. Classification of algorithms for aspect graph creation. Algorithms are categorized by model of viewpoint space, method of creation and class of object shape allowed. Cited references are in the bibliography.

2.1 Models of Viewpoint Space

The model of viewpoint space has perhaps had the greatest effect on the various algorithms. Two basic models of viewpoint space are commonly used. One is the *viewing sphere*. In this model, the space of possible viewpoints is the surface of a unit sphere, defining a 2–D parameter space. The sphere is considered to be centered around a model of the object, which is located at the origin of the coordinate system. A viewpoint on the surface of the sphere defines a line of sight vector from the viewpoint toward the origin. This direction vector is usually used to create an orthographic projection view of the object. It is possible to use perspective projection with the viewing sphere model, but this requires an assumption of a known viewer–to–object distance.

A more general model of viewpoint space is to consider all positions in 3–D space as possible viewpoints. As in the case of the viewing sphere, the object can be considered to be located at the origin of the coordinate system. Specifying a direction vector for the line of sight and a focal length for the imaging process allows the creation of a perspective projection view of the object. This normally would require potentially a 7–D parameter space to describe the viewing process. However, a simplifying assumption is made such that an aspect is concerned with all potential features seen from a given viewpoint, if the line of sight is directed appropriately. Thus only three parameters are necessary to specify the viewpoint position, and the viewer will possess a 360° field of view in all directions. Later, during the pose estimation portion of the object recognition task, the exact

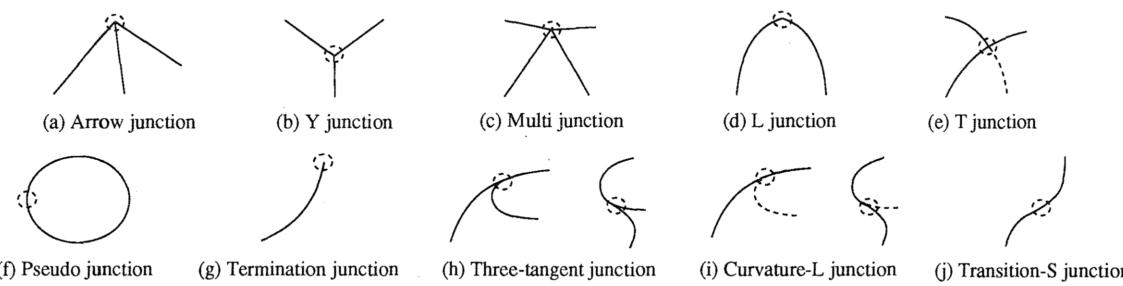


Figure 2. Junction types in image structure graph based on catalogs in (Malik, 1987; Eggert, 1991). (a) Arrow junction has three arcs with distinct tangents, one angle between a pair of arcs is greater than pi, (b) Y junction has three arcs with distinct tangents, all angles between pairs of arcs are less than pi, (c) Multi junction has four or more arcs meeting at point, (d) L junction has two arcs with distinct tangent, (e) T junction marks occlusion boundary, only two arcs have common tangent and curvature, (f) Pseudo junction is not a real junction, used to include closed arc in graph, (g) Termination junction marks end of a visible contour (cusp), (h) Three-tangent junction has three arcs with a common tangent, only two have common curvature, (i) Curvature-L junction has two arcs with a common tangent, but have distinct curvatures, (j) Transition-S junction has continuous tangent and curvature across junction, curvature changes sign where two arcs meet.

lines of sight can be adjusted to produce the projected feature locations that best match the observed image features.

2.2 Defining a General View

The particular set of features used to describe a view has varied between researchers. Early work used various 2–D feature types such as image edges, vertices, silhouette moments, and holes. The 3–D features taken from range imagery also included edges and vertices, as well as face area, orientation and location. Recently, the most common form of symbolic representation for a 2–D view of a 3–D scene has become the *image structure graph.* There is a long tradition of using the image structure graph in the analysis of polyhedral scenes, and the concept has also been extended to handle curved objects (Malik, 1987). The image structure graph is a qualitative specification of the type and configuration of the elements in the line drawing, omitting any quantitative data such as the lengths of lines or the size of the angles between lines.

The image structure graph is formed by abstracting the original intensity image into an idealized *line drawing,* and then labeling the lines and junctions formed by these lines. For polyhedral objects, the line drawing is entirely determined by the *lines* and the *junctions* formed where two or more meet. To be clear about the terminology, a *line* in the line drawing is a 2–D projection in the image plane of a 3–D *edge* on the object surface. An *edge* is a locus of points where there is a discontinuity in the surface normal (for example, where two faces of a polyhedron meet).

For curved objects, the situation is more complicated. In addition to edges that project to lines, curved objects also have *contour generators* that project to *occluding contours,* or simply *contours,* in the line drawing. A *contour generator* (also called the *rim* or *limb*) is a locus of points on the object surface where the line of sight is tangent to the surface (for instance, the apparent side of a cylinder). For curved objects, both lines and contours may appear in the image as either straight or curved.

The lines/contours and junctions in a line drawing may be assigned symbolic labels that indicate their 3–D interpretation. For example, a line in a polyhedral scene may be given the symbolic label *convex* to indicate that the internal angle between the two faces meeting at the corresponding edge on the object is less than 180°, or *concave* for angles greater than

2.3 Object Domains

Over the years, the domain of objects for which an aspect graph algorithm exists has steadily expanded. The viewing sphere, being simpler, has allowed more analyses to be carried out both in theory and in practice. The more complicated model of 3–D space has not limited theoretical developments as much as it has implementation. Polygons and polyhedra are now well understood, and programs exist which will construct their aspect graphs under either model. Curved shapes, and more complex shapes such as those that are articulated or deformable, require sophisticated construction techniques. Only a few such algorithms have been implemented and those are restricted almost entirely to the viewing sphere approach. These different domains will be discussed later in the paper. An unlimited domain of objects is available if one is willing to construct only an approximate aspect graph, the topic of the next section.

180°. Similarly, junctions may be given labels to indicate the quantity and type of contours intersecting at the point. A fairly complete catalog of junction types, based on that of Malik (Malik, 1987), is shown in Figure 2. For polyhedra, only the first few junctions can exist, while all may occur for a curved surface object. Even the labeling of the image structure graph is not entirely consistent among researchers, as some may choose to omit certain junction types from consideration, or let such things as line labels be implicit rather than explicit. Other possibly more robust image representations will be discussed in a later section.

Based on the image structure graph, viewpoint space can be partitioned into regions of *general viewpoints* separated by boundaries of *accidental viewpoints.* Under either model of viewpoint space, a general viewpoint is one from which an infinitesimal movement can be made in any direction and the resulting image structure graph is isomorphic to the original. An accidental viewpoint is one for which there is some direction in which an infinitesimal movement will result in an image structure graph that is not isomorphic to the original. A *visual event* is said to occur on the passing from one region of general viewpoints through a boundary of accidental viewpoints into another region of general viewpoints.

3. APPROXIMATE ASPECT GRAPHS

There are a variety of difficulties involved in developing an algorithm to compute the aspect graph for a class of objects.

It is not a trivial problem to determine all of the fundamental visual events that may possibly occur for any object in the class, and to derive the partition of viewpoint space from the particular set of visual events that occur for a given object. The intricacies are such that most implementations have only recently been completed. A simpler, and perhaps more commonly applied approach is not to compute an exact aspect graph at all, but rather an approximation based on a quasi-uniform sampling of viewpoint space.

For the viewing sphere model, the approximation approach typically begins with an *icosahedron* centered around the origin of the coordinate system, so that the vertices of the icosahedron lie on the surface of the sphere. An icosahedron is the *regular* solid having the greatest number of faces, there being 20 congruent equilateral triangles.

