Toward a Fuzzy Hidden Surface Algorithm

Hong Lip Lim

ABSTRACT

This paper describes an algorithm for the efficient determination of obviously invisible surfaces in 3D hidden surface removal. The algorithm exploits the coherence between nearby viewpoints. It performs fuzzy analysis of the surface and patch projections. By carrying out imprecise calculations on a z-buffer and applying approximations which always err on the side of caution, it can efficiently detect surface patches totally invisible to these viewpoints. These patches may be culled, and need not undergo elaborate visibility computations for each viewpoint.

KEYWORDS

Hidden surface removal, visible surface detection, z-buffer, fuzzy projection.

1. INTRODUCTION

Visible surface detection is one of the most basic operations in 3D graphics. It is applied to generate images of surfaces directly visible to a viewer. Recently, it has also been used in the radiosity method to compute the energy interactions between surfaces (Cohen 1985.)

The standard strategy of visible surface detection is to divide surfaces into patch elements and compare the spatial relationship between these elements. Using this strategy, the visibility of surfaces cannot be determined until they have been analyzed in detail. This limitation can seriously affect the speed of visible surface computation if the scene is complicated.

This paper describes an algorithm for overcoming the above limitation. The algorithm applies the coherence between nearby viewpoints. By carrying out imprecise calculations, it can efficiently compute surfaces obviously invisible to these viewpoints. It therefore can substantially improve the speed of the visible surface and radiosity computations.

2. THE LIMITATIONS OF THE CURRENT HIDDEN SURFACE ALGORITHMS

The early visible surface techniques mainly applied various sorting schemes to find the occluding surface primitives (Sutherland 1974.) However, with the advancement in hardware technology, it is now common practice to reduce the need for sorting and comparisons by using the memory intensive BSP-tree (Fuch 1980, Gordon 1991) and z-buffer algorithm (Catmull 1974.)

The z-buffer algorithm is simple and has a very low growth rate. However, it still requires elaborate depth evaluations and comparisons during the scan-conversion of patches. On the other hand, an orderly traversal of the BSP-tree would be sufficient to establish the depth order. However, the BSP-tree needs to be re-organized whenever the scene changes.

There have been two main strategies for avoiding detailed depth analysis of totally invisible entities. One strategy applies the property that visibility changes can only occur at the contour edges of surfaces (Appel 1967, Hubschman 1981, Hornung 1984, Lim 1987, Teller 1991). Visibility computations of internal edges or patches can be reduced by first comparing them with these edges. The other strategy uses the invisible coherence at scan lines to avoid detailed rendering and depth evaluation of invisible surfaces (Crocker 1984, Gordon 1991.)

However, the contour-oriented techniques usually restrict the environments to those consisting exclusively of convex, axis-aligned or non-penetrating surfaces. They require the comparison of edges/patches with the contour edges regardless of their visibility. The invisibility coherence techniques still demand some sorting and maintenance of invisible edge and patch segments at each scan line.

3. AN OVERVIEW OF THE FUZZY HIDDEN SURFACE REMOVAL ALGORITHM

This paper describes an algorithm for improving the efficiency of visible surface computations. Unlike the earlier techniques, the algorithm is applied before the actual visible surface computations are carried out. Similar to the technique described by Teller et al (1991), it detects patches whose visibility can be easily determined. These patches may then be filtered out and need not undergo detailed visibility analysis.

The algorithm classifies viewpoints into groups. By simultaneously mapping all the view projections of each group onto a group projection cube, the occlusion relationship between surfaces is collectively considered for the group.

With respect to the whole group there would be vagueness of the viewing position and hence imprecision in the view projection of any surface. Therefore, the information content of the combined view projection is fuzzy. However, particularly for large surfaces, these regions may contain sub-regions that are under the view projections of all the current group of viewpoints. If the combined projection of a surface element is always behind this kind of sub-regions, it is always hidden from any of the viewpoints.

