[McMillan95b] McMillan, L., *"A List-Priority Rendering Algorithm for Redisplaying Projected Surfaces,"* **UNC Technical Report TR95-005,** University of North Carolina, 1995.

[McMillan95c] McMillan, L., and G. Bishop, *"Plenoptic Modeling: An Image-Based Rendering System,"* **Proceedings of SIGGRAPH 95,** (Los Angeles, CA) August 6-11, 1995, pp. 39-46.

[Prazdny83] Prazdny, K., *"On the Information in Optical Flows,"* **Computer Vision, Graphics, and Image Processing**, Vol. 22, No. 9, 1983, pp 239-259.

[Rogers85] Rogers, D.F., **Procedural Elements for Computer Graphics**, McGraw-Hill, New York, NY, 1985.

[Sutherland74] Sutherland, I. E., R. F. Sproull, and R. A. Schumacker, *"A Characterization of Ten Hidden-Surface Algorithms,"* **ACM Computing Surveys**, 6(1), March 1974, 1-55. (Also in J. C. Beatty and K. S. Booth, eds, **Tutorial: Computer Graphics**, Second Edition, IEEE Comp. Soc. Press, Siler Spring, MD, 1982 pp. 384-441)

[Wang90] Wang S.C. and J. Staudhammer, *"Visibility Determination on Projected Grid Surfaces,"* **IEEE Computer Graphics and Applications**, July 1990.

[Wolberg90] Wolberg, G., **Digital Image Warping,** IEEE Computer Society Press, Los Alamitos, CA, 1990.

[Wright73] Wright, T., *"A Two-Space Solution to the Hidden Line Problem for Plotting Functions of Two Variables,"* **IEEE Transactions on Computers**, January 1973.

would be equivalent to the spherical re-projection which has been proven to be occlusion compatible.

Using the simple methodology outlined above, visibility algorithms can be established for any projection manifold that is topologically equivalent to a sphere (i.e. its range function is strictly single valued). While this is the case for all projected surfaces, it does not obviously extend to more general surface representations. This leads to many interesting avenues for future exploration. Consider this: it is a simple matter to represent arbitrary objects in a projective space by arbitrarily choosing some center of projection for them (for example the object's geometric center or the object's center of mass). The resulting representation is essentially the same as the projective mapping, $\tilde{P}_i : R^3_W \rightarrow R^3_{S_i}$, defined in Section 3. If the proposed visibility algorithm could be simply extended to properly handle surfaces of this type, the resulting algorithm would be truly general purpose.

Acknowledgments

This research was done as a course project requirement for COMP 257 in the fall of 1994. The course was taught by Stephen M. Pizer.

5.0 References

[Adelson93a] Adelson, S. J., and L. F. Hodges, *"Stereoscopic Ray-Tracing"*, **The Visual Computer**, Springer-Verlag, 1993, pp. 127-144.

[Adelson93b] Adelson, S. J., and L. F. Hodges, *"Exact Ray-Traced Animation Frames Generated by Reprojection"*, **Technical Report GIT-GVU-93-30**, Georgia Institute of Technology, (Atlanta, GA), July 1993.

[Anderson82] Anderson, D., *"Hidden Line Elimination in Projected Grid Surfaces,"* **ACM Transactions on Graphics**, October 1982.

[Chen93] Chen, S. E. and L. Williams. *"View Interpolation for Image Synthesis,"* **Computer Graphics** (SIGGRAPH'93 Proceedings), pp. 279-288, July 1993.

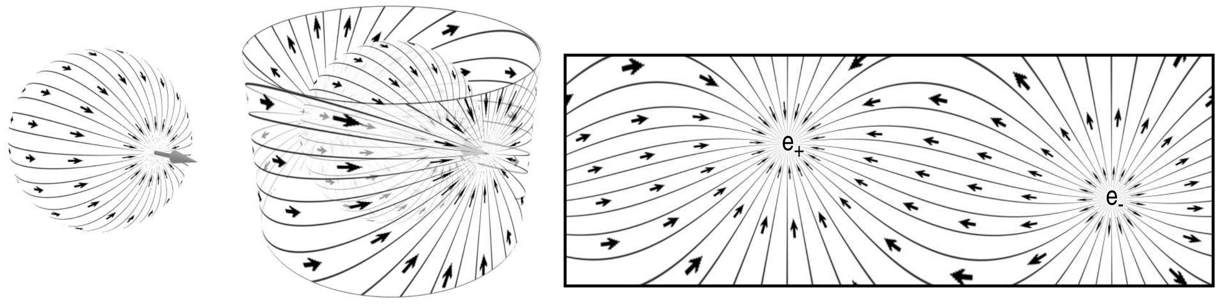
[Laveau94] Laveau, S. and O. Faugeras, *"3-D Scene Representation as a Collection of Images and Fundamental Matrices,"* **INRIA, Technical Report No. 2205**, February, 1994.

[McMillan95a] McMillan, L., and G. Bishop, *"Head-Tracked Stereo Display Using Image Warping,"* **1995 IS&T/SPIE Symposium on Electronic Imaging Science and Technology,** SPIE Proceedings #2409A, (San Jose, CA) February 5-10, 1995, pp. 21-30.

requires that a surface never backtrack upon itself (i.e. it can never overwrite a surface point which has already been written) is essentially the inverse of maintaining only the last surface written at each point on the desired re-projection. Anderson's enumeration order has the advantage that it never generates points that are subsequently overwritten. This makes a lot of sense when considering that his algorithm was developed for use in line plotter applications where, unlike a modern computer graphics display, once ink is drawn on the paper it cannot be overwritten with satisfactory results. This reduction in the number of writes, however, comes at the cost of maintaining the auxiliary data structure to represent the surface boundaries' extents. Depending on the implementation, the representation of the boundary buffer can itself introduce problems. If the extents of the boundaries are represented only along a discrete set of isoparametric lines, then quantization artifacts will be introduced that can create numerical precision problems for future comparisons. If the boundary curves are represented explicitly as line segments, then the storage requirement are potentially very large.

Adelson and Hodges proposed a method for generating stereo pairs from a single planar projective image by re-projecting the right eye's image from the left eye's image. The algorithm they describe is equivalent to the special case of the given visibility algorithm where the epipolar ray is both parallel to the view plane and it is oriented along an isoparametric line of the planar projection. This unique geometry allows for the right eye's image to be generated by enumerating and re-projecting the left eye's image along all scan lines from left to right.

The given visibility algorithm can also be extended to other projection manifolds, using the same procedure of mapping the problem from the original projection manifold to a spherical manifold, re-projecting it, and then mapping it back onto the original surface. The following figure shows this progression with a cylindrical projection manifold.



Once again the longitudinal lines of the spherical projection correspond to the projected images of the epipolar planes that are induced by a change in center of projection in the direction of the epipolar ray. Images of these epipolar planes can be mapped onto a cylindrical projection manifold that shares the same center of projection. The cylindrical projected surface can then be reprojected along these lines of longitude in an order progressing from the negative to the positive epipolar rays. The resulting enumeration

By Theorem 1, topological multiplicities are constrained to occur only within epipolar planes. The projected-spherical flow field is the image of these epipolar planes. If a series of isoparametric lines are mapped in the order determined by the sheet's slope component in the transverse direction to the isoparametric lines, then each point along the epipolar plane will be re-projected in order of decreasing θ. This relative ordering is independent of the *u* or *v* parameter choice. Theorem 2 proves that such an ordering is occlusion compatible. Therefore, the resulting planar enumeration is equivalent to mapping the planar-projected surface onto a spherical manifold and performing the re-projection there. Thus, the planar algorithm given is also occlusion compatible.

The various partitioning and enumeration cases of the planar visibility algorithm all arise from the structure induced on the infinite planar manifold by the parameterization. The choice of the parametric domain determines both the subdivision boundaries as well as the number of sheets. The enumeration direction relative to the epipolar ray's image is determined by whether that image is of the positive or negative epipolar ray. The complementary epipole's image appears at infinity. Thus, the enumeration order of any projective surface, which is topologically equivalent to some spherical region, is always from the negative epipole toward the positive one.