By treating the center point on each face of the icosahedron as defining a viewing direction, the icosahedron provides a uniform sampling of 20 points on the sphere. Since 20 viewing directions is generally not sufficient to capture the full visual complexity of an object, each face of the icosahedron is typically subdivided some number of times to provide a finer, quasi-uniform sampling of the viewing sphere. The subdivision is done by connecting the midpoints of the three edges of the current triangular face to one another to subdivide it into four triangles. The three new vertices are "pushed out" to the surface of the sphere, and the center of each new triangle defines a new viewing direction. The eventual size of the tessellation of the sphere varies, with some researchers using more than 5000 faces.

For each sample point on the sphere, a set of visible object features is calculated. Neighboring triangles having the same feature description are merged together to represent one node in the aspect graph. The set of features found for these views is often different from an image structure graph, since they need not be well suited for exact visual event boundary analysis. The above approach has been employed by a large number of researchers (Burns and Kitchen, 1987; Camps, Shapiro and Haralick, 1991; Shapiro, 1988; Chen and Kak, 1989; Dickinson, Pentland and Rosenfeld, 1992; Fekete and Davis, 1984; Goad, 1983; Hansen and Henderson, 1989; Hebert and Kanade, 1985; Ikeuchi, 1987; Hutchinson and Kak, 1989; Korn and Dyer, 1987; Raja and Jain, 1992) and is now a standard technique in computer vision. A similar technique has been applied to 3-D viewpoint space (Wang and Freeman, 1990).

A main advantage of the technique is that it is relatively easy to apply to a rigid object of any shape. (It can also be useful for investigating the behavior of nonrigid objects, as will be discussed later.) One disadvantage is that it is difficult to know *a priori* what the appropriate resolution for the viewing sphere should be for an arbitrary object. If the resolution is not fine enough, some views of the object may not be included in the representation. If it is too fine, unnecessary work will be done in creating the representation.

4. EXACT ASPECT GRAPHS FOR POLYHEDRA

Initial research on the automatic construction of exact aspect graphs began with the simplest of objects, polygons (Gualtieri, Baugher and Werman, 1989; Watts, 1988). This advanced to 2.5-D polyhedra (Castore, 1984; Castore and Crawford, 1984), which are formed by sweeping a polygon along a straight axis. Next came 3-D convex polyhedra (Plantinga and Dyer, 1986, 1987, 1990; Watts, 1988; Stewman and Bowyer, 1990) and eventually general polyhedra were analyzed by several groups of researchers (Gigus and Malik, 1990; Gigus, Canny and Seidel, 1991; Plantinga and Dyer, 1987, 1990; Seales and Dyer, 1990; Stewman, 1991; Stewman and Bowyer, 1988).

The general algorithm for automatically computing the aspect graph of an object has the following steps:

1. Each class of objects has a fundamental set of ways in which accidental viewpoints may occur for any object in the class, and a catalog of these can be developed. From this catalog, specific events for any particular object must be enumerated in some manner.
2. Each visual event represents a hyperplane in viewpoint space that is a boundary between general views. The partition of viewpoint space into cells of general viewpoint is computed from the set of hyperplanes representing the visual events.
3. Finally, the aspect graph itself is obtained by traversing the partition of viewpoint space. This may require merging certain regions of viewpoint space into one, depending on the algorithm used in step 2. Representative views of each aspect are usually calculated for a central point in the cell.

4.1 Visual Events

For polygons, visual events are marked by the alignment of two vertices, causing an edge to be either partially or totally occluded from view. The event boundary is merely the line joining the two aligning vertices. Figure 3 shows the subdivision of the plane into areas where different aspects are seen for a C-shaped polygon. Notice that the event lines are not meaningful everywhere. For general polyhedra, the two fundamental types of accidental views are edge/vertex alignments, and edge triplet alignments. An edge/vertex alignment occurs when a vertex and some point on an edge of the object project to the same point in the image. In general, this event represents the beginning or ending of occlusion,

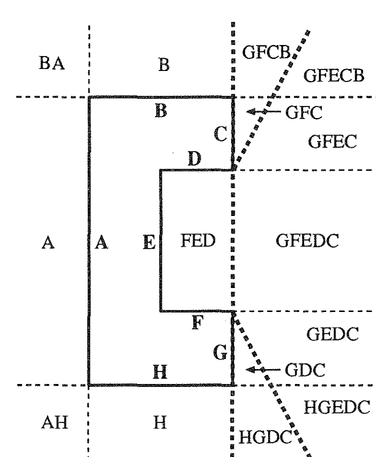


Figure 3. Parcellation of viewpoint space for C-shaped polygon. Normal dashed lines mark visibility limits of polygon edges. Bold dashed lines mark occlusion limits for two edges. Regions (aspects) are labeled with visible edges.

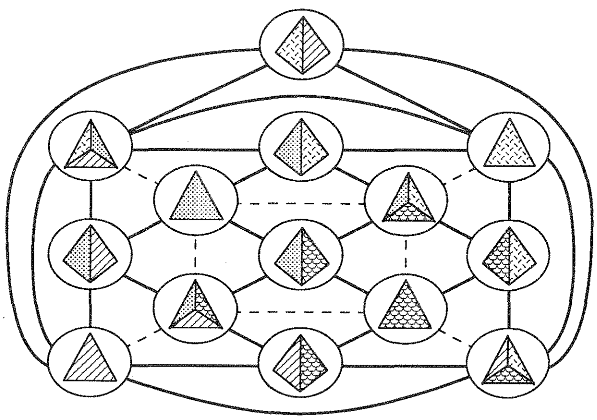


Figure 4. The aspect graph of a tetrahedron. For this object the overall structure is the same for either viewpoint space model. Bold arcs follow the visual event convention of Koenderink and van Doorn (1979), while the additional dotted arcs follow the convention of Plantinga and Dyer (1986).

marked by a T–junction in the image. (It also includes, as a special case, the projection of a face of the object to a line segment in the image.) The viewpoints from which such an accidental alignment can be seen lie on a plane in the 3–D model of viewpoint space, or on a great circle of the viewing sphere. Several of these events can be seen in Figures 4 and 5, which show the aspect graphs of a tetrahedron and a triangular prism with a hole through it, under both viewing models, respectively.

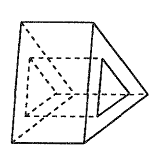
An edge triplet alignment occurs when points on three different edges of the object project to the same point in the image (see Figure 6). In general, this event represents a change in the ordering of the visible T–junction relations between the three edges. The viewpoints from which such an accidental alignment can be seen lie on a quadric surface in the 3–D model of viewpoint space, or on a quadric curve on the surface of the viewing sphere (Gigus and Malik, 1987; Stewman and Bowyer, 1988).

4.2 Parcellating Viewpoint Space

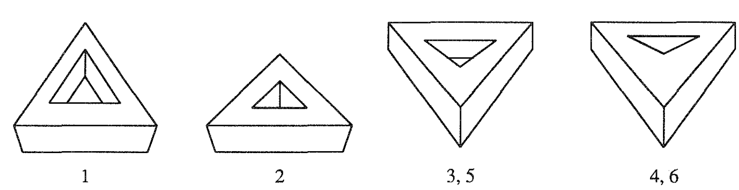
The set of visual event boundaries is found by considering all pairs of edges and vertices, and all triplets of edges. Many of these combinations may be discarded by performing a local visibility test to see if a portion of the object lies between the interacting features, therefore blocking the alignment. For the surviving events, the corresponding surfaces must be used to subdivide viewpoint space.

Under the assumption of orthographic projection, the great circles and quadric curves on the sphere surface can be decomposed into portions which are single–valued with respect to a particular 2–D parameterization of the sphere. This set of curves is then qualitatively equivalent to a set of lines in a plane, and the well known *plane sweep* algorithm can be used to determine curve intersections and subdivide viewpoint space (Gigus and Malik, 1990; Gigus, Canny and Seidel, 1991; Watts, 1988).

