

**VISIBILITY, OCCLUSION
AND THE ASPECT GRAPH**

by

Harry Plantinga

and

Charles R. Dyer

Computer Sciences Technical Report #736

December 1987

Visibility, Occlusion, and the Aspect Graph*

W. Harry Plantinga – Charles R. Dyer

Department of Computer Sciences
University of Wisconsin - Madison

Abstract

In this paper we study the ways in which the topology of the image of a polyhedron changes with changing viewpoint. We catalog the ways that the “topological appearance” or *aspect* can change. This enables us to find maximal regions of viewpoints of the same aspect. We use these techniques to construct the *viewpoint space partition* (VSP), a partition of viewpoint space into maximal regions of constant aspect, and its dual, the *aspect graph*. In this paper we present tight bounds on the maximum size of the VSP and the aspect graph and give algorithms for their construction, first in the convex case and then in the general case. In particular, we give tight bounds on the maximum size of $\Theta(\mathbf{n}^2)$ and $\Theta(\mathbf{n}^6)$ under orthographic projection with a 2-D viewpoint space and of $\Theta(\mathbf{n}^3)$ and $\Theta(\mathbf{n}^9)$ under perspective projection with a 3-D viewpoint space. We also present properties of the VSP and aspect graph and give algorithms for their construction.

* This work was supported in part by the NSF under grant DCR-8520870.



Table of Contents

1. Introduction.....	1
2. Visibility and Occlusion.....	3
3. The Viewpoint Space Partition and the Aspect Graph.....	7
4. Convex Polyhedra.....	12
4.1. Orthographic Case.....	13
4.1.1. An Upper Bound.....	13
4.1.2. A Lower Bound.....	14
4.1.3. An Algorithm.....	15
4.2. Perspective Case.....	16
4.2.1. An Upper Bound.....	16
4.2.2. A Lower Bound.....	17
4.2.3. An Algorithm.....	17
5. Non-Convex Polyhedra.....	18
5.1. Orthographic Case.....	18
5.1.1. Occlusion under Orthographic Projection.....	19
5.1.1.1. Edge-Vertex Occlusion Boundaries.....	19
5.1.1.2. Edge-Edge-Edge Occlusion.....	21
5.1.2. An Upper Bound.....	23
5.1.3. A Lower Bound.....	23
5.1.4. An Algorithm.....	25
5.1.4.1. The Aspect Representation.....	27
5.1.4.1.1. Representing Asps.....	30
5.1.4.1.2. Asps and Visibility.....	31
5.1.4.1.3. Aspect Surfaces.....	33
5.1.4.1.4. Size of Asps.....	37
5.1.4.1.5. Constructing the Asp.....	38
5.1.4.2. Using the Asp to Construct the VSP.....	39
5.2. Perspective Case.....	40
5.2.1. Occlusion under Perspective Projection.....	41
5.2.2. Upper Bound.....	41
5.2.3. Lower Bound.....	42
5.2.4. An Algorithm.....	43
5.2.4.1. The Asp under Perspective Projection.....	44
5.2.4.2. Using the Asp to Construct the VSP.....	46
6. Related work.....	48
7. Conclusion.....	50
REFERENCES.....	52



1. Introduction

A number of researchers in computer vision have suggested that a reasonable approach to 3-D object recognition is the “characteristic view” approach, in which several “characteristic” views of an object are stored and all are compared to an image for the purpose of recognition [Rosenfeld, 1987; Chakravarty and Freeman, 1982; Castore and Crawford, 1983; Ikeuchi, 1986; Korn and Dyer, 1987; Schneier, Lumia, and Kent, 1986]. The idea of the characteristic view approach is that if a view similar to the image is stored, the recognition process will be simple and fast. The approach works particularly well in parallel, since the image can be compared with characteristic views in parallel.

The views are generally taken to be topologically distinct in the interest of minimizing the number of views necessary, with the idea that a topological match of the image and the view should be enough for recognition. However, in order to calculate these characteristic views one must calculate ranges of viewpoints with particular appearance characteristics, for example, maximal regions of viewpoints in which the object has the same “topological appearance.” No general algorithm exists for constructing such views along with the viewpoints over which they are defined.

In the automatic visual inspection of parts, a computer may control a moveable camera in order to inspect a part and determine whether it was manufactured or assembled correctly. In order to do so, the computer must be able to direct the camera to a viewpoint from which it can see appropriate sections of the part. To do this, the computer must calculate viewpoints from which particular sets of faces of the part are visible.

In motion planning for a robot arm, it may be necessary to find a path for the arm to approach an object in such a way that a particular face is visible or exposed. For example, in order to insert a screw, it is necessary to know from what viewing direction the bottom of the screw-hole is visible. The screw can then be inserted into the hole in that direction. Thus, it is necessary to know from what viewpoints certain parts of the object are visible.

These problems have in common the need to know from which viewpoints an object has particular characteristics of appearance. In this paper we present techniques for dealing with problems such as these. We show how to find and represent regions of viewpoint space of particular visibility characteristics and

boundaries in viewpoint space at which the appearance changes. We do this by cataloging the ways that the *aspect* or topological appearance of a polyhedron can change with changing viewpoint. These changes we call *events*. The events give rise to boundaries in viewpoint space separating regions of viewpoints from which the aspect is different. Finding all of these boundaries enables us to construct a partition of viewpoint space into regions of constant aspect, which we call the *viewpoint space partition (VSP)*. We can merge these regions to find the regions of viewpoint space in which a particular face or feature is visible.

We also use the VSP to construct the aspect graph, a graph with a vertex for every “topologically distinct” view of an object (*aspect*) and with edges connecting adjacent aspects. We show how to do these things under two viewing models. The first uses orthographic projection and restricts viewpoints to points on an infinite sphere containing the object, and the second uses perspective projection and allows viewpoints anywhere in the world. We give tight bounds on the maximum sizes of the VSP and the aspect graph in all cases, thus determining the maximum number of topologically-distinct views of an object. In Table 1 we present a summary of the size of the VSP and aspect graph in various cases, and in Table 2 we present a summary of the time required to construct them under various conditions.

	convex polyhedra	non-convex polyhedra
orthographic	$O(n^2)$	$O(n^6)$
perspective	$O(n^3)$	$O(n^9)$

Table 1. The maximum size of the VSP and aspect graph

	convex polyhedra	non-convex polyhedra
orthographic	$O(n^2)$	$O(n^6 \log n)$
perspective	$O(n^3)$	$O(n^9 \log n)$

Table 2. Construction time for the VSP and aspect graph

In an earlier paper we dealt with the construction of the aspect graph for polyhedra under orthographic projection [Plantinga and Dyer, 1986]. That paper incorrectly claims that the aspect graph for a non-convex polyhedron under orthographic projection has size $O(n^4)$. The error occurred in the analysis of the algorithm—overlooking edge-edge-edge occlusion (see Section 5.1.1.3). The algorithm given in that paper is correct, however. We thanks John Canny [1987] for pointing out the error.

The events or ways that aspect can change are also relevant to understanding the changing image field resulting from a moving viewer. In understanding an image field that changes as a result of change in viewpoint, one can learn about the surfaces in the world by finding regions of the image field that yield a constant value of certain functions. For example, regions of a constant gradient of color or texture may correspond to a single object, and regions with a constant gradient of optical flow may correspond to a flat surface.

As the viewpoint changes, some regions in the image will disappear or appear, and others will split or merge. If the world is static, these changes are the same as the events separating regions of viewpoints of constant aspect. Changes in the image occur whenever the viewpoint crosses one of the aspect boundaries in viewpoint space.

In Section 2 we describe our viewing models and introduce the notions of visibility and occlusion. In Section 3 we define the viewpoint space partition and the aspect graph and discuss some of their properties. In Section 4 we give upper and lower bounds on the maximum size of the VSP and the aspect graph for convex polyhedra and present an algorithm for their construction. We do this first under the orthographic viewing model and then the perspective model. In Section 5 we consider the same problems for the case of non-convex polyhedra. We characterize the types of boundaries of changing aspect that occur for non-convex polyhedra and present upper and lower bounds on the maximum size of the VSP and the aspect graph. We then present an algorithm for constructing the VSP and the aspect graph. We do this first under the orthographic model and then the perspective model. In Section 6 we review related work.

2. Visibility and Occlusion

Visual perception involves acquiring information about the surrounding world from an image of the world. The image is a representation of the array of ambient light available at a viewpoint. When the viewer is a camera, the viewpoint is the center of the lens. The ambient light available to the viewer is a solid angle of light arriving at the viewpoint.

In general, an object in the world is said to be visible in an image whenever light arrives at the viewpoint from the object, so that the object is represented by a region of light intensity values in the image. However, in some cases it is desirable

to define a more restrictive model of visibility, in order to focus attention on a particular aspect of the problem of visual perception.

In this paper we study the change in the topological structure of images of polyhedra, and we use two models of visibility that enable us to study the problem in two slightly different ways. In both models the world contains solid, opaque polyhedra and the edges of the polyhedra are visible in the image. Thus, in the models the image plane is an unbounded plane containing line segments corresponding to the projections of the visible edges of the polyhedra in the world. We assume that the resolution of the image is infinite; nothing is too small to resolve. Note that under these models the image of a polyhedron is more than a silhouette; it is a line drawing with hidden lines removed. Equivalently, the image can be thought of as a set of polygonal regions corresponding to visible faces or parts of faces of the polyhedron.

Since the image plane can only capture light incident upon the viewing plane from some directions (half of them), and since some part of the image plane may be preferred for objects of interest (e.g. the fovea), we also speak of a *viewing direction*. The viewing direction or line of sight is the direction in which the camera is pointed, so that objects in the viewing direction appear at the preferred point of the image. We also assume that the camera has a fixed orientation with respect to the viewing direction, so that objects generally appear upright.

For some problems it is sufficient to assume that the viewpoint is a large distance away from the object of interest and that the whole object is visible in the image. For example, in object recognition it may be reasonable to assume that the whole object to be recognized appears in the image. It would not do for the viewpoint to be inside the object to be recognized or for the viewpoint to be so close to the object that only one face is visible. For other problems the space of possible viewpoints must range over the whole world. For example, in motion planning the visible surfaces represent the boundaries of free space in any direction. In useful motion planning problems, obstacles may be on all sides of the object to be moved. The object may even be inside a workspace.

Therefore we use two viewing models. The first model is designed to handle the first case, in which the whole object is in front of the viewer in any image. In order to model this case, we restrict viewpoints to an infinite sphere containing the object and use orthographic projection in forming the image. Thus the whole object is in front of the viewer from any viewpoint. We call this the orthographic model. It is a useful model because it represents a restriction on visibility that makes some representations smaller. In the second model we allow the viewpoint to be any point in the world. In this case it is natural to use perspective projection in forming the

image from a viewpoint. This model is intended to resemble natural vision, in which objects can be all around the viewer, and the viewer can even be inside objects, such as houses.

The directions of incident ambient light represented in the image plane are those directions more than 90° away from the viewing direction (since the viewing direction is *away* from the viewer but the visible light is *toward* the viewer). In problems such as motion planning, visibility information from all directions is of interest, since the visible surfaces represent the boundaries of free space in any direction. Visibility from all directions can be represented by two image planes from opposite viewing directions along with points at infinity in one of the planes or by an image *sphere* representing all viewing directions. For the problem of looking at an object and recognizing it or perceiving its shape, however, a section of a single image plane is sufficient. We assume that the viewer is interested in a particular object or small part of the world and “looks at” the object.

In the perspective model, viewpoint space is \mathbb{R}^3 and the viewing direction is the direction from the viewpoint to the origin. Thus, the image formation process can be modelled mathematically as a perspective projection of the visible edges of the object onto the image plane, in such a way that the origin of the world projects onto the origin of the image plane. The object of interest will project onto a region of pixels near the origin of the image.

In the orthographic model, we need only represent the *direction* of the viewpoint, rather than its location in the world. In this model viewpoint space is the 2-D space of directions from the origin. We can represent these directions as points on a unit sphere, with a point corresponding to the direction from that point to the origin. We will speak of points on the unit sphere and viewing directions interchangeably. We will denote viewing directions by (θ, ϕ) (see Figure 1). That is, if the point $(0,0,1)$ corresponds to the viewing direction “straight ahead” (i.e. some reference viewing direction) then that point rotated by θ counterclockwise about the y -axis and then by ϕ counterclockwise about the x -axis corresponds to the viewing direction (θ, ϕ) . (In Figure 1, we show a negative ϕ and positive θ rotation).

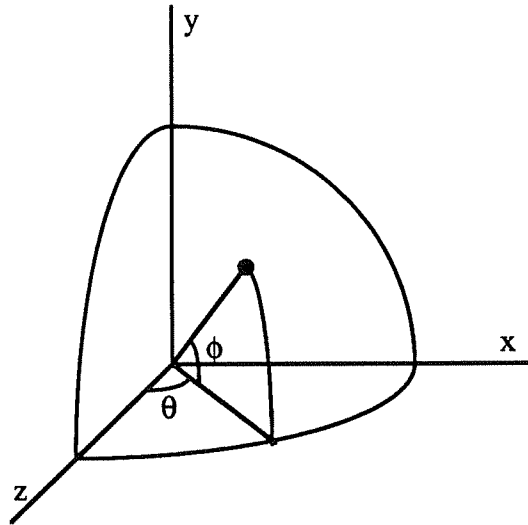


Figure 1. The point (θ, ϕ) on the unit sphere.

Occlusion is the general term for the hiding of some feature in the world from some viewpoint. That is, when the view of something is blocked from a particular viewpoint, we say that it is occluded. *Occlusion* can also be used as a verb, referring to the change from visible to invisible of some feature. These notions of occlusion are object-centered in the sense that they concern whether a feature of an object is visible. There is also an image-centered notion of occlusion, concerning the change in the image resulting from occlusion. For example, from some viewpoint, a part of a face may be visible in a region of the image. As the viewpoint changes, the region may shrink and disappear as that part of the face becomes occluded.

In this paper we are concerned with the changes in occlusion that bring about a topological change in the image. A topological change in the image results when a region appears or disappears, or when regions split or merge. For example, the merging of two regions of the image results in a topologically-distinct image (see Figure 2).

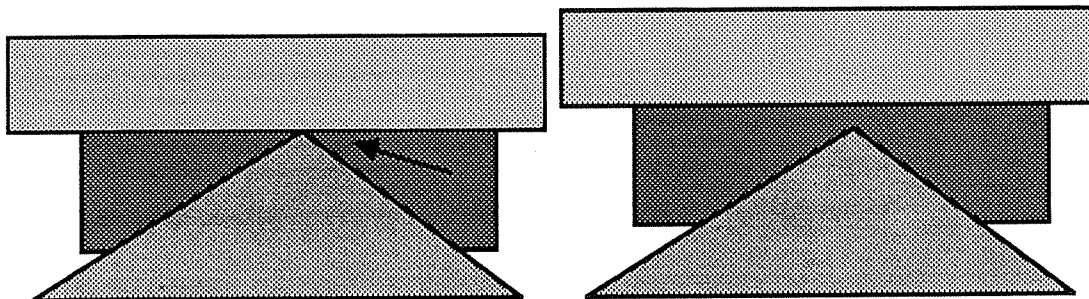


Figure 2. The merging of two regions in the image is a topological change.

However, the change in a region of the image from a 4-sided polygon to a 5-sided polygon does not by itself result in a topologically distinct image (see Figure 3).

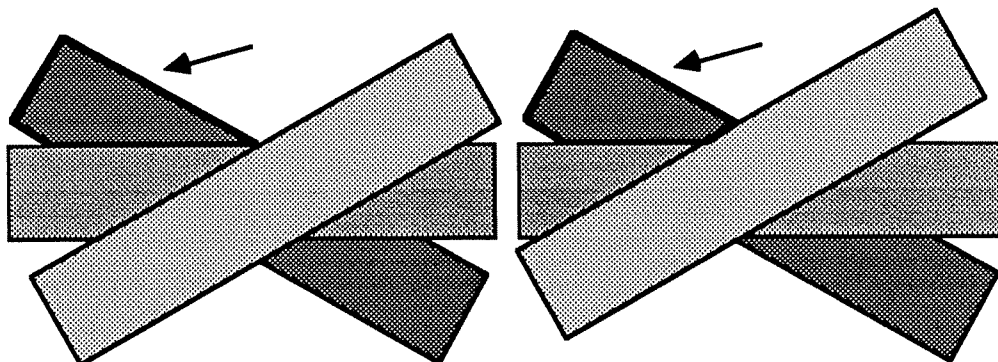


Figure 3. A change in the number of sides of a polygon in the image is not a topological change.