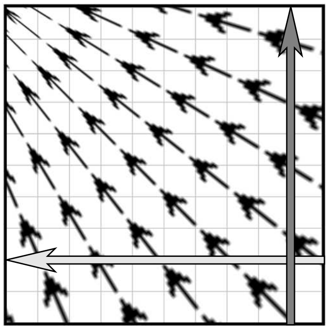
Now, consider the case where the projection plane is parallel to the epipolar ray. Since the projected image of all the epipolar rays onto the planar manifold are parallel, it can be considered a special case of the re-projection where the entire infinite plane consists of a single sheet. The parameterization still determines the enumeration order along isoparametric lines based on the slope of the flow field induced by the image of the spherical frames' lines of longitude. Even though both of the epipolar rays project to infinity on the planar manifold, the direction from the negative to the positive epipolar ray's image is still well defined since all of the epipolar plane images are parallel. The visibility algorithm will handle this correctly if interaction toward epipolar rays at infinity is handled consistently.

4.0 Discussion and Conclusions

Special cases of the visibility algorithm presented in this paper have been discovered previously by Anderson [Anderson82] and Adelson and Hodges [Adelson93a]. Anderson's original hidden-line algorithm handled the special case where the initial surface was defined in an euclidean space rather than a projective one. This corresponds to setting the initial center of projection at infinity. This proves to be a useful representation for bivariate functions of the sort that Anderson was interested in displaying. It is also notable that Anderson's algorithm enumerates the surface in an order opposite of the presented algorithm. This requires a buffer to be maintained where the minimum and maximum extents of the projected surface are stored. Anderson's painting rule, which then the various cases of the visibility algorithm become obvious. If one coordinate of the epipolar ray's image onto the plane falls outside of the parametric domain, then the region is divided into two subregions. If both coordinates fall outside of the parametric domain, then the entire region has a single common slope direction. This last condition corresponds to the single sheet case of the algorithm.

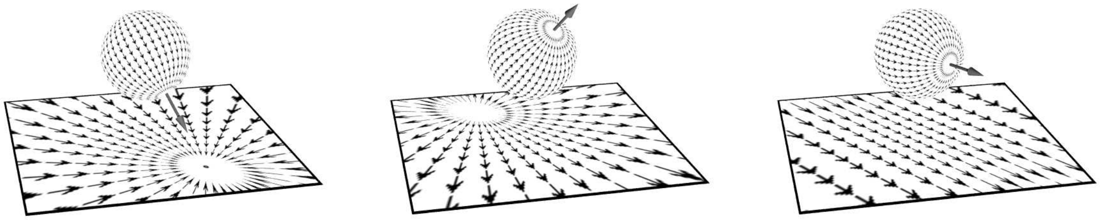
The final step in mapping the spherical-projection-manifold's visibility proof to a planar-projection-manifold is determining the order of enumeration. It is clear that points on the planar surface can be reprojected in the order implied by the projected flow lines (i.e. the image's of the spherical epipolar planes), and the result would be consistent with the spherical re-projection. This result would correspond to mapping the original planar projected-image onto a spherical-projected image with the same center-of-projection, then re-projecting the result onto a new spherical frame. While this approach works, it is inconvenient since it involves extra mapping steps. These extra steps can be avoided if the enumeration order is specified in the parametric space of the original projection manifold, as is done in the visibility algorithm presented. Next we show that the combination of the partitioning and enumeration steps specified by the proposed visibility algorithm are equivalent to mapping along the epipolar planes of the spherical-projected surface.

In the spherical-projection case each longitudinal line corresponding to an epipolar plane was considered independently. In the planar case it is necessary to consider all epipolar planes simultaneously. In the following figure a single sheet is shown with two directed isoparametric lines superimposed.



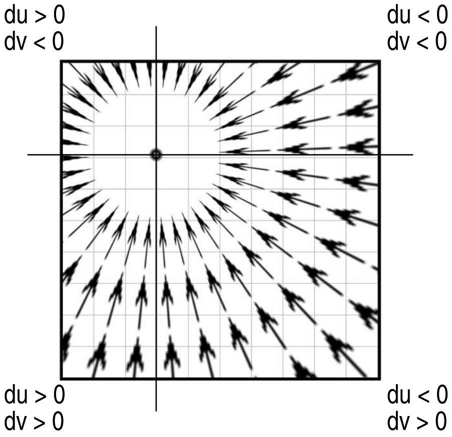
Notice the following properties of these directed isoparametric lines. Each lines' direction is consistent with the spherical flow field's projection onto the corresponding parametric axis. This direction is the same for all isoparametric lines within a given sheet since both components of the flow field's slope have a constant sign. A given isoparametric line will cross the projected image of each epipolar plane exactly once. If a series of these isoparametric lines are enumerated in the direction of the flow field's transverse component, then the overall image of the spherical-projected flow field is preserved. Furthermore, the relative ordering of the two parameters is arbitrary.

ness and negativeness of the epipolar rays are established by the index number of the singularity. Also, observe that the epipolar rays' images may or may not fall within the parametric domain of the planar image. Nonetheless, all images of the epipolar planes either converge to the image of the positive epipolar ray or diverge from the negative epipolar ray's image. It can easily be shown that there are only three possible configurations of the epipolar rays in relation to the planar projection manifold. The infinite plane of projection is intersected by either the positive or the negative epipolar ray, or it is parallel to both. Therefore, the set of epipolar planes, whose spherical images correspond to the longitudinal lines of the sphere, will, on a planar manifold, map to directed lines that either all converge, all diverge, or are parallel as determined by the projected image of the epipolar rays (see figure below).



Initially, we will consider only the two cases where one of the epipolar images intersects the projection manifold.

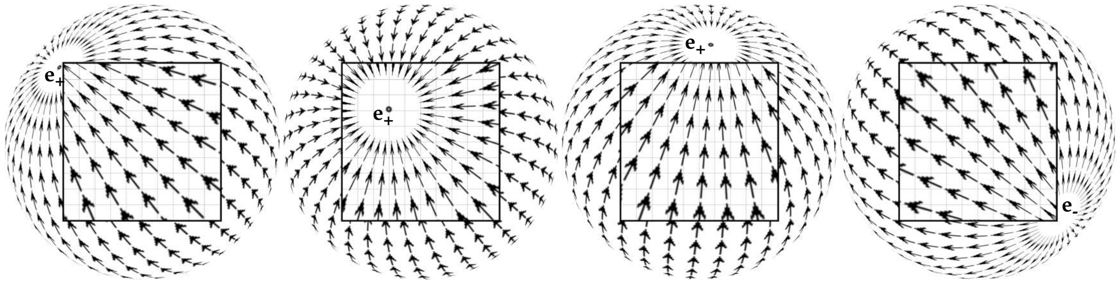
The second step of the visibility algorithm divides the manifold along isoparametric lines that are defined by the coordinates of the epipolar ray's image. If the projected epipolar ray's coordinates falls within the parametric domain of the planar image, this partitioning step divides the plane into four regions. When the epipolar planes' images converge or diverge, there exists exactly two epipolar planes that coincide with isoparametric lines. These planes define boundaries across which the vector field defined by the projected epipolar planes changes sign in slope.



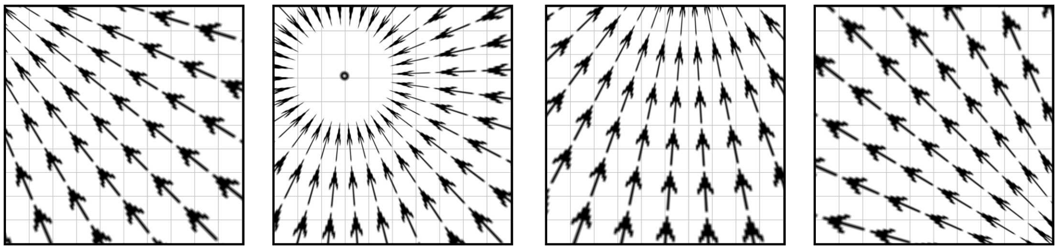
If we consider the subdivision of the manifold to be based on the change of sign in the slope of the epipolar plane images when mapped into the parametric coordinate system, rays described by a center of projection and a planar projection manifold are a subset of those rays given by a spherical manifold with the same center.