The suggested algorithm can be outlined by the pseudo-codes in the following page.

```
group viewpoints into groups;
for (each group of viewpoint) {
    for (each surface) {
        compute the umbra regions;
        for (each umbra region of the surface)
            scan convert the region to a z-buffer;
    }
    for (each surface) {
        for (each patch of the surface) {
            if (the patch is totally behind umbra regions stored in
                  the z-buffer elements under its fuzzy projection)
                   the patch is totally invisible;
        }
    }
    for (each viewpoint) {
        for (each patch) {
            if (patch is totally invisible)
                 bypass the patch
            else
                 subject the patch to normal hidden surface computations;
        }
    }
}
```

4. VIEW PROJECTION OF A SINGLE VIEWPOINT

A **viewpoint** $v \in \mathbf{R}^3$ is the spatial position and orientation of a viewer at an instance of time. The **view projection** of an entity s to a viewpoint is the mapping

$$m_p : s \rightarrow p \quad s \in \mathbf{R}^3, p \in \mathbf{P} \subseteq \mathbf{R}^2, v_i \in V \tag{1}$$

In 3D graphics, **P** is usually a plane called the view projection plane. However, to enable the correlation of viewpoints with different orientations, this paper applies the global cube concept in (Immel 1986) and defines **P** as the surface of a cube whose centre is at the viewpoint. Such a cube is called a **projection cube**.

The projection cube is usually axis-aligned with respect to the current group coordinate system. The sides of a projection cubes are called the **projection faces**.

Locations on a projection face can be expressed by a coordinate system called the **projection coordinate system**. This coordinate system is normalized so that all the projection cubes would have the same size if expressed in their respective coordinate systems.

5. MERGING THE VIEW PROJECTIONS OF A GROUP OF VIEWPOINTS

Several viewpoints may be grouped together through properties they share. The **current group** $V = \{ v_1, v_2, \ldots, v_n \}$ is the group of viewpoints that are currently being considered. Each group of viewpoints is associated with a coordinate system called the group coordinate system. The **viewpoint bounding box** of a group of viewpoints is the smallest right quadrangular prism enclosing these viewpoints and aligned with the coordinate system.

The view projections of a group of viewpoints can be correlated by a cube whose edges are aligned with the view projection cubes of the group. Because the view projections of various viewpoints are *simultaneously* mapped onto it, the combined mapping to the fuzzy projection cube would form a fuzzy image of that surface. Therefore, this cube will be called the **fuzzy projection cube**.

Each face of the cube is called a **fuzzy projection face**. Again, the coordinate system associated with this cube is normalized so that the latter has the same dimension as its associated projection cubes if measured in their respective coordinate systems.

Since all the cubes are parallel and normalized, all points on the view projection faces that have the same projection coordinates would mapped to the same point on the fuzzy projection cube **F**. This mapping is defined as the **fuzzy projection** f of a view projection p:

$$m_f : p \to f \quad p \in \mathbf{P} \subseteq \mathbf{R}^2, f \in \mathbf{F} \subseteq \mathbf{R}^2 \tag{2}$$

With this mapping, points on projection faces having the same projection coordinates and hence representing the same viewing direction would map to the same point on the associated fuzzy projection face. Therefore, like the projection faces, a point on the fuzzy projection face also represents a unique viewing direction. However, since the point is associated with several viewpoints, there is intrinsic vagueness in the viewing position it represents.

The relationship between the projection cubes and the fuzzy projection cube is shown below:

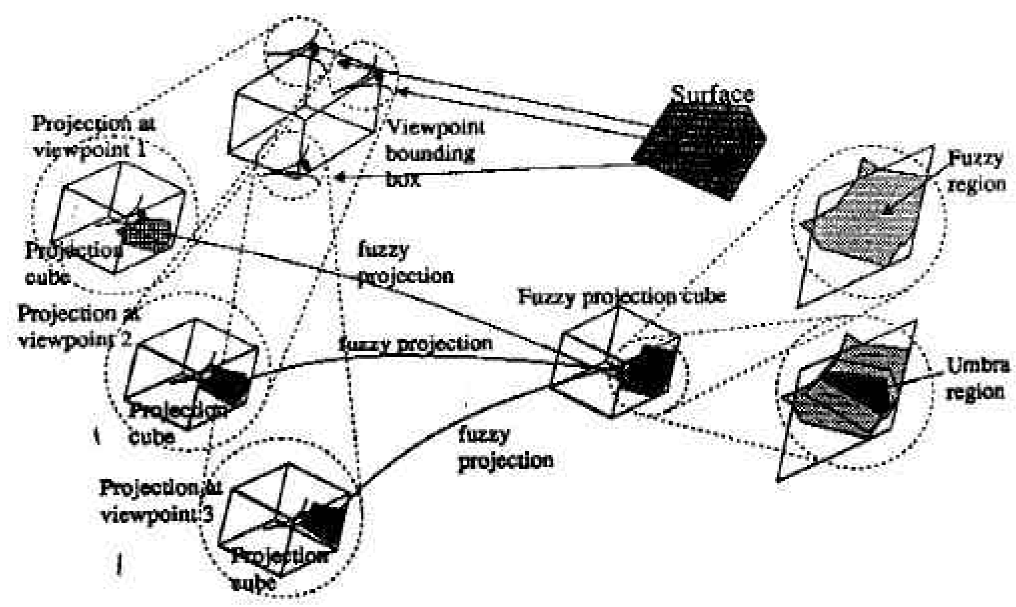


Figure 1: Merging the view projections on the fuzzy projection cube

6. THE CLASSIFICATION OF REGIONS ON THE FUZZY PROJECTION CUBE

Consider a surface $S \subset \mathbf{R}^3$ which is seen from a group of viewpoints $v_i \in V$. Applying (1), the view projection of S to viewpoint v_i is

$$P_i = \{ p \mid p = m_p(s),\ s \in S,\ p \in \mathbf{P},\ v = v_i \in V \} \tag{3}$$

From (2), the fuzzy projection of P_i onto the fuzzy projection cube is

$$F_i = \{ f \mid f = m_f(p),\ p \in \mathbf{P}_i,\ f \in \mathbf{F} \} \tag{4}$$

While the projection status on a point on a projection cube with respect to a surface S is simply one of the boolean values {0, 1}, the degree of projection at a point $x \in \mathbf{F}$ with respect to the group of viewpoints is more sophisticated. It may be measured by the characteristic function:

$$\mu(x) = \frac{\text{Number of times the fuzzy projections of } S \text{ map on } x}{\text{Total number of viewpoints}}$$

$$\mu(x) \in \{ 0,\ \frac{1}{n}, \frac{2}{n}, \dots \frac{(n-1)}{n},\ 1 \} \tag{6}$$

$\mu(x)$ represents the degree S is seen from the current group of viewpoints in the direction corresponds to x. Since it is not binary unless n is 1, the set of points on the fuzzy projection cube $\mathbf{F}$ with $\mu_A(x)$ is a fuzzy set (Zadeh 1965, Kaufmann 1975) in F:

$$A = \{ (x,\ \mu_A(x)) \mid x \in \mathbf{F} \} \tag{7}$$

Based on the value of $\mu_A(x)$, the following regions can be defined on the fuzzy projection cube for each surface:

A **fuzzy region** U is the *support* of A, i.e.

$$U = \sup A = \{x \in \mathbf{F} \mid \mu(x) > 0\} \tag{8}$$

U is the area on the fuzzy projection cube which is under the fuzzy projection of the surface from at least one of the viewpoints in the current group. Since $\mu(x) > 0$ if and only if $(x \in F_1) \vee (x \in F_2) \dots (x \in F_n)$,

$$U = \bigcup_{i=1}^{n} F_i \quad \forall\, v_i \in V \quad \Rightarrow \quad F_i \subseteq U \quad \forall\, v_i \in V \tag{9}$$

An **umbra region** C on the fuzzy projection cube is the α-*cut* of A with $\alpha = 1$, i.e.

$$C = \{x \in \mathbf{F} \mid \mu(x) \geq \alpha = 1\} \tag{10}$$

The umbra regions are so called because in all the viewing angles covered by them, all surfaces further than their surfaces would be hidden from the current group of viewpoints. Since $\mu(x) = 1$ if and only if $(x \in F_1) \wedge (x \in F_2) \dots (x \in F_n)$,

$$C = \bigcap_{i=1}^{n} F_i \quad \Rightarrow \quad C \subseteq F_i \quad \forall\, v_i \in V \tag{11}$$

The fuzzy and umbra regions are shown in figure 1.