The changes in occlusion that result in a topological change in the image we call *events*. These events are the sort of occlusion we deal with in this paper. That is, we are interested in the ways in which the topological structure of the image of polyhedra changes with changing viewpoint, through the accretion, deletion, splitting, and merging of regions in the image. We characterize the way that the structure of the image changes by studying the viewpoints at which such changes occurs. These viewpoints are interesting because they are the boundaries in viewpoint space of volumes of viewpoints from which the world has the same topological appearance.

Figure 3 shows an event that does not result in two images that are topologically distinct. However, the images *are* “linearly distinct,” in the sense that not every edge of the second can be obtained by a linear transformation of an edge of the first. By also considering events such as that shown in Figure 3, we could modify our work below from the study of topologically-distinct images to the study of linearly-distinct images.

3. The Viewpoint Space Partition and the Aspect Graph

From any viewpoint a polyhedron has a particular appearance, and varying the viewpoint slightly does not generally change the topology of the image. Thus, there are “stable views” or *aspects* for a polyhedron—views of the polyhedron with the same topological image structure over a region of viewpoints. We call a maximal connected region of viewpoints from which the views of the object are topologically

equivalent a *maximal viewing region of constant aspect* or *viewing region* for short. Some changes in viewpoint result in a change in the topology of the image, however. These changes pass from one viewing region to another.

We can define a viewing region more formally as a set of viewpoints that are equivalent under an equivalence relation defined as follows. For two viewpoints v_1 and v_2 , we say that $v_1 \approx v_2$ whenever there is a path of viewpoints from v_1 to v_2 such that from every viewpoint along the path (including v_1 and v_2) the aspect is constant, i.e. the image has the same topological structure. \approx is an equivalence relation since it is reflexive, symmetric, and transitive. Therefore it partitions viewpoint space into equivalence classes consisting of regions or volumes of constant aspect, i.e. viewing regions.

We call this partition of viewpoint space the *viewpoint space partition* or *VSP*. In the orthographic case it consists of a partition of the sphere, and in the perspective case it consists of a partition of \mathbb{R}^3 . The boundary points of the VSP are points from which an arbitrarily small change in viewpoint causes a change in the topology of the image. Figure 4 shows a tetrahedron and Figure 5 shows the structure of the viewpoint space partition generated by the tetrahedron under orthographic projection. The viewpoint space partition is flattened out onto a plane, and the shapes of the regions are not preserved, but the topological structure is preserved.

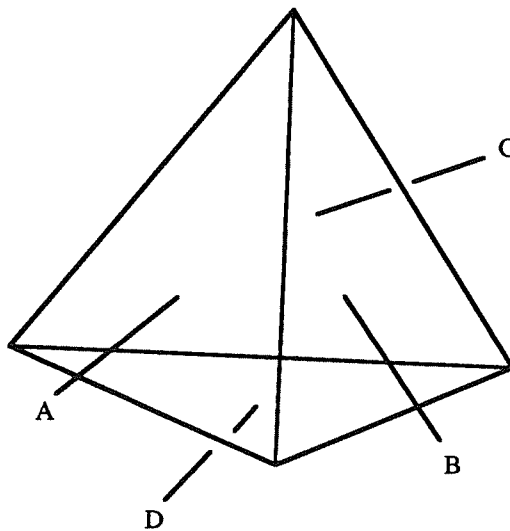


Figure 4. A tetrahedron.

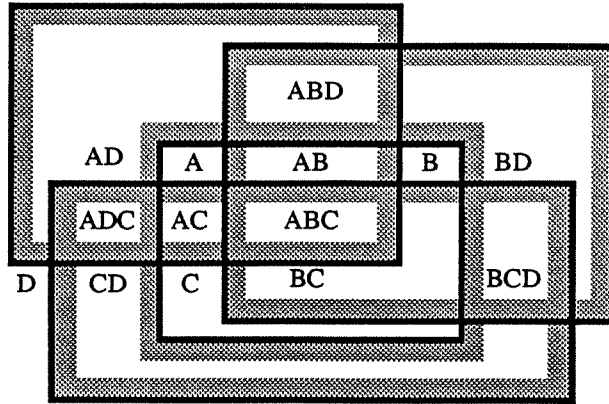


Figure 5. The viewpoint space partition for a tetrahedron.

The shaded side of a boundary represents the inside of a region. Each region is labelled by the faces visible from viewing directions in that region. Since a face is not visible edge-on, the regions are open and their complements are closed.

By definition, the regions of viewpoints of the VSP contain viewpoints in which the topology of the image is constant. Some differences in the image at different viewpoints are still possible, however. No image regions can appear, disappear, split, or merge, but they can change in shape. In fact, polygonal regions of the image can change in the number of edges bounding the region without changing the topology of the image, and the structure of image features such as T-junctions can change without changing the topology of the image. Characterizing images as “topologically equivalent” is not as strong as characterizing them as, say, “linearly equivalent,” where images are related by a linear transformation.

The definition of the VSP does not include labels for the regions. They are included in Figure 5 to aid understanding the figure. If one wishes to label the regions with the corresponding aspects, labeling the regions with the names of the visible faces is sufficient in the convex case, as we have done in Figure 5. It is not sufficient in the general case, however, since images with the same sets of visible faces can have different topological structure. In that case we can label a region by adding a pointer to an image of the object from a viewpoint in that region. Alternatively, we could label the boundaries with a description of the change that occurs in the image.

We define the dual of the region-boundary structure of the VSP to be the *aspect graph*. That is, for every region of the VSP there is a vertex in the aspect graph, and vertices of the aspect graph are connected by edges whenever the corresponding regions of the VSP are adjacent. Thus, the aspect graph has a vertex for every aspect, and vertices for adjacent aspects are connected. Figure 6 shows the aspect graph for the tetrahedron of Figure 4 overlaid on the VSP, and Figure 7 shows

the aspect graph by itself. In this case the aspect graph is the same under the orthographic and perspective models.

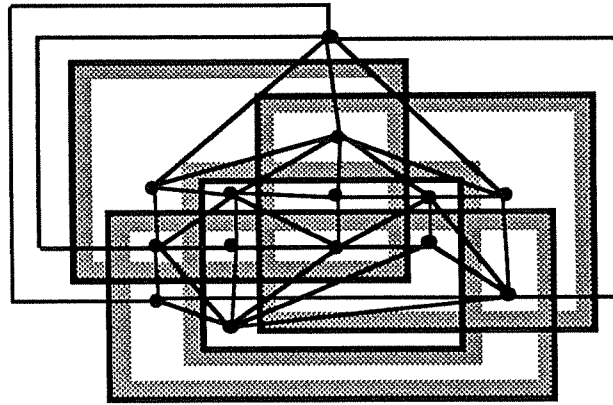


Figure 6. The aspect graph is the dual of the VSP.

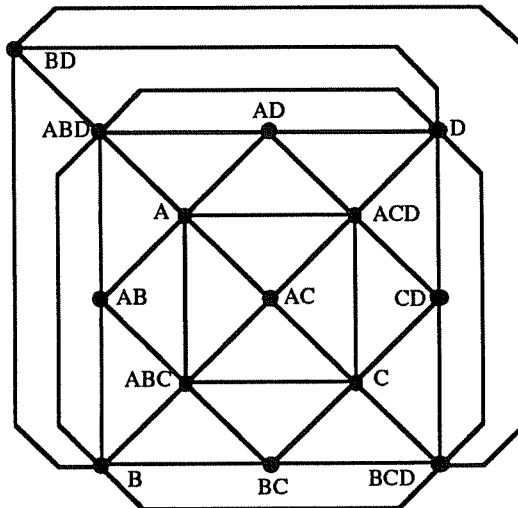


Figure 7. The aspect graph for a tetrahedron.

In Figure 7 the vertices of the aspect graph are labelled by the names of the faces visible from the corresponding aspect. Again, the labels are not included in the definition of the aspect graph. In order to associate the aspect with vertices of the aspect graph, it is sufficient in the convex case to label each vertex with the names of the faces visible from a viewpoint in the corresponding region. In the non-convex case this is not sufficient, and one can label the vertices with a corresponding image of the object or the edges with a description of the change that occurs.

Note that under our definition of the VSP and the aspect graph two different vertices of the aspect graph can correspond to aspects that are topologically equal. This happens when the appearance of the object is the same from two different, unconnected maximal regions of viewpoint space. Furthermore, the appearance from the two aspects can be the same in two different senses. In the more restrictive

sense, the same set of faces may be visible and have the same topology from a different region of viewpoint space. In the less restrictive sense, a completely different set of faces may have the same topological appearance. These are also reasonable notions of the identity of aspects, but they differ from our notion.

Since the aspect graph has a vertex for every topologically-distinct view of a polyhedron, it is a tool for characterizing the distinct views of an object. It has been used in computer vision for that purpose. However, the VSP represents essentially the same information with the addition of the actual boundaries in viewpoint space of the viewing regions, and the space it requires is only a constant factor larger. Since the VSP represents viewing regions and their boundaries, it is a more appropriate tool for most applications that are concerned with the particular viewpoints associated with each aspect.

The observant reader may have wondered why in Figure 7 vertices A and ABC are connected, whereas vertices AB and AC are not. After all, the corresponding regions of viewpoint space seem to have the same relationship to each other in both cases: paths of viewpoints connecting a viewpoint in one region with a viewpoint in the other all pass through a single point P (see Figure 8).

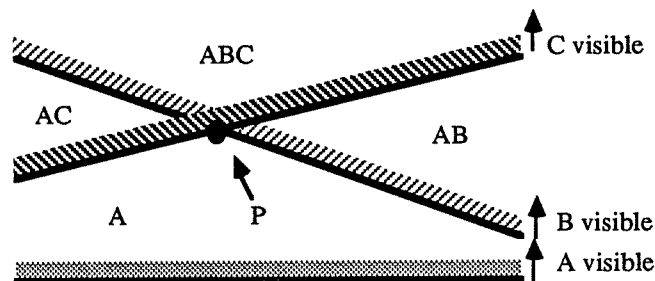


Figure 8. The boundary between four regions of viewpoint space.

The answer is a result of the fact that faces are not visible edge-on. The regions of viewpoint space in Figure 8 are shown with shading on open boundaries. Thus, the point P is in region A but not in regions AB, AC, or ABC. Therefore it is possible to find a path of viewpoints from a point in region A to a point in region ABC which doesn't go through AB and AC, but it is not possible to find such a path between AB and AC.

J. Koenderink and A. van Doorn [1976] first introduced the notion of a stable view or aspect, but their definition differs slightly from ours. They assume that

objects are smooth, so they do not have any edges, unlike polyhedra. They study the topological structure of the singularities of the visual mapping, which occur at image points corresponding to world points that are tangent to or normal to the line of sight. If they had defined a model of visibility, they might have said that the only things that appear in an image are occluding contours and points at which the corresponding surface in the world is normal to the line of sight, i.e. local minima and maxima of distance.

In our model, we allow edges that are not occluding edges, but we require that faces be flat. We do not consider local minima and maxima of distance. The difference between the viewing models is illustrated in Figure 9, which shows an image of a cube under our viewing model and that of Koenderink and van Doorn.

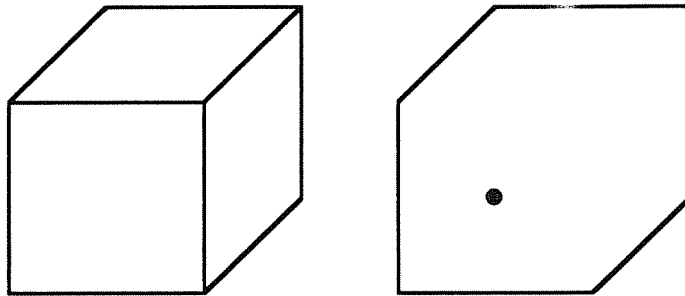


Figure 9. An image of a cube under our viewing model and that of Koenderink and van Doorn.

In our model all of the visible edges of the polyhedra appear as line segments in the image. In Koenderink and van Doorn's model, sharp edges are not allowed, so they are considered to have been smoothed. Only occluding contours and local minima and maxima of distance appear in an image. Koenderink and van Doorn define a change in aspect to be a change in the topological structure of these singularities. We define a change in aspect to be a change in the topological structure of the regions in the image formed by the projection of the edges of the object.

Koenderink and van Doorn [1979] first introduced the aspect graph (which they call the *visual potential graph*) in order to formalize the ideas of the topologically-distinct views of an object. Their definition of the aspect graph differs from ours in the same way that their definition of aspect differs.

4. Convex Polyhedra

In this section we consider the VSP and aspect graph for a single convex polyhedron. The restriction to convex polyhedra greatly simplifies the problem because a face is always visible from viewpoints "in front of" it; no face ever occludes

another face. The only way that a face may become invisible with changing viewpoint is by “turning away from” the viewer or “going below the horizon.” That is, when the viewpoint passes through the plane containing the face, the face becomes invisible (or visible) due to occlusion of the face by the rest of the object. Also, there cannot be two vertices in the aspect graph for the same set of faces since if the same set of faces is visible from two viewpoints, that set is also visible from all viewpoints along the shortest path (geodesic) in viewpoint space connecting them. Thus, in the convex case a change in aspect and a change in the visibility of some face of the polyhedron are equivalent events.

4.1. Orthographic Case

Under the orthographic viewing model, viewpoint space is a 2-D spherical space of directions. A face of a convex polyhedron is visible in any viewing direction “pointing toward” the front of that face. That is, if \mathbf{d} is a viewing direction and \mathbf{n} is the outward normal of the face, that face is visible whenever $\mathbf{n} \cdot \mathbf{d} < 0$, i.e. the angle between \mathbf{n} and \mathbf{d} is greater than 180° . Therefore the boundary between the regions of the sphere of viewing directions where the face is visible and where it is invisible is the great circle defined by $\{\mathbf{d} : \mathbf{n} \cdot \mathbf{d} = 0\}$. This is the intersection of the sphere with a plane containing the origin and having normal \mathbf{n} .

4.1.1. An Upper Bound

The great circle corresponding to a face is the boundary on the sphere of viewpoints between the region (hemisphere) where the face is visible and the region where it is invisible. For a polyhedron with \mathbf{n} faces, the corresponding \mathbf{n} great circles therefore partition viewpoint space into a subdivision in which the same set of faces is visible from all of the viewpoints in each region. That is, the \mathbf{n} great circles partition viewpoint space into the VSP.

Assuming that no two faces have antiparallel normals, each pair of great circles intersects in exactly two points. In that case the total number of intersection points is at most $2 \mathbf{n} (\mathbf{n}-1) = O(\mathbf{n}^2)$. (The number can be less than $2 \mathbf{n} (\mathbf{n}-1)$ if several great circles intersect in the same point.) If two faces of the polyhedron are parallel, the corresponding great circles coincide, and the number of vertices in the VSP is reduced. Thus the VSP has $O(\mathbf{n}^2)$ vertices. Since a subdivision of a 2-D space is by definition planar and a planar graph has a linear number of edges and regions, the size of the VSP is therefore $O(\mathbf{n}^2)$. The aspect graph is the dual of the VSP, so it is also planar and also has size $O(\mathbf{n}^2)$.

4.1.2. A Lower Bound

The lower bound on the maximum size of the VSP and the aspect graph for a convex polyhedron of size n under orthographic projection is $\Omega(n^2)$. We prove this by exhibiting a class of such polyhedra. Consider a band of m square faces arranged around a circle (see Figure 10). In Figure 11 we show two such bands arranged orthogonally around a sphere, with additional faces added to form a convex polyhedron.

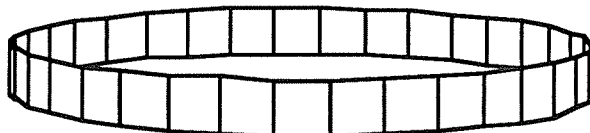


Figure 10. A band of m sides.