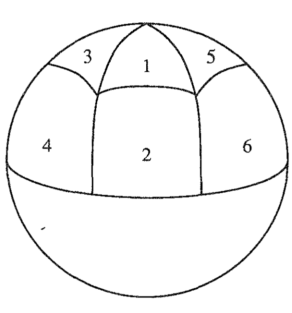
Assuming perspective projection, another method is to construct what is known as a *geometric incidence lattice* to represent the parcellation of 3–D space. This lattice structure defines the various volumes of space, their bounding surface patches, the curves of intersection between the patches and finally the points of intersection between the curves. The calculation of the curves of intersection between the planes and quadrics in 3–D space is sufficiently well known that this structure is constructible for polyhedra (Stewman, 1991; Stewman and Bowyer, 1988).



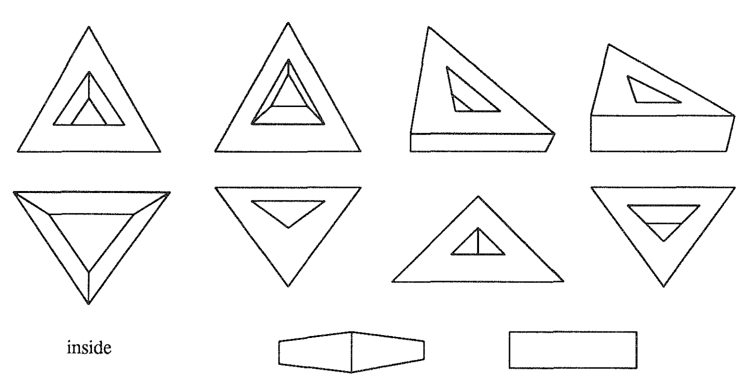
(a) Example nonconvex polyhedron having equilateral triangular faces with through hole of same shape.



(c) Representative views of regions on viewing sphere in (b). Note, only four different views exist even though there are 24 total regions on the sphere.



(b) Partition of upper quadrant of the viewing sphere. Partition is rotationally symmetric about upper hemisphere and mirror symmetric about the equator.



(d) Additional views of object when the 3-D viewing space model is used that are not seen from viewing sphere. There are 81 total regions for the subdivision of 3-D space.

Figure 5. Components of aspect graph of nonconvex polyhedron in (a) include viewing sphere partition in (b), along with the representative views of its regions, and also in (d) are additional views seen using the 3-D viewing space model.

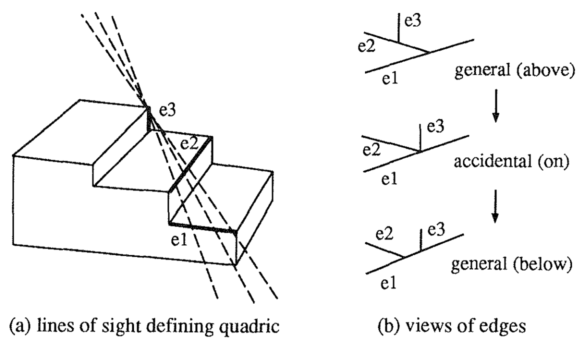


Figure 6. Visual event involving three edges. The lines of sight passing through three skew edges define the rulings of a quadric surface in (a). The views of the three edges as seen from above, on, and below the event surface are shown in (b).

Yet another method exists, that is applicable under either viewing model. This technique uses an intermediate representation known as the *asp* (Plantinga and Dyer, 1986, 1987, 1990). This data structure describes the appearance of every feature of the object in terms of its image plane location as a function of viewpoint. When using a viewing sphere model, the resulting space is 4–D in nature, two for the sphere and two for the image plane. (It would be a 5–D structure for 3–D viewpoint space). A vertex on the object corresponds to a 2–D surface in this space, such that for any viewpoint, its position in the image plane can be calculated. Edges and faces are correspondingly represented by 3–D and 4–D structures in the space. One property of the asp is that occlusion of one feature by another is formed by intersecting the structures corresponding to the two entities. In this way stable feature configurations will eventually be represented by 4–D "volumes", which can then be projected into viewpoint space to define the boundaries in the parcellation.

The asp also has other uses. Some of these will be alluded to in the section on graphics applications. One other is the *rim appearance* model (Seales and Dyer, 1990). In this case, the only observed features of the polyhedral model (which is considered an approximation of a smooth surface) are those edges that project to the occluding contour for a particular viewpoint. These features can be easily identified in the asp, and only those events relating to changes in the visible occluding contour saved (roughly 25% of all actual events). This is a basic approximation to the more exact techniques developed in the next section for curved objects.

4.3 Traversing Viewpoint Space

In general, the resulting viewpoint space will initially be "over–partitioned" by the event boundaries after step two. There are two reasons for this. One is that an event surface may intersect a portion of the object between the observer and the features, making a section of it meaningless due to this global occlusion. Another is that since edges on the object have only finite length, a potential event alignment is actually visible only over a subset of the circle/curve (or plane/surface) that it defines. Many of these potential event surface portions can be discarded entirely, by calculating the true "active" subset, using different visibility tests (Gigus, Canny and Seidel, 1991). Thus some "extra" processing at an early stage of the algorithm can reduce the degree of over–partitioning and save processing in later stages.

Once the partition of viewpoint space is determined, the aspect graph structure is created by traversing the partition. During this traversal, final merging of over–partitioned cells is done. A node is created in the aspect graph for each cell in the partition and an arc for each boundary between cells. Upper bounds on the number of nodes in the aspect graph have been determined to be $\Theta(n^6)$ and $\Theta(n^9)$ for a general polyhedron of n faces under the viewing sphere and 3–D space models, respectively (Plantinga and Dyer, 1990). Of the algorithms mentioned only a few have been fully implemented. These include programs for convex polyhedra (Stewman and Bowyer, 1990; Watts, 1988), general polyhedra (Stewman, 1991), and the rim appearance of polyhedra for 1–D viewing sphere paths (Seales and Dyer, 1990).

5. ASPECT GRAPHS OF CURVED OBJECTS

Research into algorithms for computing exact aspect graphs of curved objects has really occurred only in the past few years. The notion of partitioning the viewing sphere based on visual events for curved objects was first proposed by Callahan and Weiss (Callahan and Weiss, 1985), but no algorithm was given. Original algorithms were developed for solids of revolution (Eggert, 1991; Eggert and Bowyer, 1990, 1992; Kriegman and Ponce, 1990), due to their rotational simplicity. Since then several researchers have analyzed objects bounded by surface patches of different characteristics. These include quadric patches (Chen and Freeman, 1991), C^3 smooth surfaces (Sripradisvarakul and Jain, 1989) and parametric or algebraic surfaces (Ponce and Kriegman, 1990; Petitjean, Ponce and Kriegman, 1992; Rieger, 1990, 1992). All but some of the work on solids of revolution (Eggert and Bowyer, 1992; Eggert, 1991) has been limited to an analysis using the viewing sphere model. The basic approach to computing the aspect graph of curved objects is the same as that for polyhedra, but the details are more complex.