The solid angle of this bundle can never be more than 2π, and this maximum occurs only when the projection plane is unbounded. We can, therefore, map any planar-projected surface onto a subset of a spherical-domain with a re-parameterization of the plane. Once the planar-projected surface is so mapped, the re-projection to a desired center of projection can proceed along epipolar planes in order of decreasing θ from the negative epipole towards the positive epipole as described previously. The resulting spherical mapping can then be re-parameterized to match the desired planar projection manifold.

We can also consider this series of steps in its entirety. In the figures shown below, the initial surface of projection is shown overlaying the equivalent spherical projection surface. The enumeration direction implied by the change in center of projection is also overlaid on both surfaces. The enumeration direction vector field shown on the planar surface is a projection of the field seen on the sphere. Notice that the field lines which correspond to epipolar planes map onto lines in their planar image. This is to be expected since the intersection of any two planes in the generic case is a line.



In the following images the enumeration direction fields are shown without the underlying spherical map. Isoparametric lines of the planar domain are also shown for each of the planar projection manifolds.



Consider the relationship of these images to the visibility algorithm given in Section 2. In the first step of the algorithm, the projection of the desired viewpoint onto the initial image is computed. By definition, this projection lies upon one of the two epipolar rays. Observe that these points are singularities of the vector field. In retrospect, the positive-

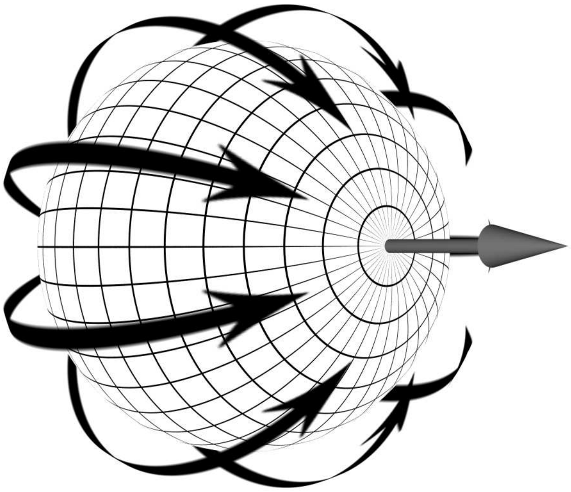
$$\sin\alpha\cot\theta_1 - \cos\alpha > \sin\alpha\cot\theta_2 - \cos\alpha, \qquad \text{(subtraction from both sides)}$$

$$1 > \frac{\sin\alpha\cot\theta_2 - \cos\alpha}{\sin\alpha\cot\theta_1 - \cos\alpha} \qquad \text{(since both sides are strictly positive)}$$

Therefore, $\frac{a}{b} < 1$ and $a < b$. For angles θ_1 and θ_2 within the interval $[0,\pi]$ the relationship $\cos\theta_1 > \cos\theta_2$ also implies that $\theta_1 > \theta_2$.

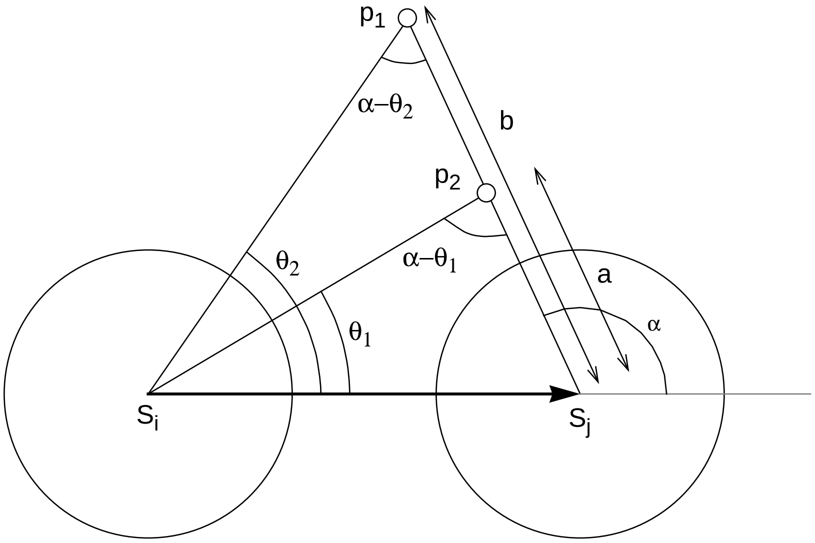
Next, consider all of the surface points of V_i whose rays fall along a common epipolar plane. If, during the re-projection from S_i to S_j, each point is processed in order of decreasing θ then, by Theorem 2, when a multiplicity occurs along a given ray defined on S_j, the most recently processed point will always have a closer range value than any point processed previously. We can, therefore, compute an occlusion compatible mapping, $\tilde{V}_{ji}(\tilde{x}_i)$, by mapping in order of decreasing θ and allowing later surface points to overwrite previous ones. Notice that this mapping procedure only considers the positions of the initial and desired centers of projection, and makes no use of the range information itself.

Since all surface interactions are constrained to take place within a single epipolar plane, we can compute all such planes independently. We then arrive at an algorithm for computing an entire occlusion-compatible surface from the re-projection of any other topologically consistent projective surface. The resulting surface will also be topologically equivalent to a sphere since only a single surface element is retained along each ray. The resulting re-projection order is depicted in the figure shown below.



3.3 Mapping to a planar manifold

Next, consider the implications of this result to a planar-projected surface. The bundle of shown in the following figure.



Theorem 2. *If* $\cos\theta_1 > \cos\theta_2$ *and* $0 < \theta_1, \theta_2, \alpha < \pi$ *then* $a < b$.

Proof: The length of sides a and b can be computed in terms of the angles θ_1, θ_2, and α using the law of sines as follows.

$$\frac{a}{\sin\theta_1} = \frac{1}{\sin(\alpha - \theta_1)} \qquad \frac{b}{\sin\theta_2} = \frac{1}{\sin(\alpha - \theta_2)}$$

Thus,

$$a = \frac{\sin\theta_1}{\sin(\alpha - \theta_1)} \quad \text{and} \quad b = \frac{\sin\theta_2}{\sin(\alpha - \theta_2)}$$

$$a = \frac{\sin\theta_1}{\sin\alpha\cos\theta_1 - \cos\alpha\sin\theta_1} \qquad b = \frac{\sin\theta_2}{\sin\alpha\cos\theta_2 - \cos\alpha\sin\theta_2}$$

$$a = \frac{1}{\sin\alpha\cot\theta_1 - \cos\alpha} \qquad b = \frac{1}{\sin\alpha\cot\theta_2 - \cos\alpha}$$

Note that the denominators in the expressions of a and b must be strictly positive, since their lengths are positive over the range of angles defined. Furthermore, the ratio of a to b can be expressed as:

$$\frac{a}{b} = \frac{\sin\alpha\cot\theta_2 - \cos\alpha}{\sin\alpha\cot\theta_1 - \cos\alpha}$$

The given relationship

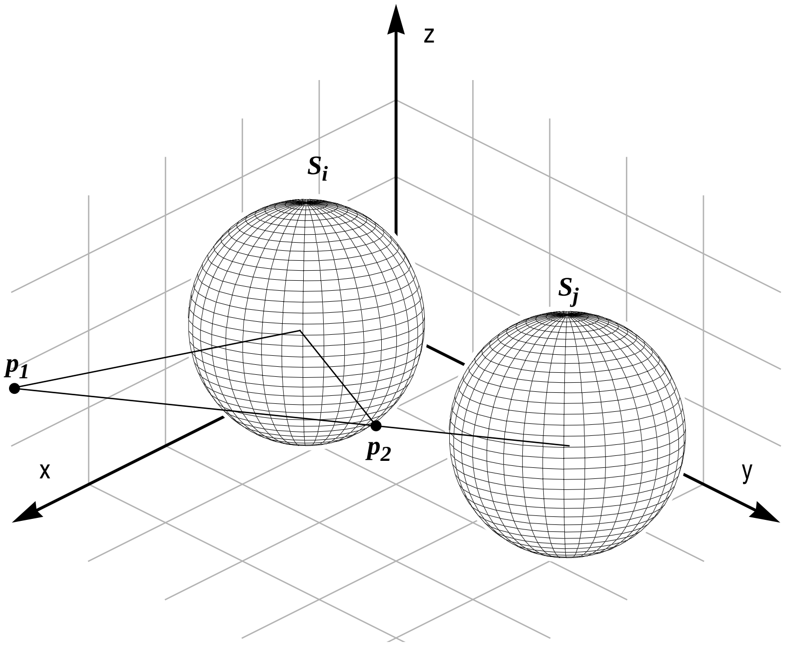
$$\cos\theta_1 > \cos\theta_2,$$

implies

$\cot\theta_1 > \cot\theta_2$, (since $\cos\theta$ and $\cot\theta$ are monotonically decreasing)