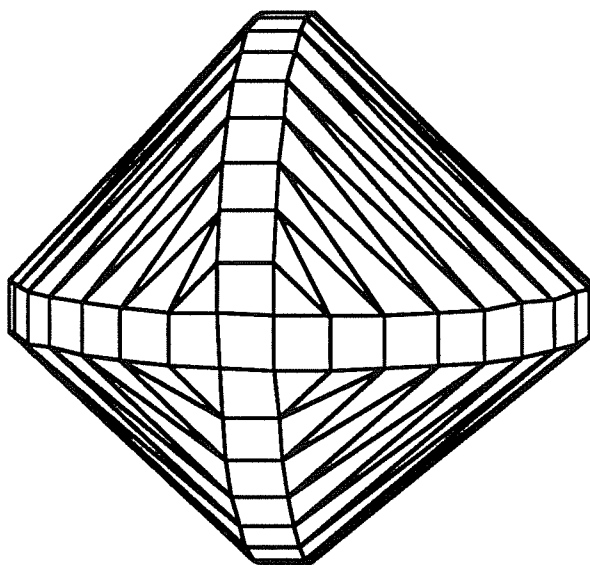


Figure 11. A convex polyhedron with two orthogonal bands.

Select one of the faces visible in Figure 11 from each band, excluding the face that is common to both bands. The great circles bounding the visibility regions on the sphere of the two selected faces have two intersection points, and these points are vertices of the VSP. Each pair of faces thus selected has a unique pair of intersection points, so there are $\Omega(m^2)$ vertices in the VSP. The faces added to the bands to form a polyhedron have total size $O(m)$, so if the polyhedron has size n , its VSP has size $\Omega(n^2)$. Since the aspect graph is the dual of the VSP, it also has size $\Omega(n^2)$.

In fact, a polyhedron would have to be highly degenerate to have a VSP of size less than $\Theta(n^2)$. For example, any class of polyhedra that has bounded vertex degree and bounded face degree has a VSP of size $\Omega(n^2)$. This is because the only way to reduce the size of the VSP is to have the intersection points of the great circles

coincide, and if the number of edges around each face and the number of faces around each vertex is bounded, then the number of great circles that can intersect in a single point is bounded. An example of a highly degenerate class of polyhedra is the class of n -sided approximations to a cylinder; for polyhedra in this class the VSP has size $O(n)$.

4.1.3. An Algorithm

The great circles defined by the face normals subdivide the sphere of viewing directions into the VSP. Therefore to give an algorithm for constructing the VSP it is only necessary to show how to merge these great circles into one subdivision data structure. This data structure represents regions, edges, and vertices of the subdivision and adjacencies of the various items. Edelsbrunner *et al.* [1986] present an algorithm for constructing an arrangement (subdivision) formed by n lines in the plane in $O(n^2)$ time. The algorithm for constructing the arrangement can be used for constructing a subdivision formed by great circles on the sphere since they behave very much like lines on a plane. In fact, the great circles can be projected onto lines on two parallel planes and the arrangements constructed in the planes. This approach is similar to that of McKenna and Seidel [1985], in which they construct minimal or maximal shadows of a convex polytope generated by a point light source at infinity.

The algorithm for constructing the arrangement works by starting with an empty subdivision and adding one line at a time. Adding a line l to the subdivision involves finding an intersection point of l with a line already in the subdivision. Both lines are split at the intersection point: the nodes in the subdivision for the edges that intersect are duplicated, and one endpoint of each of the four edges is set to the intersection point. l crosses a region of the subdivision in both directions from that point. The edges of one of the regions are tested for intersection with l , and the intersection point thus found is added to the subdivision. The region just crossed by l is split into two regions, one on either side of the new edge on l . This process is continued in both directions from the first intersection point of l with an edge of the subdivision, until all of l has been inserted into the subdivision.

This algorithm constructs the arrangement defined by m lines in a plane in $O(m^2)$ time. Thus the VSP for a convex polyhedron can be constructed in $O(n^2)$ time, which is worst-case optimal. In fact, the algorithm is optimal for all but highly degenerate polyhedra, since only in that case does the VSP have size smaller than $\Theta(n^2)$.

The aspect graph is the dual of the region-edge structure of the VSP (see Figure 6). That is, the aspect graph has a vertex for every region of the VSP and edges connecting vertices corresponding to adjacent regions. Thus the aspect graph can be constructed by copying the region-edge subgraph of the VSP, changing the regions to vertices, and changing the edges separating regions to edges connecting adjacent vertices. This can be done in linear time in the size of the VSP, so constructing the aspect graph can be done in $O(n^2)$ time.

4.2. Perspective Case

Under the perspective model the viewpoint is not restricted, so the viewpoint space is R^3 . Also, objects are not restricted to being in front of the viewpoint; they can be in front of or behind the viewer. The viewpoint can even be inside a solid object, but in that case nothing is visible since only the front side of a face is visible. Thus the image changes as the viewpoint moves in any one of three independent directions rather than two. Therefore the VSP and aspect graph are much larger in this case. The reason for the larger size is the higher dimensionality of viewpoint space and not the change from orthographic to perspective projection.

The only kind of occlusion that occurs in this case is a face turning away from the viewer, as in the orthographic case. However, under the perspective model a face is visible from all points in front of the plane spanned by that face (rather than from all viewing directions pointing toward that face). Therefore viewpoint space is cut by n planes for a polyhedron with n faces, and the resulting subdivision is the VSP. The aspect graph is the dual of the cell-face structure of the VSP.

In fact, the aspect graph for a polyhedron under perspective projection contains as a subgraph the aspect graph for that polyhedron under orthographic projection. This is true because in the limit of increasingly large spheres centered on the polyhedron, the planes spanned by the faces of the polyhedron cut the sphere in the same great circles as in the orthographic case.

4.2.1. An Upper Bound

The VSP is the subdivision of viewpoint space generated by the n planes corresponding to the n faces of the polyhedron. Since the intersection of three general planes is a point, there can be at most $O(n^3)$ intersection points of n planes. In order to get the largest possible VSP, we can assume that every set of three planes

intersects in a distinct point. These intersection points are vertices of the VSP, so the VSP has $O(n^3)$ vertices.

At worst, each plane is intersected by all the others, resulting in $O(n)$ lines subdividing each plane. Thus each plane is cut into a subdivision of $O(n^2)$ vertices, edges, and regions. Therefore among all n planes there are $O(n^3)$ vertices, edges, and faces. A cell of the VSP is bounded by only four faces in the worst case, so the VSP has at most $O(n^3)$ vertices, edges, faces, and cells. The aspect graph is the dual of the cell-face structure of the VSP, so its size is also $O(n^3)$.

4.2.2. A Lower Bound

The lower bound example of the orthographic case (see Figure 11) is also a lower bound in the perspective case. Consider the faces of the two bands in any octant, i.e. one quarter of each band in the same octant (see Figure 12).

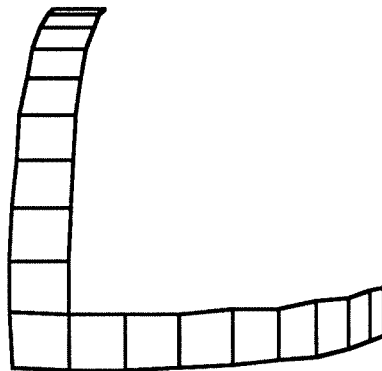


Figure 12. Two quarter-bands.

Any three distinct faces from these band-quarters (two from one band and one from the other) determine a distinct vertex in the VSP since they have a distinct intersection point. There are $\Omega(n)$ faces in each band quarter, so the VSP has $\Omega(n^3)$ vertices. The VSP must also have $\Omega(n^3)$ cells, so the aspect graph also has size $\Omega(n^3)$. As in the orthographic case, a class of polyhedra would have to be highly degenerate to have size less than $\Theta(n^3)$. For example, any class of convex polyhedra with bounded vertex degree and face degree has a VSP of size $\Omega(n^3)$.

4.2.3. An Algorithm

The n planes corresponding to the n faces of a polyhedron partition space into the VSP. In order to construct the VSP, we must find all of the intersections of the planes and construct the data structure for the VSP by finding its cells, faces, edges,

and vertices and linking incident faces. The algorithm of Edelsbrunner *et al.* [1986] for constructing arrangements of hyperplanes does the job in optimal, $O(n^3)$ time. The algorithm builds the subdivision one plane at a time, as in the orthographic case. A plane \mathbf{p} can be added to the subdivision by finding the line of intersection of \mathbf{p} and each plane already in the subdivision. The line of intersection of \mathbf{p} and another plane \mathbf{q} already in the subdivision is then inserted into the 2-D subdivisions on \mathbf{p} and \mathbf{q} using the 2-D subdivision algorithm of the orthographic case. After the 2-D subdivisions have been constructed, the cells of the 3-D subdivision can be added by traversing the subdivision data structure.

The aspect graph is the dual of the cell-face graph of the VSP. It can be constructed from the data structure for the VSP by copying the VSP, changing cells to vertices and faces between cells to edges joining them, and deleting everything else. This algorithm requires linear time in the size of the VSP, so the aspect graph can be constructed in $O(n^3)$ time.

5. Non-Convex Polyhedra

In the convex case, the only sort of event that can change the aspect is the horizon effect. In the non-convex case, faces can occlude other faces, causing several other types of events. Thus, in order to construct the VSP and the aspect graph in the non-convex case we must characterize all the ways in which aspect can change. Since there are more modes of changing aspect, the size of the VSP and the aspect graph for non-convex polyhedra can be much larger. In the following sections we list the kinds of events that can occur and the corresponding boundaries of regions of constant aspect that result in viewpoint space.

5.1. Orthographic Case

In the orthographic case, the VSP is a subdivision of the sphere of viewpoints, and the aspect graph is its dual. We first list the ways that the aspect can change in the non-convex case and the kinds of boundaries in viewpoint space that these changes generate. We then present upper and lower bounds on the maximum size of the VSP and aspect graph in this case and present an algorithm for their construction.

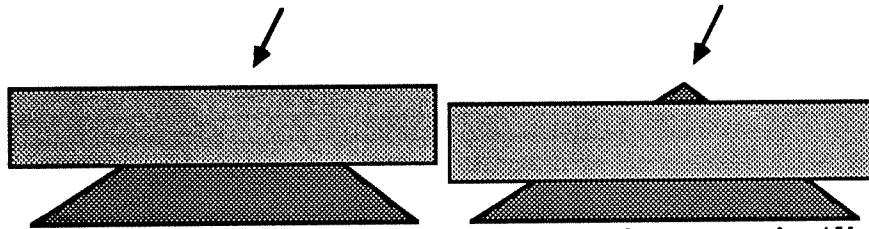
5.1.1. Occlusion under Orthographic Projection

In order to compute the regions of constant aspect in viewpoint space, we compute the boundaries of occlusion, that is, the boundaries of these regions. We define a boundary viewpoint to be a viewing direction from which an arbitrarily small change in viewpoint suffices to change the topology of the image. In this section we characterize several kinds of boundaries in viewpoint space at which regions of the image appear, disappear, split, or merge, and we argue that these are the only kinds of boundaries of regions of constant aspect that occur for polyhedra. The types of boundaries are horizon boundaries (which were discussed in the convex case), edge-vertex boundaries, and edge-edge-edge boundaries.

5.1.1.1. Edge-Vertex Occlusion Boundaries

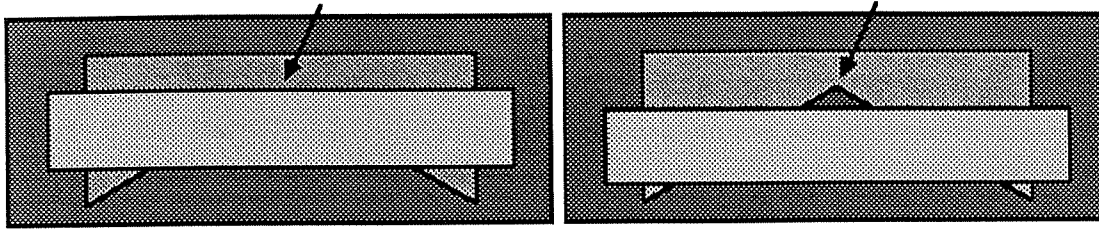
Edge-vertex occlusion boundaries occur at viewpoints from which a vertex appears on an edge in an image. These events are boundaries of occlusion since a small change in viewpoint can change the topological structure of the image. The image can change either by the accretion or deletion of a region in the image or by the splitting or merging of regions.

The accretion or deletion of a region occurs when a very small change in viewpoint makes a face visible or invisible at an image point. One way that this can happen is illustrated in Figure 13. In this case a face is occluded at a point in an image by being behind another face (see Figure 13a). A small change in viewpoint then suffices to make the face visible at that point, creating a new region in the image (see Figure 13b). Such viewpoints are boundaries of regions of constant aspect since an arbitrarily small change in viewpoint suffices to cause a new region to appear in the image.



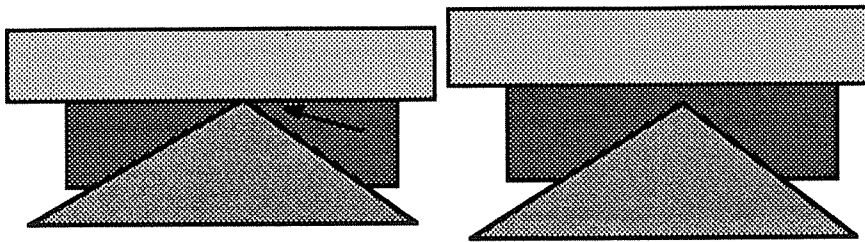
Figures 13a and 13b. Edge-vertex occlusion type 1 (region appearing/disappearing).

A region can also disappear or appear if the edges meeting at the vertex form a concave corner (see Figure 14a). In this case a small change in viewing direction can cause a new region to be formed in the image in the manner illustrated in Figure 14b.



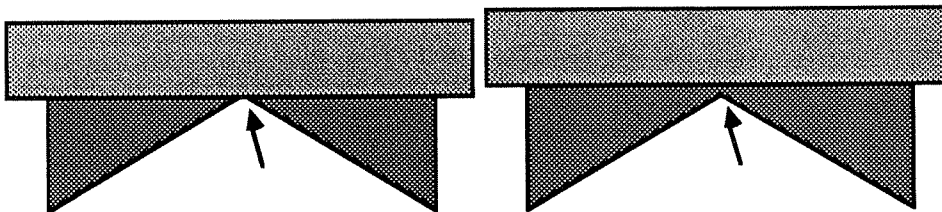
Figures 14a and 14b. Edge-vertex occlusion type 2 (region appearing/disappearing).

The image can also change in such a way that two regions of the image merge or a region splits into two. One way that this occurs is illustrated in Figure 15. If a convex corner lies on an edge in an image (Figure 15a), an arbitrarily small change in viewpoint can cause the regions on either side of the corner to merge (Figure 15b).



Figures 15a and 15b. Edge-vertex occlusion type 3 (regions merging/splitting).

Regions can merge in another way, illustrated in Figure 16. If a concave corner lies on an edge in an image (Figure 16a), an arbitrarily small change in viewpoint can cause the regions on either side of the corner to merge (Figure 16b).



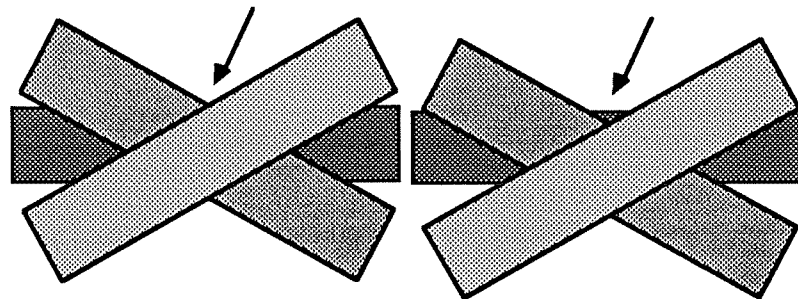
Figures 16a and 16b. Edge-vertex occlusion type 4 (regions merging/splitting).

Such viewpoints are all on boundaries of occlusion, since in all cases a small change in viewpoint causes the image to change, by regions appearing, disappearing, splitting, or merging. Boundaries in viewpoint space of occlusion of this type occur at viewpoints such that a vertex of one face is directly behind an edge of another face in the image. Therefore the only candidate viewpoints for such boundaries are viewing directions parallel to the plane containing the point and the edge. That is, for a point \mathbf{p} and an edge from \mathbf{p}_1 to \mathbf{p}_2 , the candidate viewpoints are the points on the arc of the great circle with the normal $(\mathbf{p}_1 - \mathbf{p}) \times (\mathbf{p}_2 - \mathbf{p}_1)$. The boundaries of the arc are the points of intersection with the lines $(\mathbf{p}, \mathbf{p}_1)$ and $(\mathbf{p}, \mathbf{p}_2)$.