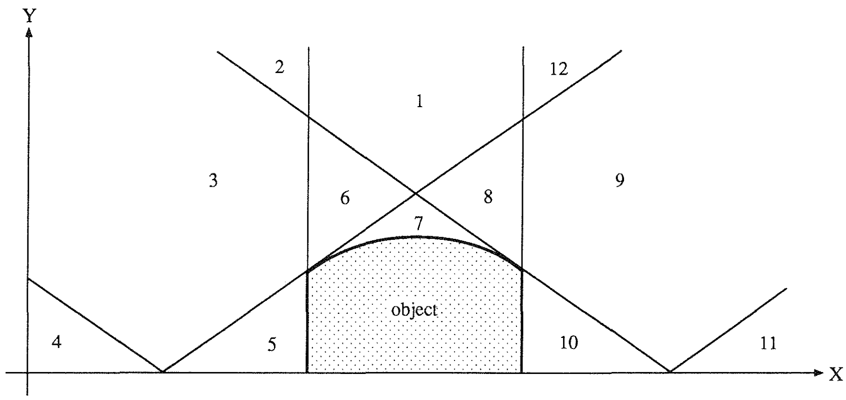
5.1 Visual Events

To begin with, the set of visual events is much more extensive. Most catalogs are based on the results of singularity theory (Whitney, 1955; Arnold, 1983; Koenderink and van Doorn, 1976), which involve the study of projections of surfaces. The two major catalogs currently in use are those of Kergosien for generic smooth surfaces (Kergosien, 1981), and its extension to piece–wise smooth surfaces created by Rieger (Rieger, 1987). Both of these were generated under the assumption of orthographic projection.

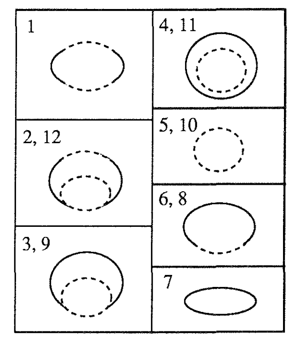
Though there are several different events, they can usually be categorized as to whether or not they involve occlusion. Those that don't, correspond to occurrences when either two contour generators meet upon the object surface (*beak-to-beak* and *lip* events), a contour generator makes contact with an edge of the object (forming curvature-L and three-tangent junctions as shown in Figure 2), or a projected contour forms a *cusp* (a *swallowtail* event). When occlusion occurs, two contours may make contact at a point other than a junction (a *tangent crossing* event), any of the junctions in Figure 2 may be occluded by a contour, or three contours may intersect at a triple point as for polyhedra.

5.2 Parcellating Viewpoint Space

Determining the particular event surfaces of a given object becomes a matter of solving systems of polynomial equations



(a) Cross-section of parcellation of viewpoint space. Remainder of parcellation not shown is symmetric vertically about the X axis. Object is shown shaded. Normal lines are intersections of cone and plane event surfaces with cross-section plane.



(b) Representative views. Solid lines indicate occluding contours while dotted lines indicate edges.

Figure 7. Components of aspect graph of example solid of revolution. Representative views (b) of regions in underlying parcellation (a) of the aspect graph are shown. Note, only unique views are drawn due to horizontal symmetry of object.

for potentially interacting features. This is complicated by the fact that contour generators change their position with viewpoint, whereas edges are viewpoint independent. The resulting event surfaces in 3–D space are ruled, tangent developable surfaces. The intersections of these surfaces with the viewing sphere are polynomial curves. Because of this, the creation of the parcellation of viewpoint space is the most difficult stage of the algorithm.

For solids of revolution, an event surface in 3–D space is restricted to also be rotationally symmetric about the object axis, meaning it is either a plane, cylinder, cone or hyperboloid of one sheet. Also, 3–D space can be adequately represented by a plane containing the axis of the object. Thus the intersection curves of the event surfaces with this plane can be processed using a plane sweep technique to construct the 3–D viewpoint space parcellation in the same manner that the viewing sphere was subdivided for polyhedra. This technique has been implemented (Eggert, 1991), as have simpler versions that divide the orthographic viewing sphere (Eggert and Bowyer, 1990; Kriegman and Ponce, 1990), which is only a matter of calculating the latitudes at which the event surfaces intersect it. The parcellation of a cross–section of 3–D space is shown for a simple solid of revolution in Figure 7.

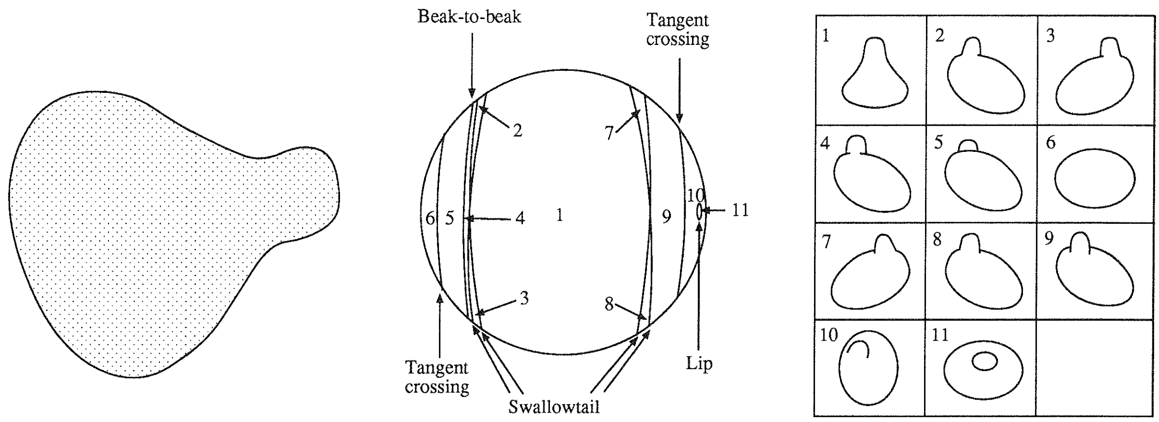
Those authors working on piece–wise smooth objects have different partitioning techniques. Chen and Freeman (Chen and Freeman, 1991) are able to calculate the characteristic views of a quadric–surfaced object by determining the radii of successively larger spheres for which the qualitative form of the subdivision of the sphere changes. Then the event curves are traced out on representative spheres between the critical ones, and the different views calculated. The definitions of the regions from which the views are seen are not saved by this approach. A program to construct this set of characteristic views has been implemented.

Two other groups of researchers (Ponce and Kriegman, 1990; Sripradisvarakul and Jain, 1989) propose methods for tracing out the visual event curves on the surface of the viewing sphere for more general curved shapes, calculating intersection points and determining the eventual viewing regions. The latter also gives a more intuitive development of a slightly different visual event catalog, with defining equations for both viewing models. Unfortunately, this method has not been completely implemented. On the other hand, the former have very recently achieved a complete implementation (Petitjean, Ponce and Kriegman, 1992) assuming orthographic projection. This method relies heavily on the ability to reformulate the aspect graph generation process into a series of steps of solving systems of polynomial equations, for which the authors use several numerical techniques recently reintroduced into the computer vision literature. A simple example generated for a gourd shaped object is shown in Figure 8.

It has so far proved too difficult to construct structures such as the geometric incidence lattice for the event surfaces of curved objects, due to the complicated nature of their intersections. As an alternative to this, one researcher has developed a symbolic algorithm (Rieger, 1992) that derives the equations of the event curves on the viewing sphere in terms of the object definition, again assuming orthographic projection. While certain portions of this algorithm have been implemented using symbolic manipulation packages, a complete system does not yet exist. Unfortunately, this approach may become prohibitive for complex objects, and it is not known how easily the algorithm can be extended to encompass perspective projection in 3–D space. But perhaps it may be effectively combined with those techniques above.

6. ASPECT GRAPHS FOR ARTICULATED OBJECTS

While research on curved objects is reaching the implementation stage, theoretical advances are beginning for shapes which are parameterized to allow articulated connections between rigid parts, an *articulated assembly* (Eggert, Sallam and Bowyer, 1992; Sallam, Stewman and Bowyer, 1990), or to allow deformations defining nonrigid motion (Wilkins, Goldgof and Bowyer, 1991). The number and limits of each degree of freedom of motion can define a *configuration space* for the object, each point in it representing a particular instance in a continuous shape family. If both the viewpoint space and this configuration space are considered together, another aspect space can be defined, conceptually similar to that used for the asp. Each visual event surface that was previously defined in terms of fixed features, is now additionally parameterized in terms of the degrees of freedom of



(a) Side view of squash-shaped object

(b) Parcellation of viewing sphere. Regions are labeled with numbers, curves are labeled according to visual event type.