$\sin\alpha\cot\theta_1 > \sin\alpha\cot\theta_2$, ($\sin\alpha$ is always positive over the interval)

S_j, as shown below:



Theorem 1. *Generically, the points of a topological multiplicity induced by a mapping from* S_i *to* S_j, $\tilde{P}_{ji}(\tilde{x})$*, and the two centers of projection lie in a plane.*

Proof: The points of the topological multiplicity are colinear with the origin of S_j since they share the same parametric coordinates. A second line segment connects the local frame origins, S_i and S_j. In general, these lines are distinct and, since they share a common point, they must be coplanar.

Notice that this plane of the multiplicity also corresponds to an epipolar plane. From Theorem 1 we can conclude that any multiplicity induced by a projective re-mapping takes place within an epipolar plane that is common to all of those rays whose surface positions take part in the multiple mapping. Thus, all interactions between surface points, when viewed from the desired center of projection, are constrained to occur within epipolar planes.

Having established the importance of these epipolar rays and planes, it is convenient to re-orient the local spherical frames so that the epipolar planes fall upon isoparametric curves. This can be accomplished with a global change of coordinate systems describable by a transformation, **A**, of *W*. The single affine transform, **A**, can be constructed to achieve all of the following results:

- Translate S_i to the origin
- Rotate S_j to lie on the x-axis
- Rotate the epipolar plane of the multiplicity into the xy-plane with a rotation about the x-axis
- Scale the system so that S_j has the coordinate (1,0,0).

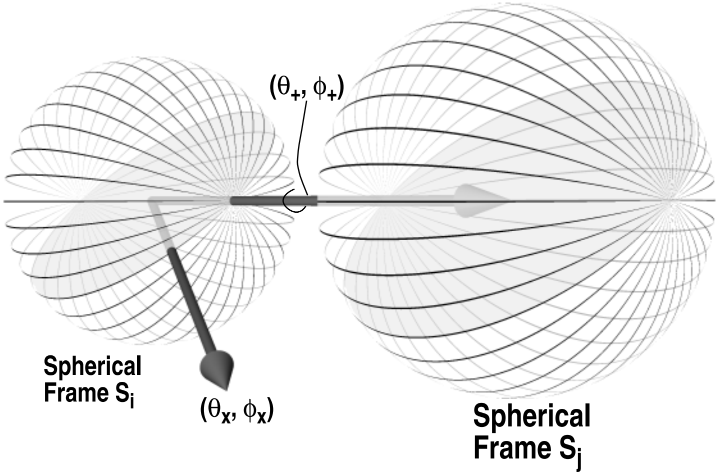
With this transformation we can consider the multiplicity entirely within the xy-plane, as

This class contains, as a subset, all perspective-projected surfaces. In a later section, I will discuss the potential for extending these results to a wider class of surface descriptions that allow for non-topologically consistent surfaces.

3.2 Derivation

Several important relationships exist between projective mappings. Embedded spherical frames, like those just defined, will be used to represent all aspects of a projective surface. The origin of the spherical frame will serve as the center of projection while the unit sphere will be used as the projection manifold. The initial projected surface is represented by the $\boldsymbol{S}_{\mathrm{i}}$ frame while the desired re-projection is represented by $\boldsymbol{S}_{\mathrm{j}}$.

In general, all possible re-projections, $\tilde{P}_{ji}$, of points falling along a given ray, (θ_x, ϕ_x), of $\boldsymbol{S}_{\mathrm{i}}$ are constrained to lie within a plane that is defined by the line segment, *l*, connecting the initial and desired centers of projection, and the ray itself. There are two exceptions to this observation. On a spherical projection manifold there exists one ray in the direction of the line segment, *l*. This ray, (θ_+, ϕ_+), which is called the *positive epipole*, is the virtual image of the desired center of projection on the original surface. A second ray, called the *negative epipole* (θ_-, ϕ_-) lies in the opposite direction. It corresponds to the vanishing point for all rays originating from $\boldsymbol{S}_{\mathrm{j}}$ in the direction of $\boldsymbol{S_i}$ when seen in projection. Points along either of the epipolar rays are constrained to lie on a line which has *l* as a segment. This line is also common to all of those constraint planes induced by the re-projection of other rays. For this reason these constraint planes will henceforth be called *epipolar planes*[1]. The images of these epipolar planes correspond to the lines of longitude for a sphere with the epipoles as poles. These relationships are shown in the following figure.



Next, consider an isolated topological multiplicity on the projective mapping from $\boldsymbol{S}_{\mathrm{i}}$ to

[1] It is common in the computer vision literature to call the intersections of the epipolar planes with the projection manifold *epipolar curves*. Likewise, the images of the epipolar rays on the projection manifold are often called an *epipolar points*.

$$r_j(\tilde{x}_i) = \min(\{\rho_j(\tilde{y}_i) | \theta_j(\tilde{y}_i) = \theta_j(\tilde{x}_i), \phi_j(\tilde{y}_i) = \phi_j(\tilde{x}_i)\}).$$

The following figure depicts the relationships between these mappings.

$$\begin{tikzcd}
R^3_{S_j} \arrow[r, shift right, "\tilde{P}_j^{-1}"'] \arrow[d, "r_j"'] & R^3_W \arrow[l, shift right, "\tilde{P}_j"'] \arrow[r, shift left, "\tilde{P}_i"] & R^3_{S_i} \arrow[l, shift left, "\tilde{P}_i^{-1}"] \arrow[d, "r_i"] \\
M_{V_j} \subseteq R^3_{S_j} \arrow[ur, "\tilde{P}_j^{-1}"] & & M_{V_i} \subseteq R^3_{S_i} \arrow[ul, "\tilde{P}_i^{-1}"']
\end{tikzcd}$$

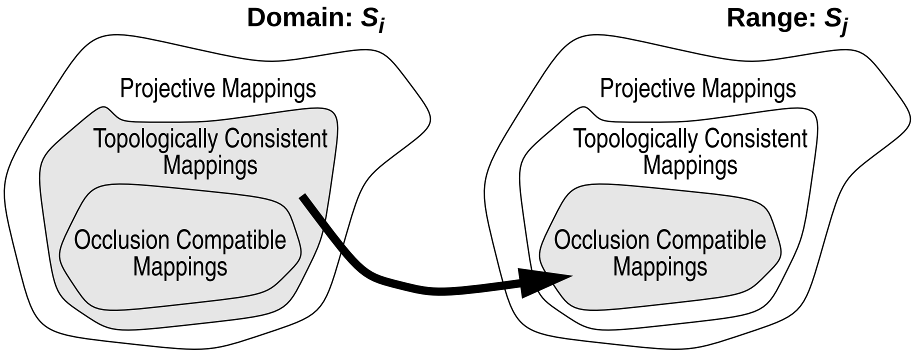
It has been claimed elsewhere [McMillan95c] that general two-dimensional mappings between spherical frames can be determined without an explicit representation of the range information, $\rho(\tilde{x})$ or $r(\tilde{x})$, as shown below:

$$\dot{P}_{ji}(\dot{x}) : R^2_{S_i} \to R^2_{S_j} = \begin{bmatrix} \theta_j(\theta_i, \phi_i, \delta_j(\theta_i, \phi_i)) \\ \phi_j(\theta_i, \phi_i, \delta_j(\theta_i, \phi_i)) \end{bmatrix} \qquad \text{where} \qquad \delta_j(\theta_i, \phi_i) = -\delta_i(\theta_j, \phi_j).$$

Unlike range information, the generalized disparity term, $\delta(\dot{x})$, can be determined directly from observable image features, such as optical flow [Prazdny83] or point correspondences [Laveau94].