7. FUZZY PROJECTIONS IN A PATCH-ORIENTED ENVIRONMENT

If surfaces in the environment are represented by meshes of patches, individual patches may also be projected onto the view and fuzzy projection cubes. The relationship between the fuzzy projections of patches and surfaces can be described by the following theorems:

Theorem 1: If the fuzzy region of a patch is totally inside the umbra region of a surface, the view projection of that patch is totally inside the view projection of the surface for every viewpoint in the current group.

Proof: Assume that P_i and F_i are the view and fuzzy projections of surface S from viewpoint $v_i \in V$, and p_i and f_i are the view and fuzzy projections of a patch p belonging to another surface and seen from v_i. If U_p, the fuzzy region of p, is totally inside C_s, the umbra region of S, then $U_p \subseteq C_S$. Applying (9) and (11),

$$\bigcup_i f_i \subseteq \bigcap_j F_j \quad \forall\ v_i, v_j \in V \tag{13}$$

$$\Rightarrow f_i \subseteq F_i \quad \forall\ v_i \in V \tag{14}$$

Since P_i and p_i are linearly mapped to f_i and F_i, (14) is true if and only if $p_i \subseteq P_i$, hence theorem 1 is the case.

Theorem 2: If the fuzzy region of a patch is totally inside the union of the umbra regions of several surfaces, the view projection of that patch is totally inside the union of the view projections of these surfaces for every viewpoint in the current group.

Proof: Assume that P_{ik}, F_{ik} are the view and fuzzy projection of surface S_k, $k \in \{1, ..., m\}$ from viewpoint v_i, $i \in \{1, ..., n\}$. Also assume that f_i, the fuzzy region of p, is totally inside the region containing the union of the umbra regions of all S_k. Applying theorem 1,

$$f_i \subseteq \bigcup_k \bigcap_j F_{jk} \subseteq \bigcup_k F_{ik} \quad \forall\ i,j \in \{1,...,n\},\ \forall\ k \in \{1, ..., m\} \tag{15}$$

$$\Rightarrow f_i \subseteq \bigcup_k F_{ik} \quad \forall\ i \in \{1,...,n\},\ \forall\ k \in \{1, ..., m\} \tag{16}$$

$$\text{Hence } p_i \subseteq \bigcup_k P_{ik} \tag{17}$$

8. APPROXIMATED FUZZY PROJECTION COMPUTATIONS

Last section indicates the relationship between a surface and a patch under vague viewing projections. However, unless the surface is very simple, its exact umbra and fuzzy regions can only be computed by merging the view projection computations from every viewpoint. The following sections describe how to avoid these computations by a series of approximations which always err on the side of caution.

8.1 Approximating the viewpoints by the viewpoint bounding box

To simplify the fuzzy projections, the viewpoints in a group are approximated by their bounding box. While this approximation would add extra imprecision to the computations, it substantially simplfies the fuzzy projection computations because the box is axis-aligned and has simple geometry. Another advantage is that the technique does not require the exact locations of viewpoints as long as the volumes containing them can be estimated.

8.2 Approximating the umbra regions of a surface by the umbra regions of its front-facing regions

To further simplify the projection computations, the umbra regions of a surface are approximated by the umbra regions of its front-facing regions. These regions are the set of patches always front-facing to the viewpoint bounding box and hence to the group of actual viewpoints.

Whether a patch is always front-facing can be determined by finding the smallest inclination angle of the patch with respect to the viewpoints. The cosine of the angle between the normal (n_x, n_y, n_z) of the patch and the view vector from a viewpoint (v_x, v_y, v_z) to a point on the patch (x_p, y_p, z_p) is

$$\cos\theta = n_x(p_x - v_x) + n_y(p_y - v_y) + n_z(p_z - v_z) \qquad (18)$$

The smallest θ corresponds to the largest possible $\cos\theta$, which can be computed by substituting the extreme components of the patch normal and the appropriate corner coordinates of the patch and viewpoint bounding box into (18).

Since the front-facing patches are facing all the possible viewpoints in the viewpoint bounding box, a front-facing region never curve back and is always facing these viewpoints. The view projections of its border therefore always enclose its projection. This is shown in figure 2.