(c) Representative views of regions on sphere

Figure 8. Components of aspect graph for squash-shaped object. The shape in (a) is defined by the implicit equation $4y^4 + 3xy^2 - 5y^2 + 4z^2 + 5x^2 - 2xy + 2x + 3y - 1 = 0$. Visual events defining curves on view sphere (b) partition regions with representative views shown in (c). This figure is adapted from figures in (Petitjean, Kriegman and Ponce, 1992).

movement. Each "volume" in the multi–dimensional space bounded by these surfaces corresponds to an aspect of the object. No direct implementation of this method yet exists.

While the above approach is a straightforward extension of the aspect graph concept, yet another generalization has been proposed (Sallam, Stewman and Bowyer, 1990). Here, the viewpoint and configuration spaces are examined separately, leading to the concepts of a *general configuration* and an *accidental configuration* of the assembly, as defined with respect to the aspect graph of the configuration space:

- A point in configuration parameter space represents a *general configuration* if, for every possible direction, an infinitesimal movement from that point results in a configuration whose aspect graph is isomorphic to that of the original configuration. (Two aspect graphs are considered isomorphic if there is an isomorphism between the graphs such that corresponding nodes are attributed with isomorphic image structure graphs.)
- A point in the configuration parameter space represents an *accidental configuration* if there exists an infinitesimal movement that results in a configuration whose aspect graph is not isomorphic to that of the original configuration.

This leads to a hierarchical graph structure, referred to as the *visual potential* of the articulated assembly, as shown in Figure 9. Nodes at the highest level represent ranges of general configuration in which the aspect graph is constant, while lower level nodes correspond to the representative aspect graphs of each configuration. Two types of *configuration events* define the boundaries of configuration space. One is a *definition* event, in which some boundary in viewpoint space is either created or degenerates as the object changes. The other is a *coincidence* event, in which boundary elements temporarily have the same defining equation. In Figure 10 a simple articulated assembly composed of two blocks is shown, along with its appearance at each of the critical parameter values in the configuration space. This representation was generated using sampling techniques similar to those for approximate aspect graphs.

7. APPLICATIONS OF THE ASPECT GRAPH

Since the aspect graph encodes information about the visual appearance of an object, it is natural that applications exist in both the areas of computer vision and computer graphics. There are currently not many systems which use the aspect graph, but this may change as more implementations of aspect graph algorithms become available.

7.1 Use in Computer Vision

The approximate aspect graph has been used by a number of researchers in "CAD–based vision" efforts (mostly in a theoretical capacity). In this context, the aspect graph is typically used as a method for feature prediction, especially in determining what features are likely to be seen in conjunction with one another. The importance or "saliency" of features could be determined by their frequency of oc-

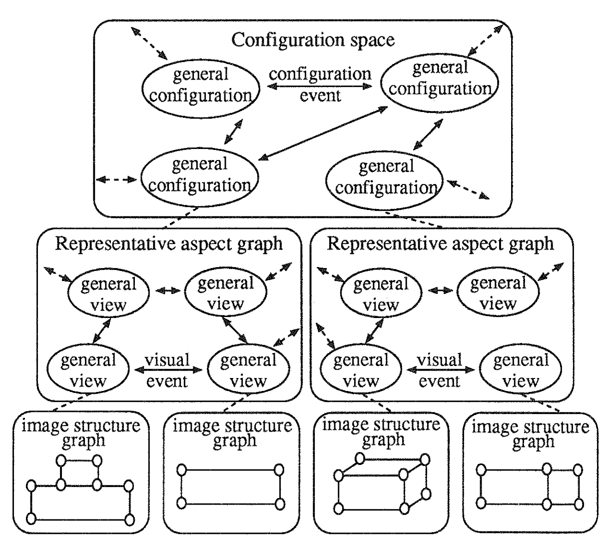
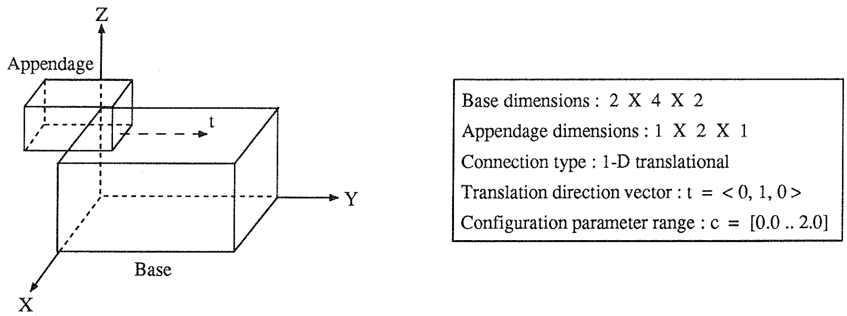
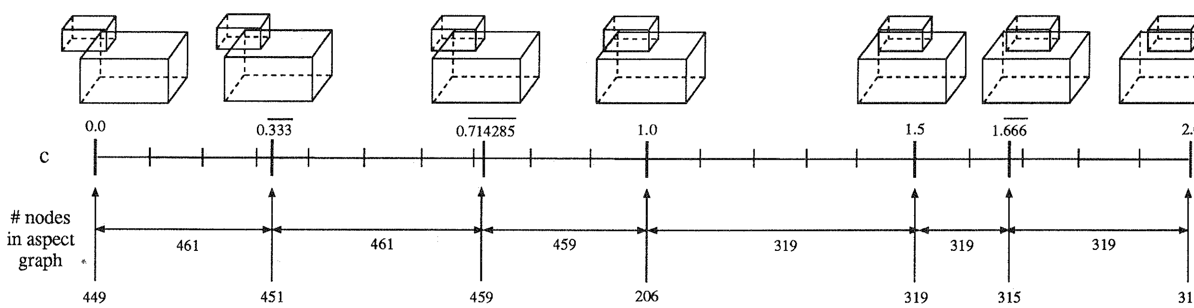


Figure 9. Hierarchical visual potential for articulated assemblies.



(a) Definition of articulated object composed of two rectangular blocks with 1-D translational connection.



(b) Subdivision of configuration space (0 < c < 2). Accidental configurations are drawn for critical parameter values indicated by vertical arrows. General configurations exist between.The number of nodes in the representative aspect graphs for each configuration is also given.

Figure 10. Depiction of hierarchical visual potential for articulated assembly in (a) as given by configuration space parcellation in (b).

currence in the views of objects in a database. All of this information could then be used as input to an algorithm for creating an *interpretation tree* to guide the interpretation of features extracted from an image. Implemented systems include (Chen and Kak, 1989; Hansen and Henderson, 1989; Hutchinson and Kak, 1989; Ikeuchi, 1987).

The use of entire aspects in the recognition process has been discussed with regards to the characteristic views approach (Wang and Freeman, 1990) and also the asp representation (Plantinga and Dyer, 1987). Only two others have used the additional information that the aspect graph provides, namely, the regions of viewpoint space from which views are seen. Ikeuchi (Ikeuchi, 1987) notes that only "linear" shape changes occur for a view from within the region an aspect is seen, and therefore developed a separate pose estimation strategy for each aspect. The other (Bowyer, Eggert, Stewman and Stark, 1989; Stark, Eggert and Bowyer, 1988) uses an exact aspect graph. For the domain of convex polyhedra, potential matching aspects were indexed according to the number of visible faces in a view, then matched using a nonlinear optimization search for pose parameters, bounded to stay within an aspect's cell of space. The best matching aspect and its pose were chosen as the solution.