Since there are $O(n)$ vertices and $O(n)$ edges in a polyhedron or polyhedral scene, there are $O(n^2)$ edge-vertex pairs, so this type of occlusion is responsible for generating at most $O(n^2)$ occlusion boundaries (arcs of great circles) on the sphere of viewpoints.

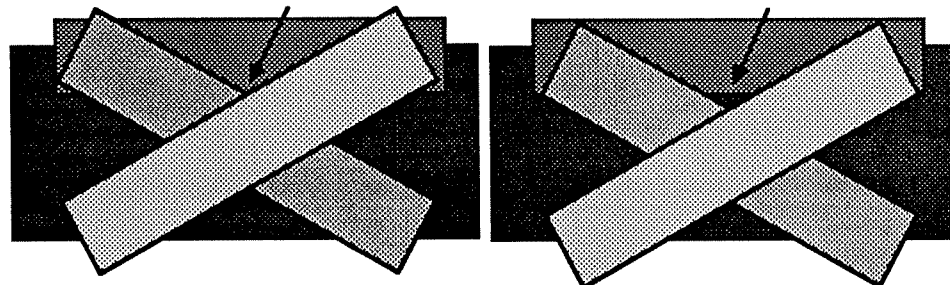
5.1.1.2. Edge-Edge-Edge Occlusion

Another type of occlusion boundary can occur at viewpoints where three unconnected object edges appear to intersect in a single image point. We call this edge-edge-edge occlusion. One way that this can occur is illustrated in Figure 17. Two faces that are not directly connected can occlude the view of a third unconnected face at an image point (see Figure 17a). Such viewpoints are aspect boundaries since an arbitrarily small change in viewpoint suffices to make the third face appear at that image point, i.e. to create a new region in the image (see Figure 17b).



Figures 17a and 17b. Edge-edge-edge occlusion type 1.

Edge-edge-edge occlusion can occur in a slightly different way as well, as illustrated in Figure 18. A viewpoint where three edges appear to intersect in a single point can occlude the view of a fourth face at that point (see Figure 18a). Such a viewpoint is a boundary of occlusion because an arbitrarily small change in viewpoint can suffice to make the fourth face visible at that point (see Figure 18b).



Figures 18a and 18b. Edge-edge-edge occlusion type 2.

These are the only kinds of edge-edge-edge occlusion that occur. Regions cannot split or merge in the case of edge-edge-edge occlusion because the merging of

regions requires that the occluding faces separate. That implies a that a vertex and an edge appear to intersect at that image point or that parallel lines appear to overlap in the image. The latter case is handled as two instances of vertex-edge occlusion.

The boundaries in viewpoint space generated by edge-edge-edge occlusion occur at viewpoints where three unconnected edges appear to intersect in a single point. In order to find the viewing directions in which three edges appear to intersect in a single point, let the endpoints of the three edges be \mathbf{p}_{11} , \mathbf{p}_{12} , \mathbf{p}_{21} , \mathbf{p}_{22} , \mathbf{p}_{31} , and \mathbf{p}_{32} respectively, and pick a point on one of the edges, say

$$\mathbf{p} = \mathbf{p}_{11} + t(\mathbf{p}_{12} - \mathbf{p}_{11}), \quad 0 \leq t \leq 1 \quad (1)$$

The other two edges appear to intersect in some viewing direction \mathbf{d} at \mathbf{p} whenever the planes defined by \mathbf{p} and each of the other two edges intersect in a line with direction \mathbf{d} . That is, the other two edges appear to intersect at \mathbf{p} from a viewing direction given by the intersection of the planes defined by \mathbf{p} , \mathbf{p}_{21} , \mathbf{p}_{22} , and \mathbf{p} , \mathbf{p}_{31} , \mathbf{p}_{32} . Normals to these planes are given by

$$\begin{aligned} &(\mathbf{p} - \mathbf{p}_{21}) \times (\mathbf{p}_{21} - \mathbf{p}_{22}) \\ &(\mathbf{p} - \mathbf{p}_{31}) \times (\mathbf{p}_{31} - \mathbf{p}_{32}) \end{aligned}$$

and the intersection of these two planes has the direction

$$\mathbf{d} = [(\mathbf{p} - \mathbf{p}_{21}) \times (\mathbf{p}_{21} - \mathbf{p}_{22})] \times [(\mathbf{p} - \mathbf{p}_{31}) \times (\mathbf{p}_{31} - \mathbf{p}_{32})] \quad (2)$$

Therefore \mathbf{d} is a viewing direction in which the three edges appear to intersect at \mathbf{p} . Note that \mathbf{d} is a quadratic function of the parameter t . Thus viewing directions do not form an arc of a great circle in viewpoint space—rather, they form a quadratic curve. In Eq. (2) \mathbf{d} is given in Cartesian coordinates. In order to transform it to the (θ, ϕ) notation, we can use a Cartesian-to-spherical coordinate transformation:

$$\theta = \tan^{-1}(d_x/d_z) \quad (3)$$

$$\phi = -\sin^{-1}(d_y/\sqrt{d_x^2 + d_y^2 + d_z^2}) \quad (4)$$

There are $O(\mathbf{n}^3)$ sets of three edges in a polyhedron, so there are $O(\mathbf{n}^3)$ such curves in viewpoint space.

Note that horizon and vertex-edge occlusion boundaries can be considered a special case of edge-edge-edge occlusion. When two of the edges of edge-edge-edge

occlusion meet at a vertex of the object, the result is vertex-edge occlusion. When all three of the edges lie on the same face of the object, the result is that the viewing direction lies in the plane containing the face. In that case the viewing direction is on the boundary of occlusion caused by the face turning away from the viewer.

We claim that the occlusion boundaries listed above represent a complete catalog of events that change the topology of an image. If the topology of the image is to change with a small change in viewpoint, then some region must appear or disappear, or regions must split or merge. If one of those things happens, it must start at some point in the image. The only way that such a change can happen is if at least three edges (or two parallel edges) meet at that point in the image, and we have listed all the unique ways in which three edges can meet in such a way that the topology of the image changes. If more than three edges meet at the point, then either they are degenerate and some subset of three generates the same event, or they are general and hence do not generate an event with viewpoint space extent; i.e., the lines appear to intersect at only a single viewpoint. Parallel edges that overlap in an image can be treated as two vertex-edge pairs.

5.1.2. An Upper Bound

We have seen that the boundary points of occlusion are viewing directions where three object edges appear to intersect in a single point and there are $O(n^3)$ curves of such points. $O(n^3)$ quadratic curves on the sphere intersect in $O(n^6)$ points. The resulting subdivision of viewpoint space is planar, so it also has $O(n^6)$ edges and regions. Therefore the VSP and the aspect graph both have maximum size $O(n^6)$.

5.1.3. A Lower Bound

We have argued that there are $O(n^3)$ curves in viewpoint space potentially bounding regions of visibility and that there are $O(n^6)$ intersection points of these curves, so the VSP has size $O(n^6)$. In fact, these bounds are tight. We show this by presenting an example family of polyhedra with n faces that has an aspect graph (and hence a VSP) of size $\Omega(n^6)$.*

Consider two grids of m strips each, with the strips close together. In front of these grids are two screens, each with m slits. The two screens and grids are

* This example is due to John Canny [1987].

arranged as in Figure 19. Note that the grid edges are not quite parallel to the screen edges.

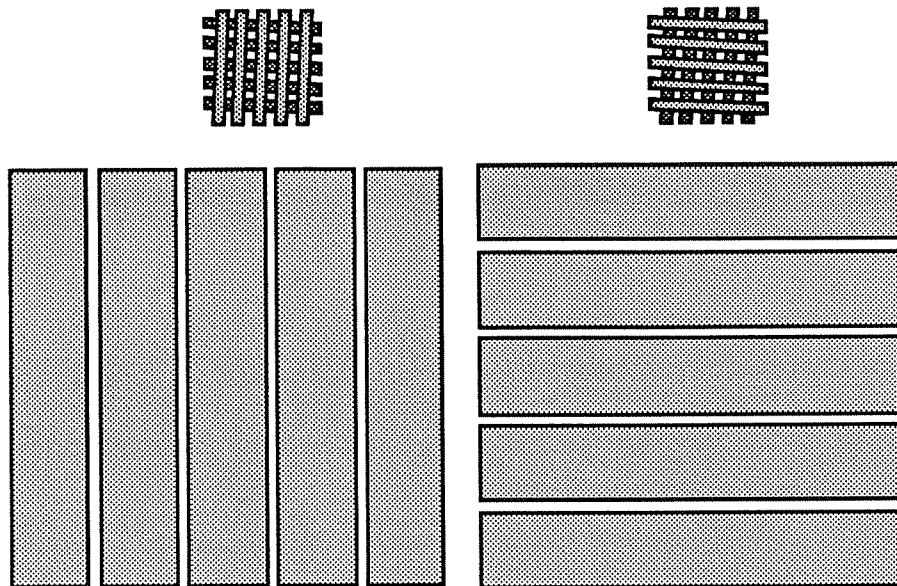


Figure 19. Two grids behind two screens.

In a typical view of the grids behind the screens, parts of the grids are visible through only one slit of each screen. Furthermore, only a small part of the grid is visible through the slit (see Figure 20).

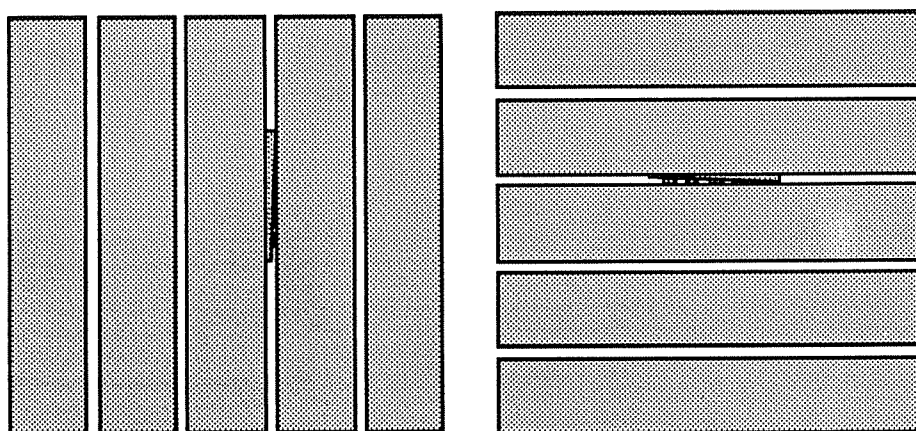


Figure 20. A typical view of the two grids (seen simultaneously).

The part of each grid that is visible behind the vertical screen is a portion of a nearly-vertical grid element and between 0 and m nearly-horizontal grid elements. The view through a slit of the horizontal grid is symmetrical. Figure 21 contains close-up views of the part of the grids visible through the two slits.

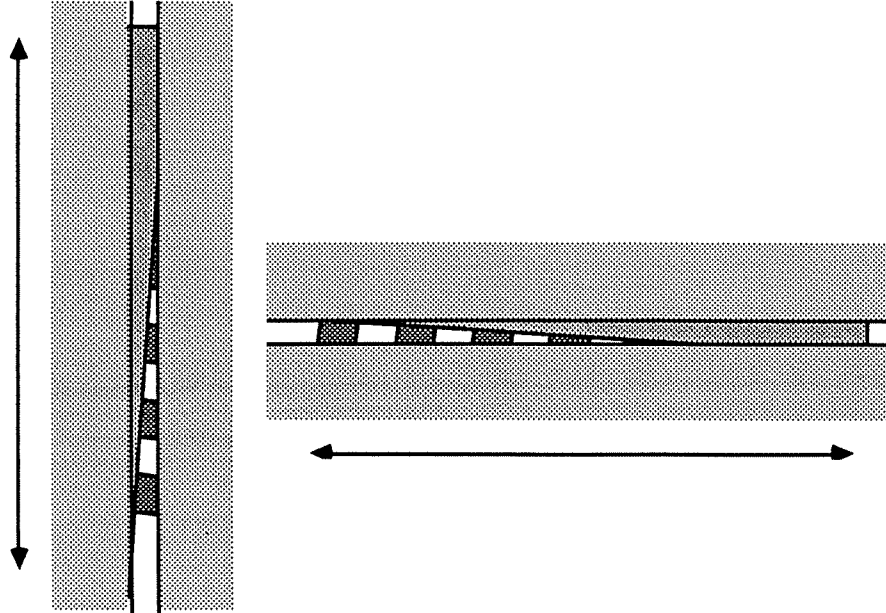


Figure 21. Close-up views through the slits.

Note that changing the viewing direction along the horizontal arrow causes the m nearly horizontal faces to disappear from view in the vertical slit and reappear one by one, generating a new boundary of occlusion in viewpoint space each time. This disappearing and reappearing of the horizontal faces in back occurs m times, so that the view through this slit requires $\Omega(m^2)$ vertices in the aspect graph. There are m vertical slits, requiring $\Omega(m^2)$ boundaries each, for a total of $\Omega(m^3)$ boundaries. Note also that the view through a vertical slit doesn't change when the viewing direction is changed parallel to the vertical arrow. These boundaries are parallel straight lines (arcs of great circles) in viewpoint space.

The views through the horizontal slits exhibit the same behavior: they cause $\Omega(m^3)$ straight boundaries in viewpoint space. However, the boundaries are orthogonal to the other $\Omega(m^3)$ boundaries, so there are $\Omega(m^6)$ intersection points of these boundaries. Therefore the VSP and the aspect graph have size $\Omega(m^6)$.

5.1.4. An Algorithm

The algorithm for constructing the VSP in the case of convex polyhedra is relatively straightforward. The VSP is the partition of the sphere generated by great circles corresponding to the faces of the polyhedron. In the case of non-convex polyhedra this approach does not work. A face may not be visible from all viewpoints in front of it because it is partially occluded from some viewpoints by other faces.

However, the types of boundaries of regions of the VSP (events) have listed above, and the VSP can be computed from them.

Since every boundary of a region of constant aspect in viewpoint space is of the form of the edge-edge-edge boundaries defined above, a naive approach is to find *every* boundary in viewpoint space generated by every set of three object edges, taking those edges in all possible orders. Eq. (1) gives a potential boundary of visibility in viewpoint space for such a set of three edges. We can draw all $O(n^3)$ of these boundaries on the sphere. It remains to find the subdivision of the sphere that they generate.

The subdivision of viewpoint space generated by these boundaries has size $O(n^6)$ and can be constructed in $O(n^6 \log n)$ time using the algorithm given in Section 5.1.4.2. This subdivision is a refinement of the VSP (i.e. has all the vertices and edges of the VSP and more) since the boundaries are only potential boundaries regions of constant aspect; not all of them actually represent changes in visibility. For example, some of the potential events may not be visible in any image. It remains to remove the boundaries that are not actual aspect boundaries.

For any boundary in viewpoint space, we know the edges that gave rise to the boundary and the image point at which the region appeared or disappeared. Thus, computing the faces visible immediately around the image point in question from viewpoints immediately adjacent to the boundary suffices to determine whether the aspect actually changes at that boundary: if the two sets of visible faces are equal, the aspect does not change, and vice versa. We then merge adjacent regions if the same set of faces is visible from both regions. The result is the VSP, and its dual is the aspect graph.

This algorithm can be executed in $O(n^7)$ time, which is somewhat worse than the worst case size of the aspect graph, $O(n^6)$. A more serious problem with this algorithm is that its best case time is $\Omega(n^7)$. For example, the VSP for a convex polyhedron has size $O(n^2)$ but this algorithm would still take time $\Omega(n^7)$. Since the worst case is very large and most aspect graphs are not as large as the worst case, an algorithm with better behavior for simple polyhedra would be much preferable.