8.3 The fuzzy extents of the boundary edges of front-facing regions

A boundary edge of a front-facing region is an edge belonging to a front-facing patch and not shared by another front-facing patch. Like patches and surfaces, it may be mapped onto the fuzzy projection cube and form a fuzzy region.

The exact fuzzy region of the boundary edge is difficult to compute. Since the view projections are linearly mapped to the fuzzy projection cube, the extent of the region can be more easily computed by finding the extrema of the view projection of the edge to the viewpoint bounding box, and mapping it to the fuzzy projection cube.

If d is the distance between the view projection face and the viewpoint, the view projection P (X, Y) of a point S (x', y', z') from a viewpoint V (x, y, z) is

$$X = (x' - x) * d / (z' - z) \qquad (19)$$
$$Y = (y' - y) * d / (z' - z) \qquad (20)$$

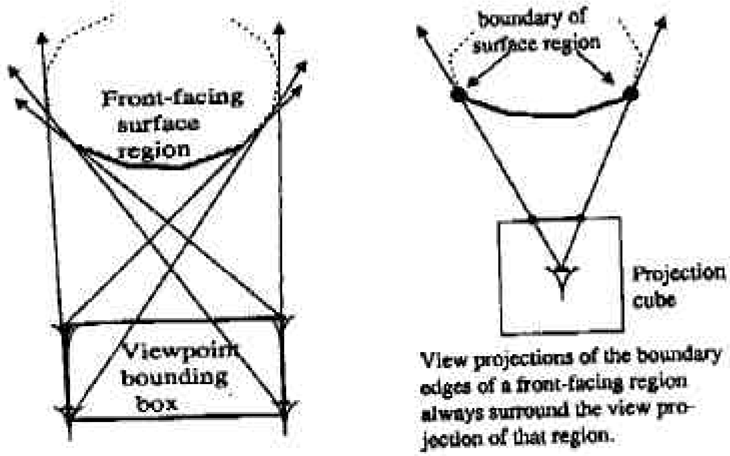


Figure 2: Cut-plane views of front-facing regions

By substituting the appropriate extreme coordinates of the axis-aligned edge and viewpoint bounding boxes into the above equations, the largest and smallest possible projection coordinates of an edge can be computed

In the hidden surface computations to be discussed, the fuzzy projections will be sampled by the pixels they cover. To ensure that each fuzzy extent surrounds its fuzzy region, the regions computed in equations (19) and (20) have to be further rounded to cover all the pixels they partially overlap. The resultant rectangle on each fuzzy projection face is called the **fuzzy extent** of the boundary edge.

8.4 Computation of the approximated umbra regions

The fuzzy regions of all boundary edges belonging to the front-facing regions of a surface S can be combined into an area B_s. Because of the nature of the front-facing regions, any change of the projection status must occur at these edges. Therefore, if a point is not in B_s, it must be either inside the fuzzy projection of these edges, or outside all these projections.

However, if such a point is already within the fuzzy projection of the region from one of the viewpoints, the second situation cannot be true. Therefore, for the front-facing region of a surface, the area inside its fuzzy projection but outside B_S is always within its fuzzy projection from any viewpoint in the viewpoint bounding box. This area must be a subset of the umbra region of the front-facing region. Therefore, if C''_s is such an area, C'_s the umbra region of the front-facing region, F'_i the fuzzy projection of this region from viewpoint $v_i \in V$, and $\bar{B}_s$ the complement of B_s, then

$$C''_s = F'_i \cap \bar{B}_s \subseteq C'_s \qquad (21)$$

Since the front-facing regions are a subset of their surface, if F_i is the fuzzy projection of the same surface, then $F'_i \subseteq F_i$. Hence

$$C'_s = \bigcap_{i=1}^{n} F'_i \subseteq \bigcap_{i=1}^{n} F_i = C_s \qquad (22)$$

Combining (21) and (22) and defining C_i'' as the **Approximated Umbra Region** of surface S, the following theorem is obtained:

Theorem 3: The approximated umbra region of a surface is all within the umbra region of the same surface.