Efforts to make more efficient use of the potentially large size of a database of aspect graphs include finding better indexing schemes based on features extracted from the image (Swain, 1988), and condensing the total number of views based on different equivalence criteria (Stewman, Stark and Bowyer, 1987; Stewman, 1991; Wang and Freeman, 1990). In the latter approach, another feature of the aspect graph can be put to use. That is the ability to predict the location of a second view that would disambiguate between several potential objects as determined from the current view. One last recent effort (Dickinson, Pentland and Rosenfeld, 1992) involves using only the aspect graphs of a set of primitives (geons) used to construct the objects in a database. Each primitive is recognized using some of the techniques mentioned above, and then further processing links the found primitives to a known object.

7.2 Use in Computer Graphics

Computer graphics is the inverse process of computer vision. The problem of real-time display of a scene as a viewer interactively moves around it, is another application that can take advantage of representations like the aspect graph and asp. Early efforts such as *binary space partitioning* (Fuchs, Kedem and Naylor, 1980) and special purpose display routines for polyhedra based on edge visibility (Goad, 1982), used representations similar in nature. The advantage of the aspect graph is that hidden-line removal for views of the object is preprocessed during its construction. The costly calculations necessary for this step are not repeated for each new view. Once the correspondence of viewpoint to aspect is made, only truly visible contours are rendered.

Simple wire-frame animation has been achieved for polyhedra (Stewman, 1991) and solids of revolution (Eggert, 1991) along particular paths by drawing the aspect associated with each viewpoint along a path. More efficient graphical rendering has been accomplished using the asp for polyhedra. Since each edge feature's appearance is implicitly encoded in the data structure, it is possible to precompute only those local edge changes that occur at particular points along a known path. During rendering only these small changes need to be made rather than updating an entire new view each time. This technique has led to efficient hidden-line elimination (Plantinga, Dyer and Seales, 1990), and shaded rendering and shadow computation (Seales, 1991) of a polyhedral scene as seen by a moving camera that is interactively controlled by the user. Similar techniques are also possible based on interpreting the visual event changes along a given path.

8. THE SCALE SPACE ASPECT GRAPH

One problem with the aspect graph representation is that the size can become quite large as the object complexity increases (Faugeras *et al.*, 1992). Some possible remedies to this were mentioned in the last section. However, other efforts at determining the set of "important" aspects center on the use of scale information. To date, the aspect graph has been computed only under the ideal assumptions of perfect resolution in the viewpoint, the projected image and the object shape, leading to the following practical difficulties:

- a node in the aspect graph may represent a view of the object that is seen only from an extremely small cell of viewpoint space,
- views represented by two neighboring nodes may differ only in some realistically indistinguishable detail, and
- small changes in the detail of 3-D object shape may drastically affect the number of aspect graph nodes.

We now address how these problems may be solved using the concept of *scale space* (Witkin, 1986). When the phrase "scale space of X" is used, it is taken to mean a parameterized family of X in which the detail of features in X monotonically decreases with increasing scale. Under some parameterized transformation, usually Gaussian blurring, the nature of X is changed in such a way that its qualitative features at a given scale can be traced back across all lower scales ("causality"). In this paper, X is an aspect graph, or more precisely, the corresponding parcellation of viewpoint space underlying the qualitative description of the aspect graph (Eggert, *et al.*, 1992a, 1992b). At the scale value $\sigma = 0$, the parcellation of viewpoint space, and therefore the aspect graph, is exactly as computed by some known algorithm. Ideally, as σ increases, this parcellation should deform so that at certain discrete values of scale the aspect graph becomes simpler.

The qualitative structure of scale space, which is a multi-dimensional space parameterized by viewpoint location and scale value (for example, a 4-D (x, y, z, σ) space under perspective projection), can be represented in the two forms mentioned earlier for articulated assemblies, with σ considered as the configuration parameter. Measures of aspect importance can be calculated in terms of the range of scale for which they exist, the "volume" of the corresponding cell in scale space, or other quantities. The question remains as to what exactly σ represents.

8.1 Interpretations of Scale

Most scale space representations interpret the scale parameter in terms of the variance of a Gaussian kernel used for blurring (Koenderink, 1984). At least three different methods of blurring can be thought of when considering the problems mentioned earlier.

8.1.1 Viewer size. This interpretation corresponds to examining the relative sizes of the aspect cells. Related work in the past (Ben-Arie, 1991; Eggert, 1991; Kender and Freudenstein, 1987; Wang and Freeman, 1990; Watts, 1988) dealt with viewing probabilities of aspects based on cell volume. Here a finite-sized observer is modeled by a sphere in viewpoint space (Eggert, *et al.*, 1992b). Increasing its radius as a function of scale will limit which cells the observer can fit into. Alternatively one can envision the process of blurring the cell boundaries until the cell disappears.

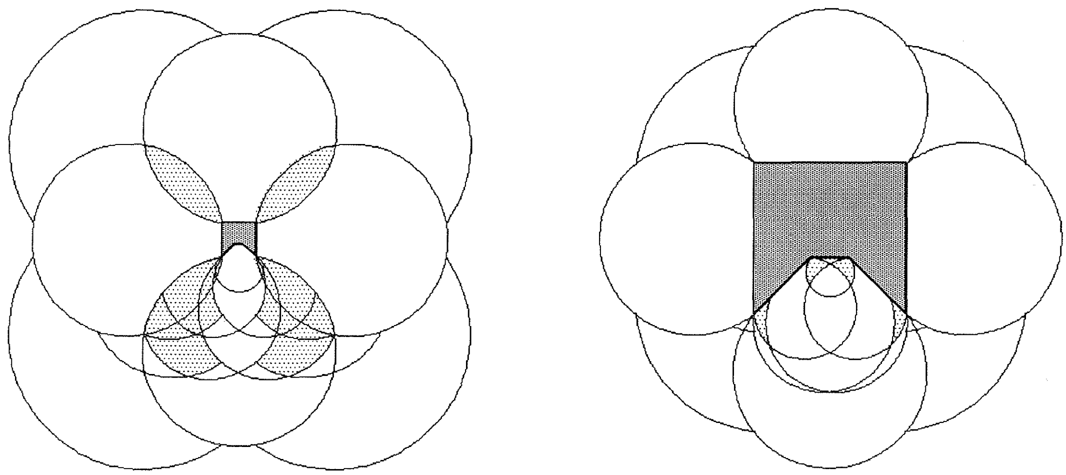
8.1.2 Image Feature size. This interpretation corresponds to imposing a finite resolution upon the image plane. If the size of a particular feature in the image structure graph falls below the resolution of a pixel, it is no longer represented. At varying values of scale, two cells whose only difference in aspect was some small feature, may be merged as that feature disappears. A particular realization of this interpretation is to measure feature size in terms of the angle of visual arc it spans. Image resolution is implicitly related through the scale parameter to a particular visual angle. In terms of the parcellation of viewpoint space, the visual event boundaries change shape with varying scale. In (Eggert, *et al.*, 1992a) an example is given for polygons in a plane. The ideal event lines are deformed into portions of circular arcs. See Figure 11 for two example parcellations of a six-sided polygon. These deformations relate to the fact that for a finite resolution camera the observer can only retreat a specific distance from the object before it appears as only a dot.

8.1.3 Object Detail. This interpretation corresponds to smoothing the object surface, thereby removing small features which might visually interact. This might also relate to the visual loss of detail noticed while moving away from an object. The *dynamic shape* concept (Koenderink, 1990) initially seems appealing, but this is a form of volumetric blurring which has no regard for visual appearance of the object. Rather than work upon *solid* shape, it is necessary to deal with *surface* shape. In (Eggert, *et al.*, 1992b) a technique is described in which the object surface is smoothed in a direction perpendicular to the viewing direction at every viewpoint. The visual appearance of the object shape is then reduced in detail as the smoothing kernel that is a function of the scale parameter is increased in size.

9. CONCLUDING REMARKS

The past and present of aspect graphs has now been discussed. So what is the future? Work will probably focus in three main areas; the view description, the domain of representable objects in both theory and practice, and developing recognition systems that take full advantage of the aspect graph's properties.