The size of the subdivision generated by m curves is $O(m^2)$, so in order to construct the aspect graph more efficiently we compute the actual boundaries of aspect *before* generating the subdivision. Then if the actual number of aspect boundaries is smaller than $\Theta(n^3)$, the runtime of the algorithm will be much less than $\Theta(n^6 \log n)$. In order to do that, we make use of the *aspect representation* or *asp* for the object. The asp represents all of the visual events for a polyhedron or

polyhedral scene, so it is simple to find the aspect boundaries from the asp. The asp is introduced in the next section. For a more detailed description of properties of the asp and algorithms for the asp see [Plantinga and Dyer, 1987a] and [Plantinga and Dyer, 1987b].

5.1.4.1. The Aspect Representation

Three edges of a polyhedron may generate an event that bounds a region of constant aspect in viewpoint space, but in order to determine whether the three edges actually generate an event, we must determine several things. First, we must determine the viewpoints from which the edges appear to intersect in a single point in the image. Then, we must determine whether the edges are arranged in a way that represents a potential event. Finally, we must determine from which of the viewpoints the event is actually visible, and not occluded by other faces.

In order to determine these things, we make use of the aspect representation or asp. The asp is a continuous, viewer-centered representation for polyhedra. It is viewer centered in the sense that it represents the appearance of a polyhedron to a viewer, rather than the volume of space that it fills, and it is continuous in the sense that it represents appearance as a continuous function of viewpoint. Since the asp represents all of the visual events related to the viewing of the polyhedron, it is a straightforward procedure to calculate the aspect boundaries in viewpoint space from the asp.

The asp is a representation of the volume of aspect space that a polyhedron occupies, where aspect space is defined as viewpoint space cross the image plane, i.e. the tangent bundle of planes on the sphere. If we denote a viewpoint by (θ, ϕ) and a point in the image plane by (u, v) , then we can denote a point of aspect space (θ, ϕ, u, v) . The orientation of the image plane with respect to the line of sight is fixed so that objects appear “upright,” i.e. the viewer can yaw and pitch but not roll. The orientation of image the image plane when $\phi = \pm 90^\circ$ is defined to be the orientation as ϕ approaches 90° while $\theta = 0^\circ$.

The asp for a face of a polyhedron is the volume of aspect space that the face occupies, that is, the object's appearance in the viewing plane for every viewpoint (θ, ϕ) . Thus, the point (θ, ϕ, u, v) is in the asp for a face whenever (u, v) is in the image of that face from the viewpoint (θ, ϕ) . The (θ, ϕ) cross-section of the asp for the face is then the appearance of the face from the viewpoint (θ, ϕ) . The asp for a polyhedron is the union of the asps for its faces.

We can construct the asp for a point at location (x_0, y_0, z_0) in a fixed coordinate system by considering what happens to the image of the point as the viewpoint changes. The projection onto the image plane can be separated into two parts, a rotation so that the viewing direction is along the z-axis and an orthographic projection into the $z=0$ plane. The rotation is given by

$$[x_0, y_0, z_0] \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} =$$

$$\begin{aligned} & [x_0 \cos \theta - z_0 \sin \theta, & (5) \\ & x_0 \sin \theta \sin \phi + y_0 \cos \phi + z_0 \cos \theta \sin \phi, \\ & x_0 \sin \theta \cos \phi - y_0 \sin \phi + z_0 \cos \theta \cos \phi] \end{aligned}$$

After orthographic projection into the image plane (u, v) , this yields

$$u = x_0 \cos \theta - z_0 \sin \theta \quad (6)$$

$$v = x_0 \sin \theta \sin \phi + y_0 \cos \phi + z_0 \cos \theta \sin \phi \quad (7)$$

These equations have two degrees of freedom, θ and ϕ . Thus a point in 3-space has as its aspect representation a 2-D surface in aspect space.

The asp for a line segment is a 3-surface in aspect space bounded by 2-surfaces. It can be written down directly by substituting parametric equations for a line segment into the point equations. We use a parametric representation for the line segment from (x_0, y_0, z_0) to (x_1, y_1, z_1) , with parameter s varying from 0 to 1. Letting $a_1 = x_1 - x_0$, $b_1 = y_1 - y_0$, and $c_1 = z_1 - z_0$ we have

$$\begin{aligned} x(s) &= x_1 + a_1 s \\ y(s) &= y_1 + b_1 s \\ z(s) &= z_1 + c_1 s \end{aligned} \quad (8)$$

The point equations become

$$u = (x_1 + a_1 s) \cos \theta - (z_1 + c_1 s) \sin \theta \quad (9)$$

$$v = (x_1 + a_1 s) \sin \theta \sin \phi + (y_1 + b_1 s) \cos \phi + (z_1 + c_1 s) \cos \theta \sin \phi \quad (10)$$

This is a 3-surface in aspect space. The bounding 2-surfaces are given by Eqs. (6) and (7) above for the points (x_0, y_0, z_0) and (x_1, y_1, z_1) .

The asp for a triangle consists of the asps for the bounding sides and vertices as described above plus connectivity information, namely, links from the asps for

bounding sides to the asps for the adjacent vertices, and so on. Figure 22 shows the asp for the triangle $(1,1,0)$, $(2,1,0)$, $(1,2,0)$ in object space. Since it is difficult to represent a 4-D volume in a 2-D figure, the figure shows two 3-D cross-sections of the asp at different fixed values of ϕ . The 2-D cross-sections of the asp within each 3-D cross-section are the images of the triangle for the corresponding values of θ and ϕ . The asp represents all of these cross-sections continuously by representing the equations of the bounding surfaces.

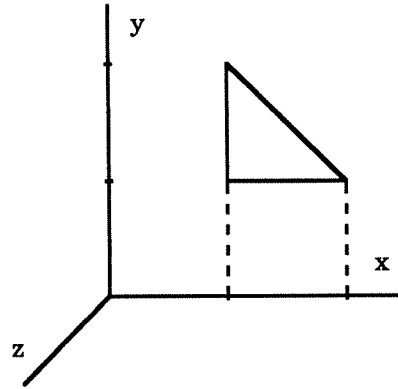


Figure 22a. The triangle $(1,1,0)$, $(2,1,0)$, $(1,2,0)$ in object space.

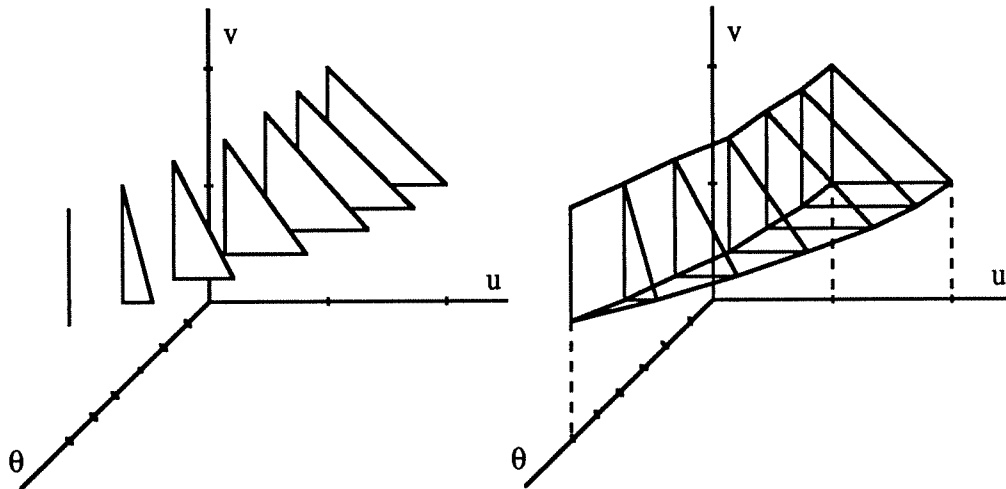


Figure 22b. A cross-section of the asp for the triangle for $\phi = 0^\circ$.

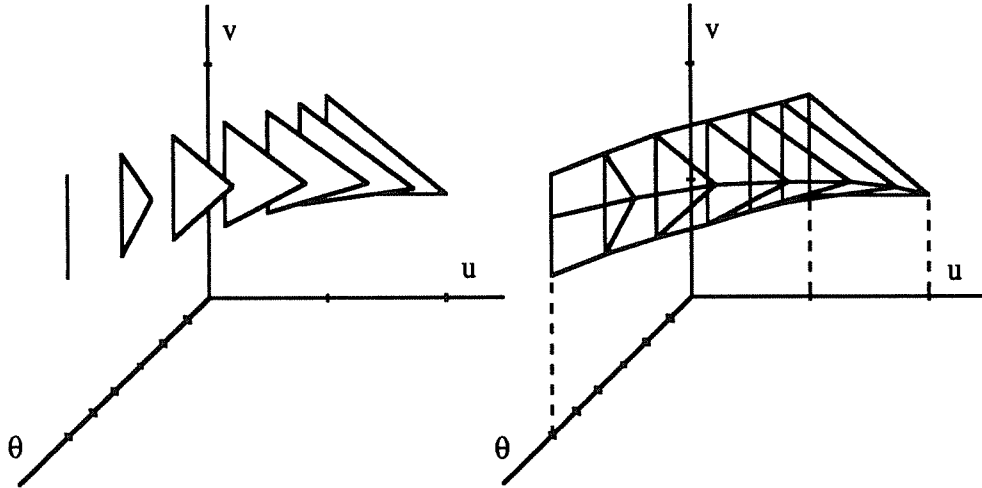


Figure 22c. A cross-section of the asp for the triangle for $\phi = 30^\circ$.

5.1.4.1.1. Representing Asps

We can represent an asp by representing the volume of aspect space that it occupies; specifically, by representing the 3-surfaces that bound the 4-D cells of aspect space. These 3-surfaces are bounded by 2-surfaces, the 2-surfaces by 1-surfaces, and the 1-surfaces by points. Unfortunately, the surfaces are not in general planar, so the object is not a polytope (i.e. the generalization of a polygon, polyhedron, etc.). However, the surfaces are well-behaved: the trigonometric functions in the equations for the surfaces can be eliminated by a spherical-to-Cartesian coordinate transformation of the representation for viewpoints, from (θ, ϕ) to $\mathbf{V} = (v_x, v_y, v_z)$. As a result, the surfaces are algebraic, and each can be represented with a few constants.

Therefore we can work with the surfaces in much the same manner as we would work with lines and planes in a polyhedron. For example, we can calculate the intersection of any two of these surfaces in closed form, so that calculating the intersection for two particular surfaces is a matter of plugging constants into an equation. The intersection of two 3-surfaces is a 2-surface and, with the viewpoint transformation above, it is a linear function of viewpoint. This kind of surface occurs in edge-vertex occlusion (Section 5.1.1.1). The intersection of three 3-surfaces is a 1-surface, a quadratic function of viewpoint. This kind of surface occurs in edge-edge occlusion (Section 5.1.1.2). A vertex is the result of the intersection of four 3-surfaces; it is a cubic function of viewpoint. The equations for these surfaces are developed below.

Thus, while the asp is not a polytope, we speak of it as if it were. Specifically, we refer to the 3-surfaces that bound the asp as *facets*, a term one would normally

use for the 3-D faces bounding a 4-polytope. We refer to the 2-surfaces that bound the facets as *ridges*, and the 1-surfaces or curves as *edges*. A 4-D volume of aspect space bounded by facets is referred to as a *cell*. In general, the asp is a 4-manifold in aspect space, i.e. a region or regions of aspect space partitioned into cells, which we refer to as a *subdivision*. The cells correspond to polygons in the image of the object and the facets bounding the cells correspond to edges.

Thus, the asp is very much like a set of 4-polytopes in aspect space except that the “faces” are curved, and we must store the constants of the equation for each face. We represent the asp with a data structure similar to that of a polyhedron, except that the asp is 4-D rather than 3-D and the volume of the asp is partitioned into 4-D cells. A cell of aspect space is represented as a node with pointers to the bounding facets. The facets are represented by the constants of the equation for the 3-surface and pointers to the bounding ridges, and so on. We can also work with asps much as we would with polyhedra. For example, we can determine whether two asps intersect by determining whether some face of one intersects some face of the other, and if not, whether one is completely inside the other.

5.1.4.1.2. Asps and Visibility

Consider again the asp for a point. Eqs. (6) and (7) for the asp for that point are defined for all values of θ and ϕ . This is as one would expect, since a point is visible from any viewing direction. We can represent a point which is not visible from every viewing direction, say a point which is occluded in some directions by a polygon, by putting bounds on the values of θ and ϕ for Eqs. (6) and (7) above. That is, if we represent only a part of the surface for the point in aspect space, rather than the whole surface, we have the asp for a point which is visible from only some viewpoints.

In fact, a fundamental property of aspect space is that occlusion in object space corresponds to set subtraction in aspect space. That is, if one polygon partially occludes another, we can characterize that occlusion in aspect space by subtracting the asp of the first from the second. For example, in Figure 23a we show two triangles in object space, one in the $z=0$ plane (left) and one in the $z=1$ plane; in Figure 23b we show the asps for the triangles (with $\phi = 0$).

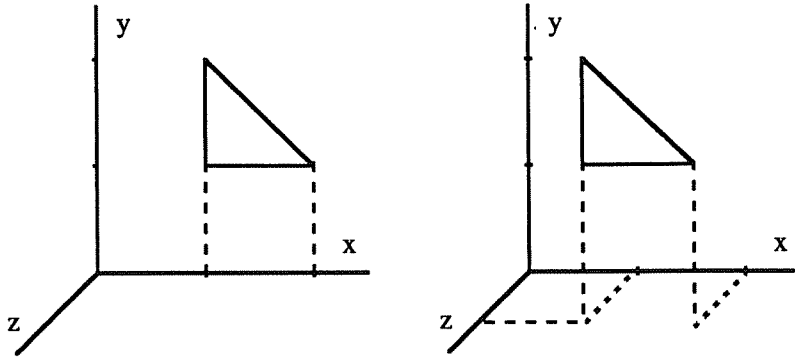


Figure 23a. Two triangles in object space.

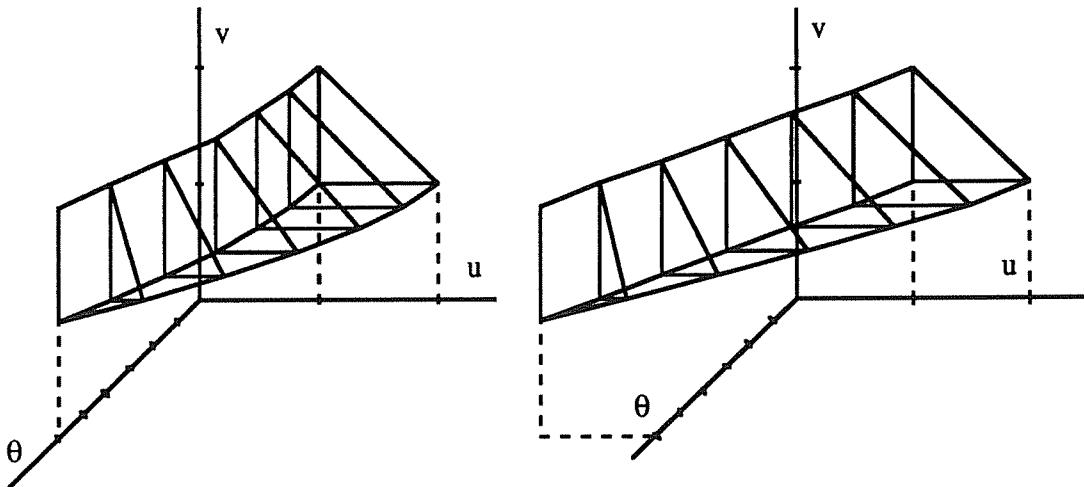


Figure 23b. The asps for two triangles.

In Figure 24 we show the set subtraction of the second triangle from the first. Since set subtraction in aspect space corresponds to occlusion in object space, any cross section of Figure 24 for a particular viewpoint corresponds to the part of the first triangle visible from that viewpoint (i.e. not occluded by the second).

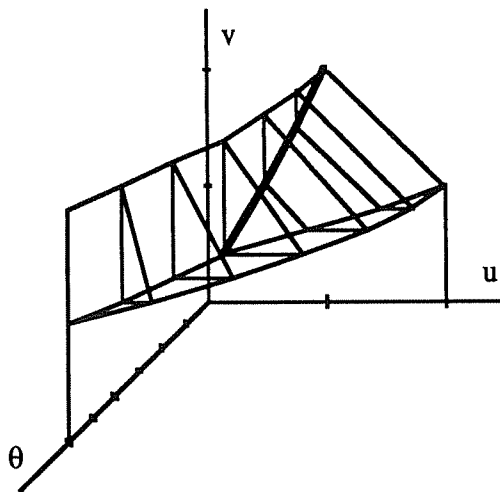


Figure 24. Occlusion in image space corresponds to subtraction in aspect space.