These two-dimensional mapping functions, $\dot{P}_{ji}(\dot{x})$, may result in several points from the domain mapping to a single point in the range. Such range points will be referred to as *topological multiplicities* of the mapping. The specific claim of this report is that an occlusion compatible mapping can also be established independent of range information.

The conversion of a general projective mapping to an occlusion compatible one involves an inherently non-linear selection process. In our definitions, this selection was made on the basis of range information. I will show that the same result can be determined via an appropriately-ordered enumeration over the parameter space of the domain's spherical frame, assuming that it is topologically consistent. While this assumption limits the generality of this approach, it is valid for the important case of those occlusion compatible mappings within the domain.


Domain: S_i
Range: S_j
Projective Mappings
Topologically Consistent Mappings
Occlusion Compatible Mappings
Projective Mappings
Topologically Consistent Mappings
Occlusion Compatible Mappings

that all possible perspective projections are consistent with this scene in terms of both visibility and the positions of the surface points within ***W***. For each spherical frame there exists a mapping of these scene points from ***W*** into the local coordinate system ***S***$_i$,

$$\tilde{P}_i : R^3_W \rightarrow R^3_{S_i} \qquad \text{by} \qquad \tilde{P}(x, y, z) = \begin{bmatrix} \rho_i(x, y, z) \\ \theta_i(x, y, z) \\ \phi_i(x, y, z) \end{bmatrix}$$

While the mapping of points from ***W*** to ***S***$_i$ is one-to-one, it is not generally a single-valued range-function when parameterized in θ and ϕ. Therefore, we are interested in a subset of this mapping where only a single value of ρ_i is defined for each unique pair of θ and ϕ. Such mappings are known as topological meshes. An example of one such unique mapping is defined by

$$\tilde{V}_i : R^3_W \rightarrow R^3_{S_i} \qquad \text{by} \qquad \tilde{V}_i(x, y, z) = \begin{bmatrix} r_i(x, y, z) \\ \theta_i(x, y, z) \\ \phi_i(x, y, z) \end{bmatrix}$$

and

$$r_i(\bar{x}) = \min(\{\rho(\bar{y}) | \theta_i(\bar{y}) = \theta_i(\bar{x}), \phi_i(\bar{y}) = \phi_i(\bar{x})\}).$$

This particular mapping will be called the *occlusion compatible mapping*, and it is consistent with a proper visibility solution as defined in previous sections.

Next, we will consider both general mappings and occlusion compatible mappings[1] between pairs of spherical frames. For example,

$$\tilde{P}_{ji} : R^3_{S_i} \rightarrow R^3_{S_j} \qquad \text{by} \qquad \tilde{P}(\rho_i, \theta_i, \phi_i) = \begin{bmatrix} \rho_j(\rho_i, \theta_i, \phi_i) \\ \theta_j(\rho_i, \theta_i, \phi_i) \\ \phi_j(\rho_i, \theta_i, \phi_i) \end{bmatrix}$$

and

$$\tilde{V}_{ji} : R^3_{S_i} \rightarrow R^3_{S_j} \qquad \text{by} \qquad \tilde{V}_{ji}(\rho_i, \theta_i, \phi_i) = \begin{bmatrix} r_j(\rho_i, \theta_i, \phi_i) \\ \theta_j(\rho_i, \theta_i, \phi_i) \\ \phi_j(\rho_i, \theta_i, \phi_i) \end{bmatrix}$$

where

[1] Two different types of occlusion compatible mappings between spherical frames could be defined: those where the domain, S_i, is a general mapping, or those where the domain is occlusion compatible. Since the latter is a proper subset of the former only the general case is considered in this definition. However, later we will deal primarily with the case where the domain is also an occlusion compatible mapping.

3.0 Proof of Algorithm's Correctness

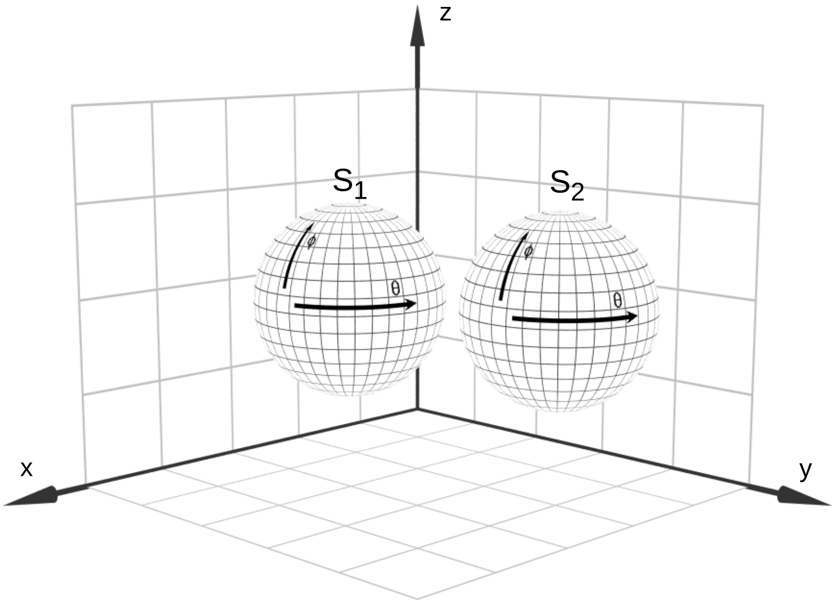
In order to prove the claims of the visibility algorithm discussed, we will generalize the planar-projection manifold to a spherical manifold. This allows for a single parameterization that encompasses all rays originating from the surface's center of projection. This generalization has two benefits. First, there will always be a positive image of the desired new center, thus halving the number of enumeration cases. Second, since the projected image of the desired center of projection will always fall within the parametric domain, the remaining nine cases are all collapsed into one.

Once the algorithm is shown to be correct for the spherical case, the result will be transferred back to the case of planar-projective surfaces. Then, an overview of how the algorithm can be extended to other projection manifolds is given.

3.1 Definitions

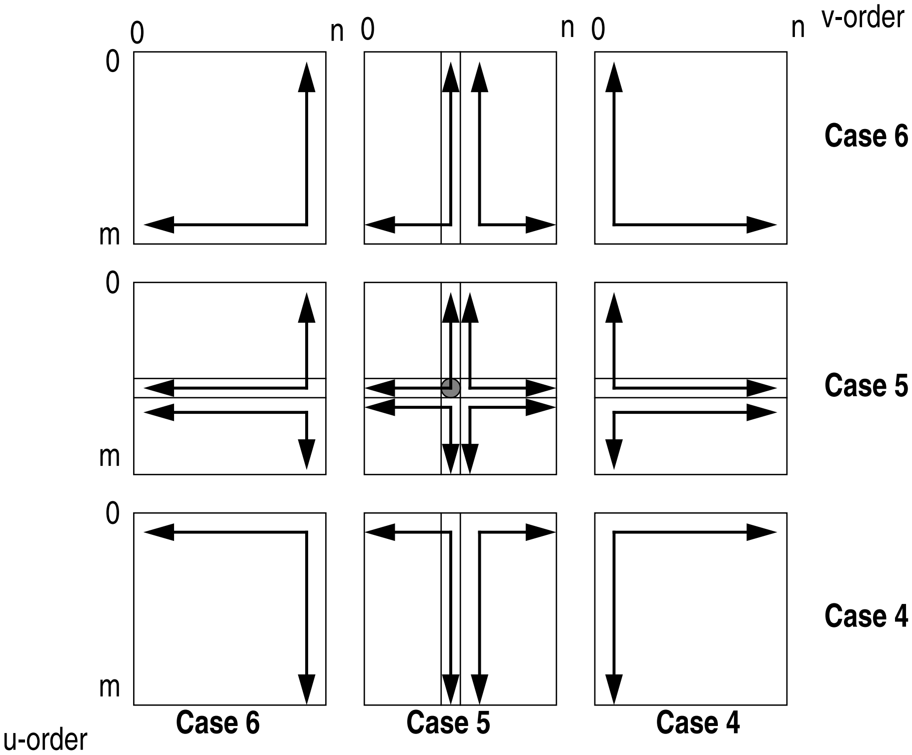
As stated previously, a perspective-projected surface is defined by a central point and a single-valued range function over some bundle of rays emanating from that point. The bundle of rays is defined over the parameter space of a projection manifold, and the central point is called the center of projection. For any perspective-projected surface it is useful to consider the equivalent surface that arises when the ray bundle is mapped onto a spherical manifold with the same center of projection. Within such a framework all the possible rays can be described by a single parameter space.