The image structure graph is a very idealized view representation. And while certain efforts are in progress (Kriegman, Vijayakumar and Ponce, 1992), it is quite difficult to extract the necessary information out of a noisy image to reliably label a line drawing. In regards to this problem, new representations like the rim appearance (Seales and Dyer, 1990), which focus on the visually stronger edges in a view, show promise. Other alternative view descriptions, such as a topological representation of the actual intensity image, have been proposed by Koenderink (Koenderink, 1984). With this form, additional features such as camera parameters and lighting could be accounted for. Initial experimentation to find the required new set of visual events has begun (Waldon and Dyer, 1991). A second approach is to use a richer source



(a) Viewpoint space parcellation at visual arc angle of 10 degrees. (b) Viewpoint space parcellation at visual arc angle of 52 degrees.

Figure 11. Cross-sections of scale space of example nonconvex polygon for two sample scale values. The polygon is darkly shaded. Lightly shaded regions in (a) are cells that do not exist in (b), and vice versa. A direct correspondence exists for all other cells between the two scales. Note, however, that several different scale events occur between depicted scale values.

of data, such as 3-D range imagery. Following the examples of certain researchers using approximate aspect graphs, different view descriptions are being examined (Kaiser, Bowyer and Goldgof, 1991; Raja and Jain, 1992).

Continuing to generalize the aspect graph concept to larger object classes should make use of the methods developed for articulated assemblies. Unfortunately, the development of visual event catalogs for these extensions will probably proceed more theoretically than practically. It appears that in order for implementations of the existing ideas to occur, new methods of calculating and representing surface intersections in high dimensions, either numerically or symbolically, must be developed. The field of computational geometry may prove useful in this regard.

And lastly, more efficient use must be made of aspect graphs in working recognition systems. There is adequate room to further explore the various interpretations of the scale space aspect graph, either individually or in conjunction, in an effort to determine the "important" aspects. Perhaps psychophysical evidence can also be applied towards this task.

10. REFERENCES

Arnol'd, V.I., 1983. Singularities of Systems of Rays, *Russian Mathematical Surveys*, 38:87–176.

Ben-Arie, J., 1990. Probabilistic Models of Observed Features and Aspects with an Application to Weighted Aspect Graphs, *Pattern Recognition Letters*, 11:421–427.

Bowyer, K.W. and Dyer, C.R., 1990. Aspect Graphs: An Introduction and Survey of Recent Results, *International Journal of Imaging Systems and Technology*, 2:315–328.

Bowyer, K., Eggert, D., Stewman, J. and Stark, L., 1989. Developing the Aspect Graph Representation for Use in Image Understanding, *Proceedings of the DARPA Image Understanding Workshop*, 831–849.

Burns, J.B. and Kitchen, L.J., 1988. Rapid Object Recognition From a Large Model Base Using Prediction Hierarchies, *Proceedings of the DARPA Image Understanding Workshop*, 711–719.

Callahan, J. and Weiss, R., 1985. A Model for Describing Surface Shape, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 240–245.

Camps, O.I., Shapiro, L.G. and Haralick, R.M., 1991. PREMIO: An Overview, *Proceedings of the IEEE Workshop on Directions in Automated CAD-Based Vision*, 11–21.

Castore, G., 1984. Solid Modeling, Aspect Graphs, and Robot Vision, in **Solid Modeling by Computer**, (Plenum Press, New York), 277–292.

Castore, G. and Crawford, C., 1984. From Solid Model to Robot Vision, *Proceedings of the 1st International Conference on Robotics*, 90–92.

Chakravarty, I. and Freeman, H., 1982. Characteristic Views as a Basis for Three-Dimensional Object Recognition, *Proceedings of the SPIE: Robot Vision*, 336:37–45.

Chen, S. and Freeman, H., 1991. On the Characteristic Views of Quadric-Surfaced Solids, *Proceedings of the IEEE Workshop on Directions in Automated CAD-Based Vision*, 34–43.

Chen, C. and Kak, A., 1989. A Robot Vision System for Recognizing 3-D Objects in Low-Order Polynomial Time, *IEEE Transactions on Systems, Man, and Cybernetics*, 19:1535–1563.

Dickinson, S.J., Pentland, A.P. and Rosenfeld, A., 1992. 3-D Shape Recovery Using Distributed Aspect Matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:174–198.

Eggert, D., 1991. Aspect Graphs of Solids of Revolution, Ph.D. Dissertation, Department of Computer Science & Engineering, University of South Florida, Tampa, FL-USA.

Eggert, D. and Bowyer, K., 1990. Computing the Orthographic Projection Aspect Graph for Solids of Revolution, *Pattern Recognition Letters*, 11:751–763.

Eggert, D. and Bowyer, K., 1992. Computing the Perspective Projection Aspect Graph of Solids of Revolution, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, to appear.

Eggert, D., Bowyer, K., Dyer, C., Christensen, H.,Goldgof, D., 1992a. The Scale Space Aspect Graph, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, to appear.

Eggert, D., Bowyer, K., Dyer, C., Christensen, H.,Goldgof, D., 1992b. "Applying the Scale Space Concept to Perspective Projection Aspect Graphs", in **Theory and Applications of Image Analysis: Selected Papers from the Seventh Scandinavian Conference for Image Analysis**, (World Scientific Pub., Singapore), to appear.

Eggert, D., Sallam, M. and Bowyer, K., 1992. Generalizations of the Aspect Graph for Objects with Moving Parts, in **Advances in Artificial Intelligence Research**, (JAI Press, Greenwich, Connecticut), vol. 3, to appear.

Faugeras, O., Mundy, J., Ahuja, N., Dyer, C., Pentland, A., Jain, R., Ikeuchi, K. and Bowyer, K., 1992. Panel Theme: Why Aspect Graphs Are Not (Yet) Practical for Computer Vision, *CVGIP: Image Understanding*, 55:212–218.

Fekete, G. and Davis, L.S., 1984. Property Spheres: A New Representation for 3-D Object Recognition, *Proceedings of the IEEE Workshop on Computer Vision: Representation and Control*, 192–201.

Fuchs, H., Kedem, Z.M. and Naylor, B.F., 1980. On Visible Surface Generation by A Priori Tree Structures, *Proceedings of SIGraph '80*, 124–133.

Gigus, Z., Canny, J. and Seidel, R., 1991. Efficiently Computing and Representing Aspect Graphs of Polyhedral Objects, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:542–551.

Gigus, Z. and Malik, J., 1990. Computing the Aspect Graph for Line Drawings of Polyhedral Objects, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:113–122.

Goad, C., 1982. Special Purpose Automatic Programming for Hidden-Surface Elimination, *Computer Graphics*, 16:167-178.

Goad, C., 1983. Special Purpose Automatic Programming for 3-D Model-Based Vision, *Proceedings of the ARPA Image Understanding Workshop*, 94–104.

Gualtieri, J.A., Baugher, S. and Werman, M., 1989. The Visual Potential: One Convex Polygon, *Computer Vision, Graphics, and Image Processing*, 46:96–130.

Hansen, C. and Henderson, T., 1989. CAGD-Based Computer Vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:1181–1193.

Hebert, M. and Kanade, T., 1985. The 3D-Profile Method for Object Recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 458–463.

Hutchinson, S.A. and Kak, A., 1989. Planning Sensing Strategies in a Robot Work Cell With Multi-Sensor Capabilities", *IEEE Transactions on Robotics and Automation*, 5: 765–783.

Ikeuchi, K., 1987. Generating an Interpretation Tree From a CAD Model for 3D-Object Recognition in Bin-Picking Tasks, *International Journal of Computer Vision*, 1:145–165.