Since the asp represents appearance, features of the appearance of an object, i.e. visual events, correspond directly to features of the asp. For example, an edge in an image is visible from a 2-D range of viewpoints and has 1-D extent in image space, so there is a corresponding 3-D surface in aspect space representing it. Two non-intersecting edges of an object can appear to intersect in a single point (a “T-junction”) from a 2-D manifold of viewpoints, so there is a 1-face in the asp corresponding to that intersection point. In Figure 24, a 1-D cross-section of such a 2-D surface is shown as a bold curve. Table 3 presents a list of features in the appearance of an object and the corresponding features in the asp.

Visual Event	Asp feature
point in object space edge in object space polygon in object space apparent intersection point of two edges apparent intersection of edge and vertex apparent intersection of three edges apparent intersection of two vertices apparent intersection of four edges	ridge (2-face) facet (3-face) cell (4-face) ridge edge (1-face) edge vertex vertex

Table 3. Visual events and corresponding asp features.

5.1.4.1.3. Aspect Surfaces

The aspect representation for a non-convex polygon is a cell of aspect space, bounded by 3-surfaces or facets. The facets correspond to the edges of the polygon. Ridges bound the visibility of facets, and there are ridges corresponding to the vertices bounding the edges of the polygon. There is also another sort of ridge, corresponding to the case where the visibility of a polygon edge is bounded by another, occluding edge. In that case the ridge corresponds to the *apparent* intersection of the two object edges. This type of ridge results from the subtraction of the asp for the occluding polygon from the asp for the given polygon. The surface on which the ridge lies results from the intersection of the surfaces on which the two facets lie. This is the general sort of 2-surface.

The boundaries of an asp ridge are a result of the horizon effect or the more general sort of event bounding the visibility of a ridge: the apparent intersection of three object edges in a single image point. Asp vertices correspond to visual events that are visible from a single viewpoint and that occupy a single point of the image plane: the apparent intersection of four object edges in a single image point.

In order to find the intersection of cells of aspect space, we must be able to find the intersection of pairs of asp surfaces of various sorts. Note that all of the asp surfaces—3-surfaces, 2-surfaces, 1-surfaces, and vertices—result from the intersection of a set of asp 3-surfaces—one, two, three, and four respectively. Thus, finding the intersection of an asp 3-surface and an asp 2-surface is equivalent to finding the intersection of the three “parent” asp 3-surfaces involved. Also, since two asp 1-surfaces that lie on the same 2-surface have two common 3-surface “parents,” finding the intersection of two 1-surfaces is equivalent to finding the intersection of the four unique 3-surfaces involved in the problem. In general, finding the intersection of two asp surfaces is equivalent to finding the intersection of the unique “parents” of those surfaces.

A 3-surface of aspect space corresponds to the visibility of a line in object space. The equations for such a surface for a line passing through the points $\mathbf{p}_1 = (x_1, y_1, z_1)$ and $\mathbf{p}_1 + \mathbf{a}_1 = (x_1 + a_1, y_1 + b_1, z_1 + c_1)$ were derived earlier and are repeated here:

$$u = (x_1 + s a_1) \cos \theta - (z_1 + s c_1) \sin \theta \quad (11)$$

$$v = (x_1 + s a_1) \sin \theta \sin \phi + (y_1 + s b_1) \cos \phi + (z_1 + s c_1) \cos \theta \sin \phi \quad (12)$$

2-surfaces of aspect space arise in the general case as the intersection of two 3-surfaces. Thus, we can find the general form of a 2-surface by finding the intersection of two copies of Eqs. (11) and (12). However, we can find the equations more simply by noting that the viewing direction \mathbf{V} must be parallel to a vector from one of the lines to the other at the points where they appear to intersect, i.e. a vector from $\mathbf{p}_1 + s \mathbf{a}_1$ to $\mathbf{p}_2 + s_2 \mathbf{a}_2$ (see Figure 25).

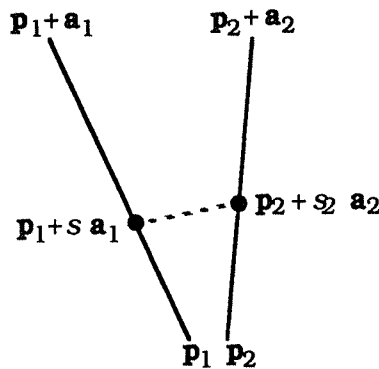


Figure 25. The viewing direction along which two lines appear to intersect.

Using $\mathbf{V} = (v_x, v_y, v_z)$ for the viewpoint, we get

$$\frac{v_x}{v_z} = \frac{(x_2 + s_2 a_2) - (x_1 + s a_1)}{(z_2 + s_2 c_2) - (z_1 + s c_1)} \quad (13)$$

and

$$\frac{v_y}{v_z} = \frac{(y_2 + s_2 b_2) - (y_1 + s b_1)}{(z_2 + s_2 c_2) - (z_1 + s c_1)} \quad (14)$$

Solving for s yields

$$s = \frac{\mathbf{V} \cdot [(\mathbf{p}_2 - \mathbf{p}_1) \times \mathbf{a}_2]}{\mathbf{V} \cdot (\mathbf{a}_1 \times \mathbf{a}_2)} \quad (15)$$

From these equations we can find u and v as a function of θ and ϕ . The surface can be represented by twelve constants: $a_1, a_2, b_1, b_2, c_1, c_2, x_1, x_2, y_1, y_2, z_1, z_2$. These are related to the coordinates of the four endpoints of the two lines involved in the visual event.

The intersection of three 3-surfaces of aspect space is a 1-surface or curve. It corresponds to the visual event of the apparent intersection of three object edges in an image. Three object edges can appear to intersect from a 1-D curve of viewpoints, and their apparent intersection is a single point, so the aspect representation for this visual event is a 1-surface or curve in aspect space.

We could find the 1-surface by finding the intersection of three pairs of equations of the form of Eqs. (11)-(12) for three 3-surfaces. However, there is a simpler way to find the form. Consider three edges in space, \mathbf{p}_1 to $\mathbf{p}_1 + \mathbf{a}_1$, \mathbf{p}_2 to $\mathbf{p}_2 + \mathbf{a}_2$, and \mathbf{p}_3 to $\mathbf{p}_3 + \mathbf{a}_3$. Pick a point $\mathbf{p}_1 + s \mathbf{a}_1$ on one line (see Figure 26) and find a line through that point that intersects both of the other lines.

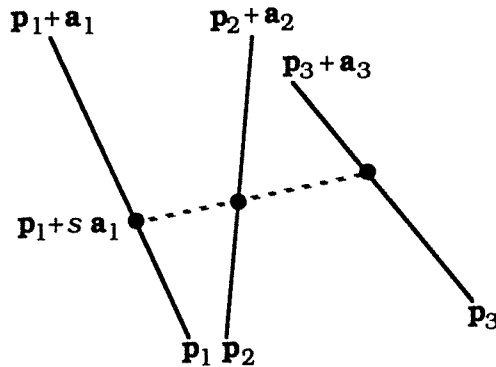


Figure 26. A viewing direction along which three object edges appear to intersect in a single point in an image.

Note the viewing direction must be along the intersection of the planes defined by the point $\mathbf{p}_1 + s \mathbf{a}_1$ and each of the other two lines. Normals to these planes are given by

$$(\mathbf{p}_1 + s \mathbf{a}_1 - \mathbf{p}_2) \times \mathbf{a}_2$$

and

$$(\mathbf{p}_1 + s \mathbf{a}_1 - \mathbf{p}_3) \times \mathbf{a}_3$$

so a vector parallel to the viewing direction in question is given by

$$\mathbf{V}' = ((\mathbf{p}_1 + s \mathbf{a}_1 - \mathbf{p}_2) \times \mathbf{a}_2) \times ((\mathbf{p}_1 + s \mathbf{a}_1 - \mathbf{p}_3) \times \mathbf{a}_3) \quad (16)$$

A vertex of aspect space results from the intersection of four asp 3-surfaces, two general 2-surfaces, two 1-surfaces on a 2-surface, or any other combination of surfaces with four unique 3-surface "parents." The visual event that gives rise to an asp vertex is the apparent intersection of four object edges in a single point. This occurs from only one viewpoint, so the corresponding asp surface is a vertex.

In order to find the viewing direction at which this visual event occurs, consider four lines in space, through \mathbf{p}_1 & $\mathbf{p}_1 + \mathbf{a}_1$, \mathbf{p}_2 & $\mathbf{p}_2 + \mathbf{a}_2$, \mathbf{p}_3 & $\mathbf{p}_3 + \mathbf{a}_3$, and \mathbf{p}_4 & $\mathbf{p}_4 + \mathbf{a}_4$. Pick a point $\mathbf{p}_1 + s \mathbf{a}_1$ on one line (see Figure 27) and suppose a line through that point intersects all three of the other lines.

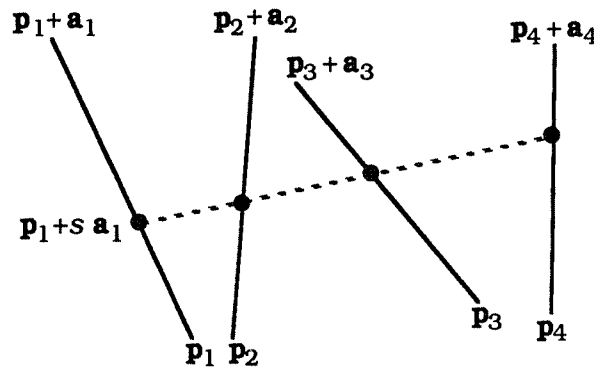


Figure 27. A viewing direction along which four object edges appear to intersect in a single point in an image.

The viewing direction must intersect the three planes define by the point $\mathbf{p}_1 + s \mathbf{a}_1$ and each of the other lines. Normals to these planes are given by

$$(\mathbf{p}_1 + s \mathbf{p}_1 - \mathbf{p}_2) \times \mathbf{a}_2$$

$$(\mathbf{p}_1 + s \mathbf{p}_1 - \mathbf{p}_3) \times \mathbf{a}_3$$

$$(\mathbf{p}_1 + s \mathbf{p}_1 - \mathbf{p}_4) \times \mathbf{a}_4$$

If the three planes intersect in a single line, then the normals have a triple cross product of zero, yielding a cubic equation in s . Thus, we can find the vertex by solving the cubic equation for s .

5.1.4.1.4. Size of Asps

In the case of a convex polyhedron, the asp has size $O(n)$ for an object with n faces since there is a face in the polyhedron for every facet in the asp and the sizes of the facets in the asp correspond directly to the size of the faces in the polyhedron. However, in the non-convex case the asp is much larger since the cross section of an asp at any value of (θ, ϕ) is a view of the corresponding object with hidden lines removed. In [Plantinga and Dyer, 1987] it is shown that the maximum size of the asp for a general polyhedron with n faces is $\Theta(n^4)$. However, the worst case occurs only for objects or scenes that are visually very complex, that is, that have a large amount of occlusion; Figure 28 is an example of such a scene.

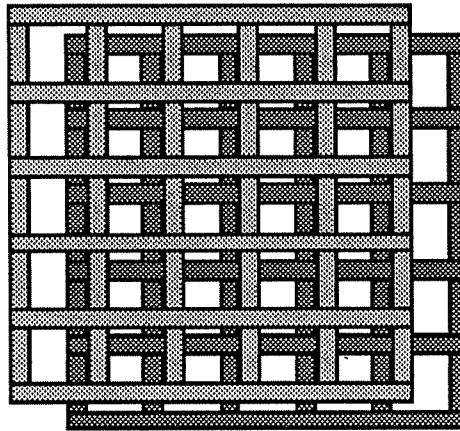


Figure 28. An example of a scene for which the asp has size $O(n^4)$.

It is perhaps surprising that the asp, which represents the appearance of an object from every viewpoint, has smaller size than the VSP, which only represents the regions of constant aspect in viewpoint space. The asp has size $O(n^4)$ while the VSP has size $O(n^6)$. The asp is smaller for a reason analogous to the reason that a polyhedron of size n can have an image of size $O(n^2)$. Forming the image of the polyhedron requires projecting lines in R^3 into R^2 , and lines that did not intersect in R^3 may intersect in R^2 . A set of n lines can have $O(n^2)$ intersection points. Similarly, projecting the $O(n^3)$ 1-surfaces of the asp into viewpoint space causes them to intersect where they did not intersect before, resulting in $O(n^6)$ intersection points.

5.1.4.1.5. Constructing the Asp

The asp for a polygon is a 4-D cell of aspect space bounded by the facets corresponding to the edges of the polygon. The edges are bounded by ridges corresponding to the vertices of the polygon. Since a polygon by itself is visible from all viewpoints in front of it, the boundary of each ridge is a great circle in viewpoint space at the image point at which the vertex of the polygon appears. The asp for a polygon can therefore be constructed by making a node for the cell corresponding to the polygon, facet nodes for the edges bounding the polygon, ridge nodes for the vertices bounding each edge of the polygon, and edge nodes corresponding to the great circles bounding the visibility of the ridges. The facet, ridge, and edge nodes of the asp have constants defining the surface on which they lie.

The asp for a polygon f that is part of a polyhedron or polyhedral scene is some subset of the asp for the polygon by itself since some parts of it may not be visible from all viewpoints. The only faces that can obstruct the view of f are the faces at least partially in front of f . Since occlusion in object space corresponds to subtraction in aspect space, we can construct the asp for f as a part of the polyhedron by subtracting the asps for all the faces or parts of faces in front of it. The result is the asp for f as obstructed by the other faces; that is, a representation for all viewpoints of the parts of f that are visible.

The asp for a polyhedron or polyhedral scene is the union of the asps for its faces, so we construct it by constructing the asp for each face and taking the union. Since the asp for each face is one or more 4-D cells of aspect space, the asp for a polyhedron is a subdivision of aspect space.

We compute the subtraction of one cell of aspect space from another by finding the intersection of the first with the complement of the second. The intersection can be found by a brute-force algorithm: compare every face of one cell with every face of the other and find all intersections. The intersection of the cells can be constructed from the intersections of faces. The subtraction of one cell of aspect space from a subdivision is subtraction of that cell from each cell of the subdivision.

Algorithms for constructing asps and finding unions and intersections are presented in greater detail in [Plantinga and Dyer, 1987a] and [Plantinga and Dyer, 1987b]. In the former paper it is also shown that subtraction of a cell from an asp can be constructed in $O(nm)$ time for asps of size n and m using the brute-force intersection algorithm, and the asp for a polyhedral scene of size n can be constructed in time $O(n^5)$ in the worst case since the asp has size $O(n^4)$.

5.1.4.2. Using the Asp to Construct the VSP

All boundaries of image regions are represented as a function of viewpoint by faces in the asp, so in particular every event that changes the topology of the image is represented in the asp. The asp faces corresponding to events are edges. However, not all asp edges correspond to events; some correspond to changes that do not result in a topologically distinct image. The edges of the asp that correspond to events are those in which the generating faces of the object are arranged in the manner of one of the kinds of events listed above. Thus, to find events, we construct the asp for the polyhedron and find the asp edges that correspond to one of the event types.

The resulting edge must be projected into viewpoint space, which can be done by solving for viewpoint as a function of a single parameter. The equation for an asp 1-surface is given in this form in Eq. 16. We must also be able to find intersections of these curves in viewpoint space. Eq. 16 is a vector equation of the form

$$\mathbf{V}' = \mathbf{a} s^2 + \mathbf{b} s + \mathbf{c}$$

The intersection point of two such curves is the solution of the vector equation

$$\mathbf{a}_1 s_1^2 + \mathbf{b}_1 s_1 + \mathbf{c}_1 = \mathbf{a}_2 s_2^2 + \mathbf{b}_2 s_2 + \mathbf{c}_2$$

The solutions of this equation can be found with the quadratic and quartic equations. There may be as many as four intersection points of the two curves.