In this derivation we consider sets of local spherical coordinate frames, $\{S_1, S_2, \ldots, S_n\}$ that are embedded within a global cartesian coordinate system, W.



This analysis assumes the existence of a set of points, called a scene, that are defined over the global coordinate frame, W, and independent of any local frame, S_i. We also assume the projection manifold with a positive range value. In these cases the sheets are enumerated toward the image of the desired center. Multiple sheets are again introduced when the desired point falls within the unit square of the surface's parameter space. We define the dominate parameter as the one that enumerates isoparametric lines of the other parameter. Since the enumeration of the re-projection can be considered as two nested loops over the points on the surface, the dominate parameter is the one enumerated in the outer loop. Both the sheet ordering and the choice of the dominate enumeration parameter are arbitrary, under the same conditions given for the one-parameter case.

The second group of enumeration cases arises when the desired center of projection projects as a negative image onto the surface's projection manifold. These nine cases are shown below.

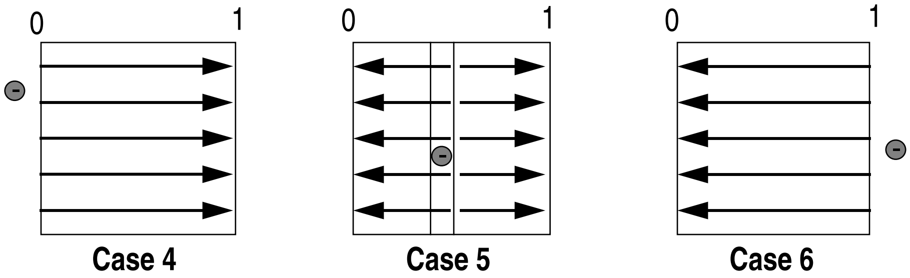


The projected surface's points are mapped along isoparametric curves that are progressively farther away from the projected image of the desired viewpoint, toward the positive image at infinity. Multiple sheets arise as before, and their relative re-projection order is independent under the conditions given previously.

In review, the algorithm described claims to compute the correct visibility for the perspective-projected surface that arises from the re-projection of another perspective-projected surface. This is done by establishing an enumeration, or painting, order which is based solely on the relative positions of the two centers of projection and the projection manifold of the original surface, and this order is independent of the surface's range function.

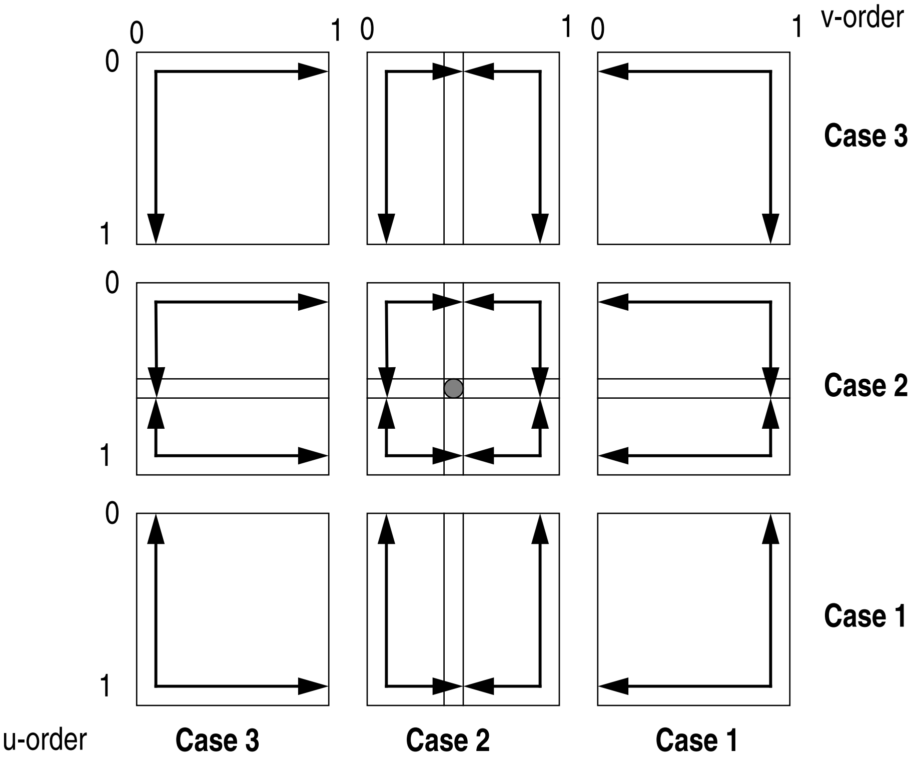
arbitrary.

In cases 4 through 6 the projected image of the desired center of projection is considered to be a negative image since the ray originating from the initial center of projection and passing through the desired one does not intersect the projection manifold. In these cases a point at infinity is treated as the desired center's positive image. The partitioning process will be the same as the positive image case; however, the enumeration order will be reversed as shown below.

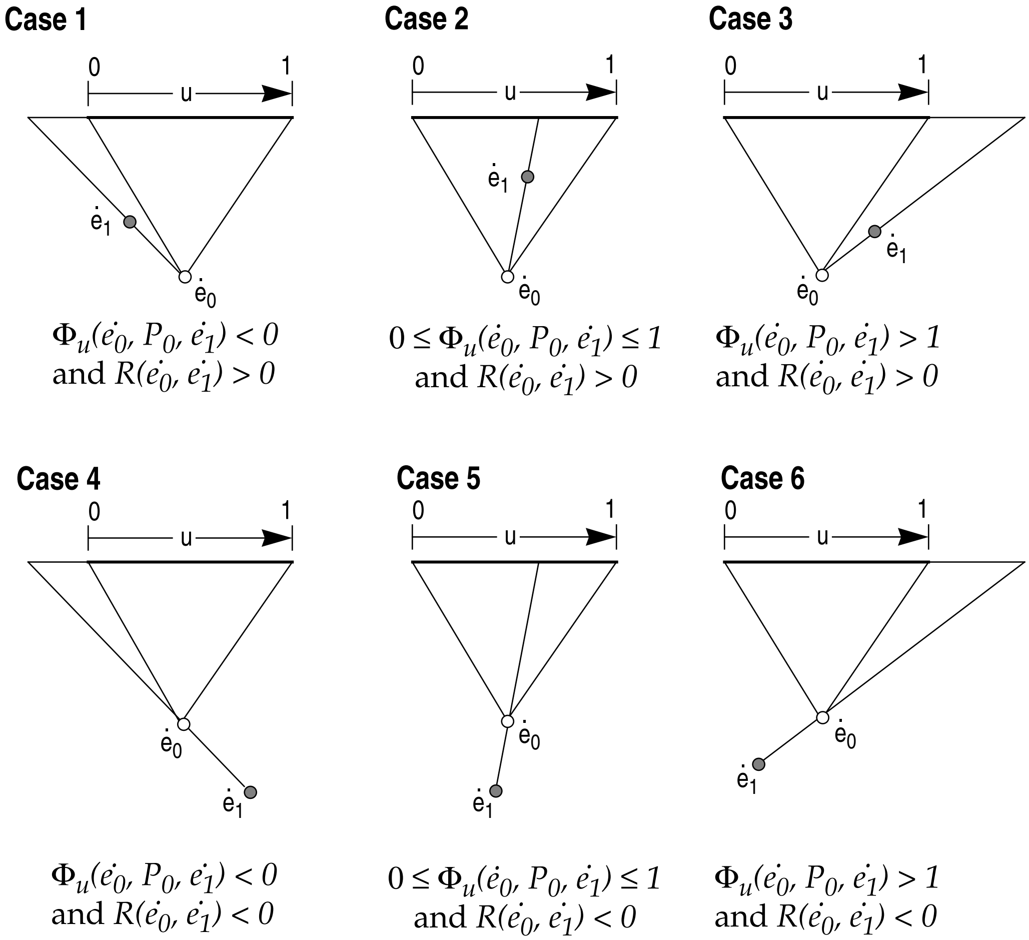


Once again, Case 5 divides the surface into multiple sheets which can be projected in an arbitrary order as long as the sheet containing the negative image of the desired view is projected first.