Kaiser, B.J., Bowyer, K.W., and Goldgof, D.B., 1991. On Exploring the Definition of a Range Image Aspect Graph, *Proceedings of the 7th Scandinavian Conference on Image Analysis*, 652—658.

Kender, J.R. and Freudenstein, D.G., 1987. What is a 'Degenerate' View?, *Proceedings of the ARPA Image Understanding Workshop*, 589—598.

Kergosien, Y.L., 1981. La Famille des Projections Orthogonales D'une Surface et ses Singularites, *Comptes Rendes, Series A*, 292:929–932.

Koenderink, J.J., 1984. The Structure of Images, *Biological Cybernetics*, 50:363–370.

Koenderink, J.J., 1990. **Solid Shape**, (MIT Press, Cambridge, Mass.).

Koenderink, J.J. and van Doorn, A.J., 1979. The Internal Representation of Solid Shape With Respect to Vision, *Biological Cybernetics*, 32:211–216.

Koenderink, J.J. and van Doorn, A.J., 1976. The Singularities of the Visual Mapping, *Biological Cybernetics*, 24:51–59.

Korn, M.R. and Dyer, C.R., 1987. 3-D Multiview Object Representations for Model-Based Object Recognition, *Pattern Recognition*, 20:91–103.

Kriegman, D. and Ponce, D., 1990. Computing Exact Aspect Graphs of Curved Objects: Solids of Revolution, *International Journal of Computer Vision*, 5:119–135.

Kriegman, D., Vijayakumar, B. and Ponce, J., 1992. Strategies and Constraints for Recognizing and Locating Curved 3-D Objects from Monocular Image Features, *Proceedings of the 2nd European Conference on Computer Vision*, to appear.

Malik, J., 1987. Interpreting Line Drawings of Curved Objects, *International Journal of Computer Vision*, 1:73–103.

Minsky, M., 1975. A Framework for Representing Knowledge, in **The Psychology of Computer Vision**, (McGraw-Hill Publishers, New York), 211–277.

Palmer, S., Rosch, E. and Chase, P., 1981. Canonical Perspective and the Perception of Objects, *Attention and Performance IX*, 135–151.

Perrett, D.I., Harries, M.H., Bevan, R., Thomas, S., Benson, P.J., Mistlin, A.J., Chitty, A.J., Hietanen, J.K. and Ortega, J.E., 1989. Frameworks of Analysis for the Neural Representation of Animate Objects and Actions, *Journal of Experimental Biology*, 146:87–113.

Petitjean, S., Ponce, J. and Kriegman, D., 1992. Computing Exact Aspect Graphs of Curved Objects: Algebraic Surfaces, Technical Report UIUC-BI-AI-RCV-92-02, The Beckman Institute, University of Illinois, Urbana, IL-USA.

Plantinga, H. and Dyer, C.R., 1986. An Algorithm for Constructing the Aspect Graph, *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, 123–131.

Plantinga, H. and Dyer, C.R., 1987. The Asp: A Continuous Viewer–Centered Representation for 3–D Object Recognition, *Proceedings of the 1st International Conference on Computer Vision*, 626–630.

Plantinga, H. and Dyer, C.R., 1990. Visibility, Occlusion and the Aspect Graph, *International Journal of Computer Vision*, 5:137–160.

Plantinga, H., Dyer, C.R. and Seales, W.B., 1990. Real–Time Hidden–Line Elimination for a Rotating Polyhedral Scene Using the Aspect Representation, *Proceedings of the Graphics Interface '90*, 9–16.

Ponce, J. and Kriegman, D., 1990. Computing Exact Aspect Graphs of Curved Objects: Parametric Surfaces, *Proceedings of the 8th National Conference on Artificial Intelligence*, 1074–1079.

Raja, N.S. and Jain, A.K., 1992. Obtaining Generic Parts from Range Data Using a Multiview Representation, *Proceedings of the SPIE: Applications of Artificial Intelligence X: Machine Vision and Robotics*, 1708:602-613.

Rieger, J.H., 1987. On the Classification of Views of Piecewise Smooth Objects, *Image and Vision Computing*, 5:91–97.

Rieger, J.H., 1990. The Geometry of View Space of Opaque Objects Bounded by Smooth Surfaces, *Artificial Intelligence*, 44:1–40.

Rieger, J.H., 1992. Global Bifurcation Sets and Stable Projections of Non–Singular Algebraic Surfaces, *International Journal of Computer Vision*, to appear.

Rosenfeld, A., 1987. Recognizing Unexpected Objects: A Proposed Approach, *International Journal of Pattern Recognition and Artificial Intelligence*, 1:71–84.

Sallam, M.Y., Stewman, J.S. and Bowyer, K.W., 1990. Computing the Visual Potential of Articulated Assemblies, *Proceedings of the 3rd International Conference on Computer Vision*, 636–643.

Seales, W.B., 1991. Appearance Models of 3–D Shape for Machine Vision and Graphics, Ph.D. Dissertation, Computer Sciences Department, University of Wisconsin, Madison, WI-USA.

Seales, W.B. and Dyer, C.R., 1990. Modeling the Rim Appearance, *Proceedings of the 3rd International Conference on Computer Vision*, 698–701.

Shapiro, L.G., 1988. A CAD–Model–Based System for Object Localization, *Proceedings of the SPIE: Digital and Optical Shape Representation*, 938:408–418.

Sripradisvarakul, T. and Jain, R., 1989. Generating Aspect Graphs for Curved Objects, *Proceedings of the IEEE Workshop on Interpretation of 3D Scenes*, 109–115.

Stark, L., Eggert, D. and Bowyer, K., 1988. Aspect Graphs and Nonlinear Optimization in 3–D Object Recognition, *Proceedings of the 2nd International Conference on Computer Vision*, 501–507.

Stewman, J.H., 1991. Viewer Centered Representations for Polyhedral Objects, Ph.D. Dissertation, Department of Computer Science & Engineering, University of South Florida, Tampa, FL-USA.

Stewman, J.H. and Bowyer, K.W., 1988. Creating the Perspective Projection Aspect Graph of Polyhedral Objects, *Proceedings of the 2nd International Conference on Computer Vision*, 494–500.

Stewman, J.H. and Bowyer, K.W., 1990. Direct Construction of the Perspective Projection Aspect Graph of Convex Polyhedra, *Computer Vision, Graphics, and Image Processing*, 51:20–37.

Stewman, J., Stark, L. and Bowyer, K., 1987. Restructuring Aspect Graphs into Aspect– and Cell–Equivalence Classes for Use in Computer Vision, *Proceedings of the 13th International Workshop on Graph–Theoretic Concepts in Computer Science*, 230–241.

Swain, M., 1988. Object Recognition From a Large Database Using a Decision Tree, *Proceedings of the DARPA Image Understanding Workshop*, 690–696.

Waldon, S., and Dyer, C. 1991. Towards an Aspect Graph for Photometric Imaging, Technical Report, Computer Science Department, University of Wisconsin, Madison, WI–USA.

Wang, R. and Freeman, H., 1990. Object Recognition Based on Characteristic View Classes, *Proceedings of the 10th International Conference on Pattern Recognition*, 8–12.

Watts, N., 1988. Calculating the Principal Views of a Polyhedron, *Proceedings of the 9th International Conference on Pattern Recognition*, 316–322.

Whitney, H., 1955. On Singularities of Mappings of Euclidean Spaces. I. Mappings of the Plane into the Plane, *Annals of Mathematics*, 62:374–410.

Wilkins, B., Goldgof, D. and Bowyer, K.W., 1991. Toward Computing the Aspect Graph of Deformable Generalized Cylinders, *Proceedings of the SPIE: Applications of Artificial Intelligence IX*, 1468:662–673.

Witkin, A.P., 1986. Scale Space Filtering, in **From Pixels to Predicates**, (Ablex Publishing Corp., Norwood, NJ), 5–19.