It remains to construct the subdivision of viewpoint space generated by these events. We call the cells of the subdivision *regions*, the boundaries of the regions *edges*, and the intersection points of the edges *vertices*. We will use as a data structure for the subdivision nodes for regions, edges, and vertices, with links between nodes corresponding to adjacent regions and edges or edges and vertices. At each edge node we store the constants of the curve on which the edge lies.

The edge-vertex structure of the subdivision generated by m curves on the sphere is constructed incrementally, starting with an empty subdivision and adding curves one by one. Adding a curve c to the subdivision involves finding the intersection points of c with every curve already in the subdivision and inserting them into the subdivision. c will intersect each curve already in the subdivision at most four times. Each intersection point is added to the subdivision by splitting the two edges that cross at an intersection point into four edges that meet at that point. The edge on which the intersection point lies can be found in $O(\log n)$ time with binary search.

After all of the curves have been added to the subdivision, the result is a structure of edges and vertices. The regions can be added with a graph-search algorithm in linear time in the size of the vertex-edge structure. Since each intersection point is found in $O(\log m)$ time, a subdivision generated by m curves is constructed in $O(m^2 \log m)$ time.

The asp is constructed in $O(n^5)$ time and results in $O(n^3)$ boundaries in viewpoint space. The subdivision can be constructed in $O(n^6 \log n)$ time, so the time to construct the VSP is $O(n^6 \log n)$. The construction time is dominated by the time to find the subdivision unless the polyhedron is highly degenerate and does not generate many events. That can occur when most of its edges are parallel or meet at a single vertex. The subdivision algorithm takes time $O(m \log m)$ for an output size of m for any subdivision that is not highly degenerate, so the runtime of the VSP construction algorithm is nearly optimal in the sense that its runtime is within a log factor of the output size for polyhedra that are not highly degenerate.

This algorithm is more efficient than the brute-force algorithm for constructing the VSP in the worst case, $O(n^6 \log n)$ vs. $\Theta(n^7)$. It is much more efficient than the brute-force algorithm for objects that are not as visually complex as the worst case. For example, constructing the asp for a convex object takes $O(n^2)$ time using the algorithm given, and finding the VSP generated by the $O(n)$ events takes $O(n^2 \log n)$ time. The brute-force algorithm requires $\Theta(n^7)$ time even in this case.

The aspect graph is the dual of the region-boundary structure of the VSP. We find it by finding the dual of the VSP, as in the convex case. The aspect graph has the same maximum size as the VSP, $\Theta(n^6)$. The aspect graph has a vertex for every distinct aspect of the polyhedron, but as we have defined it the vertices of the aspect graph contain no information about the particular aspect that they represent. In order to characterize the aspect one can store an image of the object from one of the views in the viewing region of the VSP. This requires $O(n^2)$ space for each aspect and can be computed in $O(n^2)$ time with a hidden-line removal algorithm, so the aspect graph with images at each vertex requires space and time $O(n^8)$ to construct.

5.2. Perspective Case

Under the perspective model, the viewpoint is unrestricted, so viewpoint space is R^3 . Therefore the VSP is a partition of R^3 . It has cells corresponding to volumes of viewpoint space of constant aspect. Since we are modelling “looking at” an object that is centered on the origin, the viewing direction is again the direction from the viewpoint to the origin.

We also no longer assume that the whole object is in front of the viewer as in the orthographic model. For example, in modelling a house under the orthographic model, the only viewpoints allowed are a sphere of viewpoints outside of the house. However, in the perspective model viewpoint space is R^3 , so the viewpoint can be outside of the house, in the living room, or inside a closet.

5.2.1. Occlusion under Perspective Projection

The same visual events that occur in the orthographic model (horizon effect, edge-vertex, edge-edge-edge, etc.) also occur in the perspective model. However, since viewpoint space is R^3 , the boundary generated in viewpoint space in each case is a surface in R^3 rather than a curve on the sphere. In the orthographic model, a face turning away from the viewer generates a visibility boundary that is an arc of a great circle on the sphere. In the perspective model, the boundary of visibility is a part of a plane in viewpoint space, specifically, the plane containing the face. This is true because a face “turns away from the viewer” whenever the viewpoint drops below the plane containing the face.

Edge-vertex occlusion boundaries (see Figures 15 and 16) are also parts of a plane. The plane is defined by the vertex and the edge involved in the visual event. The lines bounding the section of the plane corresponding to the occlusion boundary are the lines defined by the vertex and each of the endpoints of the edge.

Edge-edge-edge occlusion boundaries (see Figures 17 and 18) are the general sort of occlusion boundaries. In the orthographic model they are curves on the sphere; in the perspective model the boundary is a surface in R^3 . Thus there is a surface of viewpoints from which three edges appear to intersect in a point. In the perspective model, the viewing directions in which the three edges appear to intersect in a single image point is a line of viewpoints parallel to the viewing direction of the orthographic case (Eq. 16). The line passes through a point on an object edge at which all three edges appear to intersect, so the equation for a viewpoint from which the three edges appear to intersect in a single point is

$$\mathbf{V} = \mathbf{p}_1 + s \mathbf{a}_1 + r [(\mathbf{p}_1 + s \mathbf{a}_1 - \mathbf{p}_2) \times \mathbf{a}_2] \times ((\mathbf{p}_1 + s \mathbf{a}_1 - \mathbf{p}_3) \times \mathbf{a}_3] \quad (17)$$

5.2.2. Upper Bound

The surfaces in R^3 that bound the regions of constant aspect are generated by visual events involving triples of object edges. There are $O(\mathbf{n}^3)$ triples of edges and

hence $O(n^3)$ such surfaces for a polyhedron of size n . Since the surfaces are algebraic, any pair has a constant number of curves of intersection and any three have a constant number of points of intersection. Thus the $O(n^3)$ surfaces have $O(n^9)$ intersection points.

On any one of the surfaces there can be at most $O(n^6)$ edges and faces, because the intersection of that surface with the other surfaces results in a 2-D subdivision of that surface by $O(n^3)$ curves. Summing for all surfaces, a total of $O(n^9)$ vertices, edges, and faces bound cells of viewpoint space. Since a cell must have bounding faces and a face can bound at most two cells, there are $O(n^9)$ cells. Thus, the VSP has size $O(n^9)$.

5.2.3. Lower Bound

In this section we show that $O(n^9)$ is a tight bound on the maximum size of the VSP and aspect graph. We present a polyhedral scene that has a VSP of size $\Omega(n^9)$. The example is similar to the lower bound example in the orthographic case (Section 5.1.3) except that changing the viewpoint in any of three orthogonal directions changes the aspect (see Figure 29).

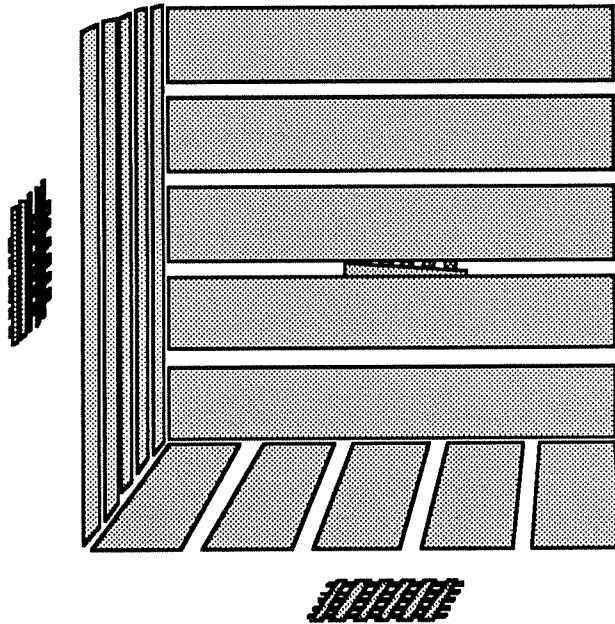


Figure 29. Polyhedra with VSP of size $\Omega(n^9)$.

Figure 29 presents a polyhedral scene with three screens on three adjacent sides of a cube. There are three grids, each one a distance behind one of the screens. We will show that there are $\Omega(n^9)$ different aspects from viewpoints inside the cube.

From any viewpoint inside the cube, some subset of faces of each grid is visible through one of the slits. In a manner similar to the orthographic example of Section 5.1.3, changing the viewpoint parallel to one of the edges of the cube changes the faces of one grid through one screen, but it does not affect the view of the other two grids. Each grid generates $\Omega(n^2)$ boundaries of visibility through each slit, or $\Omega(n^3)$ boundaries of visibility through the whole screen. All of the boundaries are parallel planes, so the $\Omega(n^3)$ planes corresponding to each screen/grid pair intersect in $\Omega(n^9)$ points. Therefore the VSP and the aspect graph have size $\Omega(n^9)$.

Note that we could just as well have used orthographic projection in constructing this example. However, in our orthographic model we define viewpoint space to be the sphere. This example would suffice as a $\Omega(n^9)$ lower bound under orthographic projection as well if viewpoint space is \mathbb{R}^3 .

5.2.4. An Algorithm

Since the VSP is a partition of \mathbb{R}^3 under the perspective model, in order to construct it we must be able to calculate the surfaces in viewpoint space that bound the aspects. These surfaces are given by Eq. (11) for sets of three object edges in all orders. As in the orthographic case, we can calculate the VSP using a naive algorithm: construct all possible boundaries of visibility using Eq. (11) for all sets of three object edges in all orders. Find the subdivision of viewpoint space that these boundaries generate; the result is the VSP.

A subdivision of \mathbb{R}^3 by m algebraic surfaces has size $O(m^3)$, so the $O(n^3)$ boundaries in viewpoint space generate a subdivision of size $O(n^9)$ with a 3-D viewpoint space. The subdivision can be constructed in $O(n^9 \log n)$ time using the algorithm given in Section 5.2.4.2. It remains to test each face f that separates two cells of the subdivision to determine whether the aspect actually changes at f . We can test whether the aspect actually changes by finding the appearance of the image immediately around the image point where the event occurs that gave rise f . If section of the image near the event are the same from viewpoints immediately to either side of f , then that face should be removed from the VSP and the cells on either side merged. This test can be performed in $O(n)$ time for each face of the VSP, so using this algorithm the VSP can be constructed in $O(n^{10})$ time.

Unfortunately, as in the orthographic case, the best-case and worst-case times are the same for this algorithm: it requires time $\Omega(n^{10})$ even for simple objects. An algorithm will perform better on the average if it finds the the exact set boundaries in viewpoint space *before* finding the subdivision that they generate. It is important

to keep the number of boundaries in viewpoint space as small as possible before finding the subdivision that they generate because m boundaries that are not highly degenerate form a subdivision of size $\Theta(m^3)$. In order to compute the exact visibility boundaries for each face of the polyhedron, we again use the aspect representation, this time constructed using perspective projection. We first describe the asp under perspective projection and then show how to use it to construct the VSP.

5.2.4.1. The Asp under Perspective Projection

Since in the perspective case we take viewpoint space to be R^3 rather than the sphere, aspect space is a 5-D space: R^3 cross the image plane. To represent viewpoint space we use spherical coordinates (θ, ϕ, r) , similar to the orthographic case but with the addition of distance r from the origin. Thus a point of aspect space is represented (θ, ϕ, r, u, v) , where (θ, ϕ, r) is the viewpoint and (u, v) is the point on the image plane. The viewing direction is the direction from (θ, ϕ, r) to the origin. A point (θ, ϕ, r, u, v) of aspect space is in the asp for a polyhedron when (u, v) is in the image of the polyhedron from the viewpoint (θ, ϕ, r) . In some cases we will also use Cartesian coordinates for viewpoints, $(v_x, v_y, v_z) = \mathbf{V}$.

Since a point $\mathbf{p} = (p_x, p_y, p_z)$ by itself in object space is visible from all viewpoints, it has as its asp one point in the image plane for each viewpoint \mathbf{V} . That is, the asp for a point is a 3-surface in aspect space. We can determine the point in the image plane in the same manner that we found the point in the orthographic case, except that we now use perspective rather than orthographic projection. In that case, after a rotation to make the viewpoint lie on the z-axis, the point \mathbf{p} was given by Eq. (5) in Section 5.1.4.1. Under perspective projection, a point (p_x, p_y, p_z) projects to

$$\left[\frac{f}{d} p_x, \frac{f}{d} p_y \right]$$

where f is the focal length and d is the distance along the z-axis from the viewpoint to the object point, i.e. the difference in z-coordinates. Thus, the asp for the point (x_0, y_0, z_0) from the viewpoint (θ, ϕ, r) is given by

$$u = \frac{f(x_0 \cos \theta - z_0 \sin \theta)}{r - (x_0 \sin \theta \cos \phi - y_0 \sin \phi + z_0 \cos \theta \cos \phi)} \quad (18)$$

$$v = \frac{f(x_0 \sin \theta \sin \phi + y_0 \cos \phi + z_0 \cos \theta \sin \phi)}{r - (x_0 \sin \theta \cos \phi - y_0 \sin \phi + z_0 \cos \theta \cos \phi)} \quad (19)$$

The asp for a line segment is a 4-surface in aspect space. The equations for the asp can be had by substituting Eq. (8) (the parametric equations for a line) for x_0 , y_0 , and z_0 into Eqs. (18) and (19). The asp for a polygon is a cell of aspect space bounded by the 4-surfaces corresponding to the edges bounding the cell.

The asp for a polyhedron under the perspective model is much like the asp under the orthographic model. The difference is that the asp is a 5-manifold in a 5-D aspect space rather than a 4-manifold in 4-space. It can be constructed in the same way as in the orthographic case, if the algorithms for intersection and union are generalized to 5-D.

The asp faces corresponding to visual events are of one dimension higher in the perspective case. In Table 4 we show visual events and corresponding asp features for the perspective case. Note that there are no true 0-D visual events. A 0-D visual event would be a visual event visible at a particular point in the image, from only one viewpoint. However, for any visual event there is some line in viewpoint space along which the same visual event is visible, at points a little closer or further from the object. The only way that a 1-D visual event ends is when the line of sight passes through a face of a polyhedron in object space.

Visual Event	Asp feature
point	ridge (3-face)
edge	facet (4-face)
polygon	cell (5-face)
apparent int. point of two object edges	ridge
apparent intersection of edge and vertex	2-face
apparent intersection of three edges	2-face
apparent intersection of two vertices	edge (1-face)
apparent intersection of four edges	edge (1-face)

Table 4. Visual events and corresponding asp features under perspective projection.

The surfaces corresponding to these events are closely related to those in the orthographic case. There are two differences, resulting from the use of perspective rather than orthographic projection and from taking viewpoint space to be R^3 rather than the sphere. The viewing directions from which an event occurs in the perspective case are the same as those from which the event occurs in the orthographic case. However, all of the viewpoints along a line parallel to the viewing direction and intersecting an object point at which the event occurs must be taken as the set of viewpoints at which the event occurs. Thus the surface is of one dimension higher in the perspective case.

Under the perspective model the asp faces correspond to the same visual events as they do under the orthographic model, except that the surfaces are in R^3 rather than on the sphere. Some visual events that were not visible in the orthographic case because they were behind other faces may be visible in the perspective case from some viewpoints in R^3 . However, the asymptotic worst-case size of the asp and time to construct the asp is the same as the orthographic case since the same events are visible in the worst case.

5.2.4.2. Using the Asp to Construct the VSP

As in the orthographic case, all boundaries of VSP regions are represented as a function of viewpoint by faces in the asp. In the perspective case, events are the projections of 2-faces of the asp into viewpoint space. Thus, we find the events for a polyhedron by constructing the asp for the polyhedron and projecting into viewpoint space those 2-faces that correspond to events. The 2-faces that correspond to events are those in which the object faces generating the 2-face are arranged in the manner of one of the types of events listed above. The boundaries found in this manner are not potential but actual events, since if they occur in the asp, they are by definition visible. It remains to project the 2-faces into viewpoint space and find the subdivision of viewpoint space they generate.