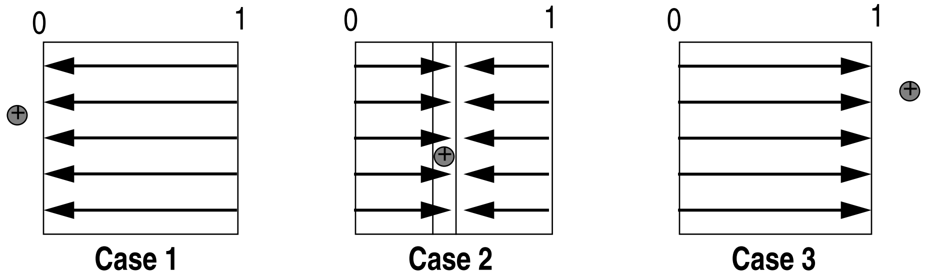
When both parameters of the projection manifold are considered, eighteen different enumeration cases arise. These cases can be considered in two groups.



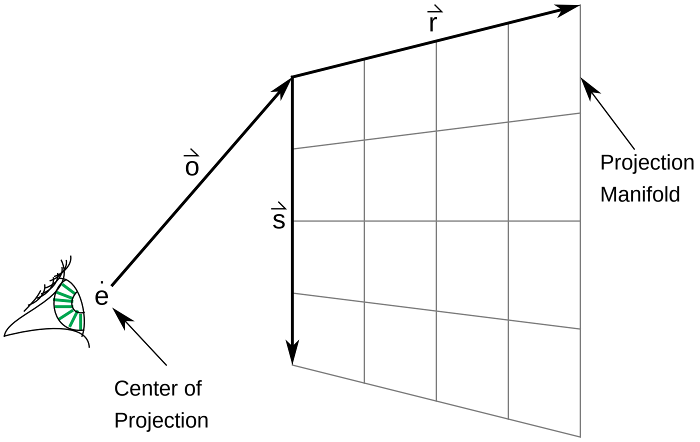
The first group, shown above, gives the painting order for all the cases where the ray from the initial surface's center of projection to the desired center of projection intersects parameter.


Case 1
Case 2
Case 3
0
1
u
$\dot{e}_1$
$\dot{e}_0$
$\Phi_u(\dot{e}_0, P_0, \dot{e}_1) < 0$ and $R(\dot{e}_0, \dot{e}_1) > 0$
$0 \le \Phi_u(\dot{e}_0, P_0, \dot{e}_1) \le 1$ and $R(\dot{e}_0, \dot{e}_1) > 0$
$\Phi_u(\dot{e}_0, P_0, \dot{e}_1) > 1$ and $R(\dot{e}_0, \dot{e}_1) > 0$
Case 4
Case 5
Case 6
$\Phi_u(\dot{e}_0, P_0, \dot{e}_1) < 0$ and $R(\dot{e}_0, \dot{e}_1) < 0$
$0 \le \Phi_u(\dot{e}_0, P_0, \dot{e}_1) \le 1$ and $R(\dot{e}_0, \dot{e}_1) < 0$
$\Phi_u(\dot{e}_0, P_0, \dot{e}_1) > 1$ and $R(\dot{e}_0, \dot{e}_1) < 0$

According to the algorithm's third step, the order of projection for the original surface is toward the positive parametric image of the desired center of projection. For the first three cases, this order is shown in the following figure, where the image of the desired center of projection is shown as a darkened circle.



Case 2 illustrates the case where the projection manifold is partitioned into multiple sheets. The only ordering constraint between sheets is that the sheet containing the image of the desired center-of-projection be rendered last. Since the center-of-projection's image can be made part of either of the two sheets, the sheet ordering is in fact the figure shown below.



We define the projection of a point, x, by two functions $\Phi_i(\dot{e}_i, P_i, x)$ and $\rho_i(\dot{e}_i, x)$, where the parametric mapping function Φ_i determines the parameter space coordinate of the ray passing from the center of projection to the point x, and the range function, ρ_i, gives the distance along this ray to x.

An algorithm for determining the correct visibility of a planar-projected surface when re-projected to an alternate viewpoint, $\dot{e}_1$, is given as follows:

1) Find the projection of the desired viewpoint, $\dot{e}_1$, on the surface to be reprojected $\Phi_0(\dot{e}_0, P_0, \dot{e}_1)$.

2) Partition the manifold into one, two, or four sheets within the parametric domain, according to the isoparametric lines given by $\Phi_0(\dot{e}_0, P_0, \dot{e}_1)$.

3) Enumerate each partition along sequential isoparametric lines toward the positive parametric image of the point, $\dot{e}_1$, while re-projecting onto the new domain Φ_1.

The visibility of the projected surface is determined by the combination of the partitioning and enumeration steps. Essentially these steps establish a visibility-compatible ordering. By this we mean any set of points that map to the same parametric coordinate in the desired view will be ordered such that the last point reprojected will be the one closest to the new center of projection. This ordering is established without any reference to the range values of the surface. Therefore, if the re-projection stage of the third step can also be accomplished without explicit reference to the range function, as in [McMillan95a] and [McMillan95c], the projected scene's visibility can be established without any depth information.

Next, several examples are given to illustrate the various enumeration directions used by the algorithm. In order to simplify the case analysis, the two parametric directions are treated independently. The following figure demonstrates the six different ways that the desired viewpoint can project onto the re-projected surface considering only the u

given by, $\dot{e}$, the center of projection[1]. The range function, $R(u, v)$, specifies a length for each ray and is defined over the same parameter space as the projection manifold. Therefore, a perspective-projected surface is itself a parameterized manifold defined over u and v as shown below.

$$S(u, v) \;=\; \dot{e} + R(u, v)\frac{P(u, v) - \dot{e}}{|P(u, v) - \dot{e}|}$$

We will consider the problem of re-projecting a perspective-projected surface. This means that, given an initial perspective-projected surface, we would like to generate a consistent new surface with a different origin and/or projection manifold. One difficulty of the re-projection arises from the fact that only a single range value is defined for each ray. However, the re-projection process allows for any number of points from the initial surface to fall along a ray. We resolve this ambiguity by defining the notion of visibility. Visibility dictates that the appropriate choice for selecting one from a number of surface points that map to the same ray is to choose the closest one to the desired projection's origin. If we consider the surface to be composed of an opaque material, then visibility corresponds to the natural visual phenomenon of occlusion where far away surfaces along a given ray are hidden by closer surfaces.

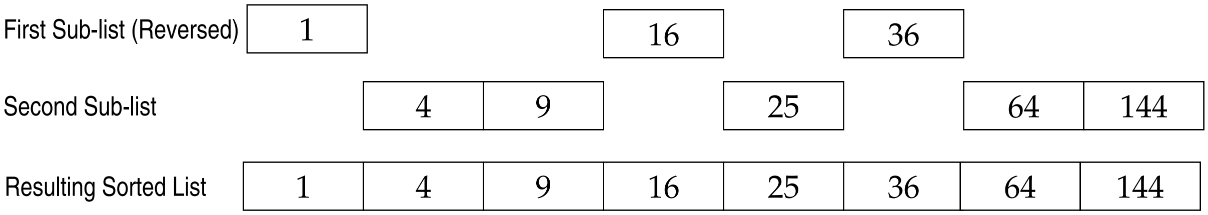
One might ask, why should we restrict ourselves to such limited surface representations in the first place? The primary motivation for this choice is that nearly all natural image formation processes directly record perspective-projected surfaces. This includes photographic images, video images, and the images processed by our visual systems. The goal of re-projecting images is to generate alternate views which correspond to different view positions.