Projecting the 2-faces into viewpoint space is done by solving the equation for a 2-surface for viewpoint as a function of two parameters. The equation for the 2-surface is already expressed in this form in Eq. 17. The form of Eq. 17 is

$$\mathbf{V} = \mathbf{a}_1 + s \mathbf{a}_2 + r [s^2 \mathbf{a}_3 + s \mathbf{a}_4 + \mathbf{a}_5]$$

The intersection of two such surfaces is a curve of the form

$$\mathbf{a}_1 + s_1 \mathbf{a}_2 + r_1 [s_1^2 \mathbf{a}_3 + s_1 \mathbf{a}_4 + \mathbf{a}_5] = \mathbf{a}_6 + s_2 \mathbf{a}_7 + r_2 [s_2^2 \mathbf{a}_8 + s_2 \mathbf{a}_9 + \mathbf{a}_{10}] \quad (20)$$

This equation is satisfied for some values of r_1 and r_2 whenever three vectors are coplanar: the two viewing directions $[s_1^2 \mathbf{a}_3 + s_1 \mathbf{a}_4 + \mathbf{a}_5]$ and $[s_2^2 \mathbf{a}_8 + s_2 \mathbf{a}_9 + \mathbf{a}_{10}]$ and the vector between the two object points at which the event occurs, $\mathbf{a}_1 + s_1 \mathbf{a}_2 - \mathbf{a}_6 - s_2 \mathbf{a}_7$. Therefore Eq. (20) is satisfied when the triple cross product of these three vectors is zero. This yields a cubic equation in s_1 and s_2 , which can be solved for s_2 in terms of s_1 with the cubic equation. The intersection of three surfaces of the form of Eq. 17 yields two algebraic equations for s_2 in terms of s_1 ; finding intersection points involves setting the equations equal and finding zeros using numerical methods.

It remains to construct the subdivision of viewpoint space generated by these events. We will call the volumes in viewpoint space of the subdivision and 2-, 1-, and 0-dimensional boundaries *cells*, *faces*, *edges*, and *vertices*, respectively. We use a data structure similar to that of the subdivision of the sphere, with the addition of nodes for cells and links from each cell to the faces bounding it. For each face we store the constants defining the surface on which it lies.

We construct the subdivision generated by m surfaces in \mathbb{R}^3 by adding surfaces to the partial subdivision, one by one. We add a surface to the subdivision by finding the intersections of that surface with every other surface. The intersection curves are added to the 2-D subdivisions on each surface in the same way that the 2-D subdivision was constructed in the orthographic case. After the vertex-edge structure of the subdivision has been constructed, the faces of the subdivision on each surface are found in the same manner as in the orthographic case, and the cells are added in linear time using a graph-search algorithm. Since each intersection point is found in $O(\log m)$ time, a subdivision of m curves is constructed in $O(m^3 \log m)$ time.

The asp is constructed in $O(n^5)$ time and results in $O(n^3)$ boundaries in viewpoint space. Therefore the subdivision can be constructed in $O(n^9 \log n)$ time, so the time to construct the VSP is $O(n^9 \log n)$. The time complexity of constructing the VSP is dominated by the time to find the subdivision unless the polyhedron is highly degenerate and does not generate many events. The subdivision algorithm takes time $O(m \log m)$ for an output size of m for any subdivision that is not highly degenerate, so the runtime of the VSP construction algorithm is nearly optimal in the sense that its runtime is within a log factor of the output size for polyhedra that are not highly degenerate.

The algorithm for constructing the VSP is more efficient than the naive algorithm in all cases: the runtimes are $O(n^9 \log n)$ and $\Theta(n^{10})$ respectively. However, the asp algorithm is far more efficient in the case where the polyhedron is visually simple. For example, for a convex polyhedron the naive algorithm requires time $\Theta(n^{10})$, but the asp algorithm takes time $O(n^3 \log n)$.

The aspect graph, the dual of the cell-face structure, is constructed in linear time in the size of the VSP in the same manner as in the convex case. The aspect from some viewpoint requires $O(n^2)$ time to compute and $O(n^2)$ space to represent, so storing the aspect at each vertex of the aspect graph in the perspective case requires time and space $O(n^{11})$.

6. Related work

J. Koenderink and A. van Doorn [1976] laid the groundwork for the study of the topologically-distinct views of an object when they introduced the idea of the “singularities of the visual mapping” for a smooth body. The singularities of the visual mapping are points of the image that map to points in the world at local minima and maxima of distance and points on surfaces tangent to the viewing direction. They catalog the ways that the topological structure of the singularities can change and show that for most vantage points the structure of the singularities does not change with small changes in viewpoint. They call a change in the topological structure an “event,” and they catalog the different kinds of events that occur for smooth bodies.

In a later paper [1979] Koenderink and van Doorn define an “aspect” as the structure of the singularities of a stable view of a smooth object, and they define the “visual potential” graph for such an object, which we call the aspect graph. However, their domain of objects is different from ours since they assume that all objects are smooth and we assume that all objects are polyhedral. In addition, their model of visibility is different from ours. We assume that the edges of a polyhedron are visible, while they are concerned with the “singularities of the visual mapping,” which are image points on occluding edges or contours and image points corresponding to object points on surfaces that are tangent to the line of sight.

In this paper we showed how to construct the aspect graph for polyhedra under a viewing model in which the edges of a polyhedron are visible in an image. We also discuss the VSP, which Koenderink and van Doorn do not define or discuss in detail. They do briefly mention something similar, however, when they state that “a general decomposition of [viewpoint space] into cells that provide a stable global aspect is by no means trivial to carry out. To our knowledge this problem has not yet been solved by geometers.” [1976, p. 57]. The cells of “stable global aspect” are the cells of what we call the VSP, except for the differences in the types of objects we allow and in the kind of events we recognize.

Little work has been published on constructing the aspect graph. Werman, Baugher, Gualtieri [1986] show how to construct an aspect graph for a convex polygon with viewpoint constrained to lie in the plane containing the polygon, using perspective projection. They also present some properties of the aspect graph in that case. McKenna and Seidel [1985] construct minimal and maximal shadows of convex polyhedra in the plane. In doing so, they project the great circles corresponding to the faces onto a plane and construct the resulting arrangement of the lines in the plane using the $O(n^2)$ algorithm in [Edelsbrunner *et al.*, 1986] or [Chazelle, Guibas,

and Lee, 1985]. The result is similar to the VSP for convex polyhedra under orthographic projection. Kender and Freudenstein [1986] discuss the meanings of terms such as “degenerate view,” “characteristic view,” “visual event,” and “general viewing position.”

Although little work has been done on constructing the aspect graph, researchers have adopted it (and related ideas) as a tool for working with the “topologically-distinct views” of objects for computer vision. Every vertex of the aspect graph corresponds to an aspect or appearance of the object from a stable view. In multiview object recognition, several characteristic views of an object are stored with the idea that matching an image to several 2-D views or aspects may be simpler and faster than matching the image to a 3-D model. In order to find an efficient set of 2-D views, topologically-distinct views are sometimes selected—views corresponding to vertices of the aspect graph. Views may also be selected on the basis of how likely they are to occur in an image. For example, a view of an object in a stable, upright position of constant aspect over a wide range of viewing directions is more likely to occur than many other views.

Researchers have used different numbers of characteristic views, ranging from a small or moderate constant (up to about 300) [Chakravarty and Freeman, 1982; Fekete and Davis, 1984; Korn and Dyer; Scott, 1984; Thorpe and Shafer, 1983] to every “topologically distinct” view of the object [Castore, 1983; Castore and Crawford, 1984; Crawford, 1985]. Ikeuchi [1987] constructs and uses an “interpretation tree” in a 3-D object recognition algorithm for bin-picking. This interpretation tree bears some resemblance to the aspect graph. Rosenfeld [1986] suggests that fast object recognition will require representing an object as a series of “characteristic views” or “aspects.” Other researchers propose that the aspect graph be used as part of a general representation for computer vision [Schneier, Lumia, & Kent, 1986].

Related problems in the general area of visibility have also received attention. In computer graphics much work has been done on hidden-line removal (see, for example, [Sutherland, Sproull, and Schumacker, 1974]), which is essentially computing the visibility of a scene from a particular viewpoint. El Gindy and Avis [1981] and Lee [1983] compute the “visibility polygon” from a point in the plane, that is, the list of edges visible from the point.

Avis and Toussaint [1981] compute visibility from many viewpoints simultaneously when they discuss the visibility of a polygon in the plane from viewpoints along an edge in the plane. However, their algorithm computes the parts of a polygon that are visible from *some* point along a line in the plane, rather than visibility from *every* point along the line. Consequently, their work does not really

compute visibility from a continuous region of viewpoints. Other work on the visibility polygon from an edge includes [El Gindy, 1984; Chazelle and Guibas, 1985; Guibas *et al.*, 1986; Suri and O'Rourke, 1986].

Canny [1984] computes the region of a plane of viewpoints from which a polygon in \mathbb{R}^3 is visible. He discusses computing the “umbra” and “penumbra” of a polygonal face of a polyhedron relative to a polygonal light source. The umbra is the region of partial visibility and the penumbra is the region of total occlusion of the face. He gives an algorithm for constructing the “shadow” of a polygon relative to another polygon on the plane of viewpoints. However, his method does not construct the shadow caused by more than one polygon because he does not consider edge-edge events.

7. Conclusion

In this paper we analyzed the visibility and occlusion of faces of a polyhedron over all viewpoints. We showed how to calculate boundaries of visibility and regions of constant aspect in viewpoint space. We did this by listing the ways in which aspect can change. The aspect changes when a region in the image appears or disappears, or when regions merge or split.

We used two tools for characterizing aspect over all viewpoints: the viewpoint space partition and the aspect graph. We present tight bounds on the maximum size of the VSP and aspect graph in the convex and non-convex cases, under two models: orthographic projection and a 2-D spherical space of viewpoints and perspective projection where viewpoint space is \mathbb{R}^3 . Thus we bound the maximum number of topologically-distinct views of a polyhedral object in those cases. Under the orthographic model the maximum size is $\Theta(n^2)$ in the convex case and $\Theta(n^6)$ in the non-convex case. Under the perspective model the maximum size is $\Theta(n^3)$ in the convex case and $\Theta(n^9)$ in the non-convex case. The difference in size between the two models is due to the difference in dimensionality of the viewpoint space in each model, not the use of orthographic or perspective projection.

We also presented algorithms for constructing the VSP and the aspect graph for convex and non-convex polyhedra, under orthographic and perspective projection. The runtimes of these algorithms are summarized in Table 2, Section 2. The algorithms involve determining the visibility of each face of the polyhedron as a function of viewpoint, which is represented using the aspect representation or asp. The asp has a cell for each region that appears in an image, so constructing the VSP is done by finding the shadows of the cells on viewpoint space and computing the

subdivision of viewpoint space that they generate. The algorithms for convex polyhedra run in time that is worst-case optimal and in fact optimal for all polyhedra that are not highly degenerate. The algorithms for the non-convex polyhedra run in nearly optimal time in the sense that the runtime is within a log factor of output size for any polyhedron that is not highly degenerate.

A motivation for constructing the aspect graph is to find the topologically-distinct views of an object for the purposes of object recognition. However, we have shown that the number of views of a non-convex polyhedron can be extremely large, $O(n^6)$ in the orthographic case and $O(n^9)$ in the perspective case. Therefore, algorithms for object recognition that work on relatively complex objects probably cannot use all topologically-distinct views of a non-convex object.

Since the VSP is large, it would be interesting to construct an approximation to the VSP that captures the most important changes in visibility. One possible approach to this is to construct a hierarchical VSP by constructing the VSP for each level of a hierarchical representation of the object. It would also be interesting to construct the VSP for objects with some curved faces, such as cylinders, or to define and construct the VSP procedurally, in a manner similar to a constructive solid geometry representation.

REFERENCES

- Avis, D. and G. Toussaint, "An optimal algorithm for determining the visibility of a polygon from an edge," *IEEE Trans. Comp.* **30**, 1981, pp. 910-914.
- Canny, J., "Algorithms for model-driven mechanical parts inspection," Research Report RC 10505 (#48869), IBM Watson Research Center, 1984.
- Canny, J., personal communication, 1987.
- Castore, G., "Solid modeling, aspect graphs, and robot vision," *General Motors Conf. on Solid Modeling*, 1983.
- Castore, G. and C. Crawford, "From solid model to robot vision," *Proc. IEEE First Int. Conf. on Robotics*, pp. 90-92, 1984.
- Chakravarty, I. and H. Freeman, "Characteristic views as a basis for three-dimensional object recognition," *Proc. SPIE 336 (Robot Vision)*, pp. 37-45, 1982.
- Chazelle, B. and L. Guibas, "Visibility and intersection problems in plane geometry," *Proc. ACM Symp. on Computational Geometry*, 1985, pp. 135-146.
- Chazelle, B., L. Guibas, and D. T. Lee, "The power of geometric duality," *BIT* **25**, 1985, pp. 76-90.
- Crawford, C., "Aspect graphs and robot vision," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 382-384, 1985.
- Edelsbrunner, H., J. O'Rourke, and R. Seidel, "Constructing arrangements of lines and hyperplanes with applications," *SIAM J. Comput.* **15** (2), 1986, pp. 341-363.
- El Gindy, H. and D. Avis, "A linear algorithm for computing the visibility polygon from a point," *J. Algorithms* **2**, 1981, pp. 186-197.
- El Gindy, H., "An efficient algorithm for computing the weak visibility polygon from an edge in simple polygons," Technical Report, School of Computer Science, McGill University, 1984.
- Fekete, G. and L. Davis, "Property spheres: a new representation for 3D object recognition," in *Proc. Workshop on Computer Vision: Representation and Control*, pp. 192-201, 1984.
- Guibas, L., J. Hershberger, D. Leven, M. Sharir, and R. Tarjan, "Linear time algorithms for visibility and shortest path problems inside simple polygons," *Proc. Second ACM Symp. on Computational Geometry*, 1986, pp. 1-13.

- Ikeuchi, K., "Precompiling a geometric model into an interpretation tree for object recognition in bin-picking tasks," *Proc. IEEE Conf. on Robotics and Automation*, 1986, pp. 321-339.
- Kender, J. and D. Freudenstein, "What is a 'degenerate' view," *Proc. IEEE Conf. on Robotics and Automation*, 1986, pp. 589-598.
- Koenderink, J. and A. van Doorn, "The singularities of the visual mapping," *Biol. Cybernet.* **24**, pp. 51-59, 1976.
- Koenderink, J. and A. van Doorn, "The Internal Representation of solid shape with respect to vision," *Biol. Cybernet.* **32**, pp. 211-216, 1979.
- Korn, M. and C. Dyer, "3D multiview object representations for model-based object recognition," *Pattern Recognition* **20**, pp. 91-103, 1987.
- Lee, D.T., "Visibility of a simple polygon," *Comput. Vision, Graphics, and Image Proc.* **22**, 1983, pp. 207-221.
- McKenna, M. and R. Seidel, "Finding the optimal shadows of a convex polytope," *Proc. IEEE Symp. on Computational Geometry*, 1985, pp. 90-99.
- Plantinga, W. H. and C. Dyer, "An algorithm for constructing the aspect graph," *Proc. 27th Ann. Symp. on Foundations of Comput. Sci.*, 1986, pp. 123-131.
- Plantinga, W. H. and C. Dyer, "The aspect representation," TR 683, Computer Sciences Dept., University of Wisconsin, Madison, 1987.
- Plantinga, W. H. and C. Dyer, "Construction and display algorithms for the asp," TR 735, Computer Sciences Dept., University of Wisconsin, Madison, 1987.
- Rosenfeld, A., "Recognizing unexpected objects: a proposed approach," *Int. J. Pattern Recognition and Artif. Intell.* **1**(1), 1987, pp. 71-84.
- Schneier, M., R. Lumia, and E. Kent, "Model-based strategies for high-level robot vision," *Comput. Vision, Graphics, and Image Proc.* **33**, pp. 293-306, 1986.
- Scott, R., "Graphics and prediction from models," in *Proc. Image Understanding Workshop*, pp. 98-106, 1984.
- Suri, S. and J. O'Rourke, "Worst-case optimal algorithms for constructing visibility polygons with holes," *Proc. Second ACM Symp. on Computational Geometry*, 1986, pp. 14-23.
- Sutherland, I., R. Sproull, and R. Schumacker, "A characterization of ten hidden-surface algorithms," *ACM Computing Surveys* **6**, pp. 1056, 1974.
- Thorpe, C. and S. Shafer, "Correspondence in line drawings of multiple views of objects," *Proc. 8th Int. Joint Conf. on Artif. Intell.*, pp. 959-965, 1983.
- Werman, M., S. Baugher, and A. Gualtieri, "The visual potential: one convex polygon," Tech. Rept. CAR-TR-212, Center for Automation Research, Univ. of Maryland, 1986.