2.1 Algorithm for determining correct visibility

Next, an algorithm for re-projecting planar perspective-projected surfaces is given [McMillan95b]. First, several quantities will be defined in order to describe the algorithm. The original perspective-projected surface is defined by an initial center of projection, $\dot{e}_0$, and a planar projection matrix, P_0. It is useful to consider the columns of the projection matrix as the independent vectors $P_0 \;=\; \begin{bmatrix}\vec{r}_0 & \vec{s}_0 & \vec{o}_0\end{bmatrix}$. We can then consider P_0 as a ray-generating function over the parametric domain, $\{\,(u, v)\,|\,u, v \in\; [0, 1]\,\}$, where the vector $\vec{o}_0$ is from the center of projection to the parametric origin (i.e. $(u, v) \;=\; (0, 0)$) of the planar-projection manifold, and, the remaining two vectors, $\vec{r}_0$ and $\vec{s}_0$, span the plane over the parametric domain. These relationships are depicted in

[1] We disallow the case where the center of projection lies on the projection manifold.

final sorted list by combining these sub-lists using a list merging algorithm similar to the one used by merge sort. This approach requires fewer operations than a general resorting after the perturbation. For example, consider the global perturbation defined by the mapping function x^2 *- 18x + 81*. This function has a single minimum at the point *x = 9*. For values of *x* less than *9* the function decreases monotonically as *x* increases, while for greater values it increases monotonically. The two sub-lists can be merged into a single sorted list in linear time as shown below. Likewise, the minimum value in the list can be determined in constant time by comparing the first values in each sub-list.



The visibility algorithm derived in this paper bears a close resemblance to the incremental sorting algorithm just described. It begins with a scene representation that can be considered as an initial sorting. Any subsequent reprojections of the scene act as global perturbations with easily established extrema. These extrema partition the representation into sub-parts whose relative ordering is unaffected by the perturbation. These sub-parts can then be merged into a final result exhibiting a correct visibility solution. In addition, special attributes of the representation, the global perturbation, and the re-projection process, allow the merging to be accomplished without referencing the actual list values.

This paper is organized as follows. First, an algorithm is given for computing the correct visibility of a perspective-projected image as it undergoes arbitrary reprojections. Both the original and its reprojections are assumed to be planar. Next, a proof is given for the correctness of this algorithm. Initially, the proof is shown for spherical projections. Finally, the proof for the spherical case is mapped to the planar case.

2.0 Visibility of Projected Surfaces

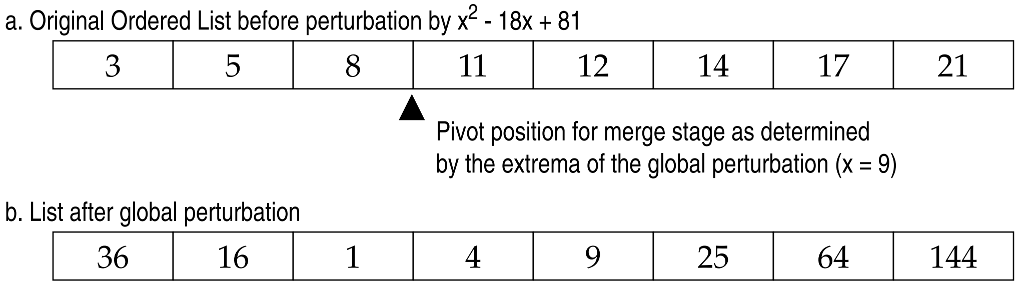
A perspective-projected surface is distinguished by an origin, a projection manifold, and a scalar range function. This origin, $\dot{e}$, which is often called the *center of projection*, can be considered as the origin of the set of all rays[1] that form the projection. It is convenient to consider the projection manifold, P, as a parameterized surface defined over a space with one less dimension than that of the manifold itself. For the sake of discussion, only three-dimensional manifolds with a parameterization over two-dimensions will be considered. Thus, every point of the manifold, $P(u, v)$, defines a specific ray whose origin is

[1] More concisely a “*pencil*” of rays.

are maintained for various regions of the output image. The geometric properties of scene elements are used to both define these list regions, as well as to simplify their management. If one is willing to tolerate sampling artifacts, then these list regions can be defined to the smallest resolvable image element (a.k.a. pixel), making the region definition independent of the geometry of the scene's elements. You need only to add to Sutherland and Sproull's original analysis the distinction between forward-mapping (i.e. for each primitive, figure out which image regions are effected) and inverse-mapping (i.e. for each screen region, figure out what scene primitives belong in the list) to include all of the known visibility approaches to date[1].

1.2 Incremental Sorting

An obvious way of simplifying a sorting problem is to retain the list elements in some naturally ordered data structure. For example, many volumetric rendering techniques compute visibility by exploiting the ordering imposed by the underlying representation. This class of techniques, which I will call incremental sorting, begins with an initially sorted list that undergoes either a global or a local perturbation. In the case of a global perturbation, knowledge of the initial ordering is exploited to simplify the subsequent sorting process. For example, consider the initial ordered list shown in the following figure.



a. Original Ordered List before perturbation by $x^2 - 18x + 81$

3	5	8	11	12	14	17	21

Pivot position for merge stage as determined by the extrema of the global perturbation (x = 9)

b. List after global perturbation

36	16	1	4	9	25	64	144

Once a global perturbation is applied, the resulting list can be partitioned into sub-lists that are separated by the extrema of the perturbation function. In the regions that lie between the extrema, the perturbing function is either monotonically increasing or decreasing. Since the initial sorted list is analogous to the number line, we can generate a

[1] In the intervening time since Sutherland and Sproull's original insight, the theoretical algorithm community has more finely honed the distinctions between various sorting problems. In today's language, visibility would more accurately be considered as an order statistic calculation problem rather than a pure sorting problem (assuming that all surfaces considered are opaque; once multiple levels of transparency are allowed, then computing visibility reverts back to a pure sorting problem). The distinction between computing an order statistic problem and the pure sorting problem is that the final output is composed of only one list element. Typically, we are only interested in the single scene element that is visible for a given screen region, rather than a list of all the scene elements that project onto the region. The relevance of this distinction is that the order-statistic problem is known to require fewer operations, ***O(n)***, than the general sorting problem, ***O(****n* log *n****)***. Thus, we can expect the cost of computing visibility to be linear in the number of scene primitives.

Computing Visibility Without Depth

Leonard McMillan
Department of Computer Science
University of North Carolina,
Sitterson Hall, Chapel Hill, NC 27599
email: mcmillan@cs.unc.edu

1.0 Introduction

A fundamental problem in computer graphics is the determination of the visible subset of surface elements within a scene given the desired viewpoint. Often, the most appropriate method for computing visibility is dependent on the form and properties of the scene's description. For example, if the geometry of a scene is described by a collection of opaque convex planar regions (i.e. triangles), then a simple depth buffer can be used to determine the correct visibility. However, if the scene is described as a level-set of some implicit function (i.e. ellipsoid), then a ray-casting algorithm might be best.

This paper presents a visibility solution for a specific type of scene description, called a *perspective-projected surface*. Surfaces of this type are commonplace, since they describe nearly all physically-based image-formation processes. The visibility algorithm works by establishing a particular drawing order in which the last surface written at each pixel corresponds to the visible surface at that point. The drawing process can then proceed without intermediate tests or auxiliary storage. In this respect it is similar to the classic painter's visibility algorithm [Rogers85]. The unique aspect of this proposed technique is that the drawing order is established independent of any information relating to the scene's geometry, including the depth information. Thus, I claim that, for an important class of scene descriptions, there exists an algorithm for computing correct visibility solutions that requires no information about the geometric contents of the scene.

1.1 Background

Sutherland and Sproull [Sutherland74] first realized that the task of determining visibility can be reduced to a sorting problem. In their classic taxonomy of hidden-surface algorithms, they showed that all of the previously known visibility algorithms can be distinguished entirely by the permutations over which each of the three-dimensions are traversed while sorting. Moreover, if the scene primitives are placed into a common canonical frame (where x and y correspond to the coordinate system of the desired output image), then it can easily be shown that only one of the three sorts (the one in z) actually determines visibility, while the remaining two sorts merely provide algorithmic optimizations.

In essence, the visibility problem can be considered parallel sorting where separate lists