The computation of the approximated umbra region is shown below:

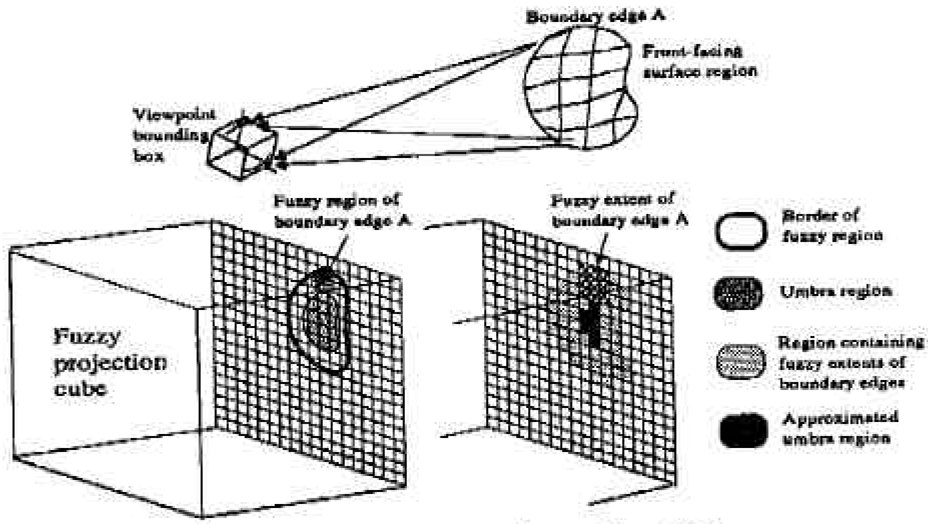


Figure 3 : The computation of the fuzzy region and the approximated internal region of a front-facing region

8.5 Inside tests between front-facing regions and a patch

Theorem 4: If the fuzzy extent of a patch is totally inside the union of the approximated umbra regions of several surfaces, the view projection of that patch is totally inside the regions combining the view projections of these surfaces for each of the viewpoints in the current group.

Proof: Assume that U_p is the fuzzy region of the patch and C_i and C_i'' are the umbra regions and the approximated umbra regions of surface S_i, $i \in \{1, ..., m\}$ then, as given,

$$U_p \subseteq \bigcup_{i=1}^{m} C_i'' \tag{23}$$

Since theorem 3 states that $C_i'' \subseteq C_i$,

$$U_p \subseteq \bigcup_{i=1}^{m} C_i \tag{24}$$

Hence the fuzzy region of the patch is totally inside the union of the umbra regions of several surfaces. Applying theorem 2, theorem 4 is derived.

9. THE COMPUTATION OF TOTALLY INVISIBLE PATCHES

To sample the projections on the fuzzy projection cube, each fuzzy projection face is tessellated into pixels. A z-buffer having as many depth fields as there are pixels on the fuzzy projection faces is allocated to store the mapping information on these faces. For each group of viewpoints, the following computations are carried out:

9.1 The z-buffer are first initialized to infinity to indicate that no surface has yet been projected on the projection cube.

9.2 The fuzzy extents of the boundary edges of every front-facing region are computed using the technique mentioned in section 8.3.

9.3 Based on section 8.4 and the following pseudo-codes, the difference between the front-facing regions and the fuzzy extents of the boundary edges of these regions is found and scan-converted. This produces the scan-line segments of the approximate umbra regions:

```
for (each surface) {
    for (each scan line) {
        compute the scan line segments corresponding to the view projection
        of surface from one of the current group of viewpoints;
        compute the scan line segments of the fuzzy extents of the boundary edges;
        subtract the view projection segments by the fuzzy extent segments;
    }
}
```

9.4 Each segment of each approximated umbra region is updated into the z-buffer. The largest possible z-distance between the associated front-facing region and the viewpoint bounding box is computed. This distance is written to the z-buffer elements belonging to the pixels under the segment if it is smaller than the value stored in these elements.

9.5 To compute patches totally invisible from the current group of viewpoints, the patches are accessed by turn. The z-buffer elements corresponding to the pixels under the fuzzy extent of the patch are accessed. Based on theorem 4, the view projections of the approximated umbra regions stored in these elements would surround the view projections of the patch from all viewpoints in the current group.

Therefore, if the distances stored in these elements are smaller than the smallest z-distance between the patch and the viewpoint bounding box, the patch would be hidden in all possible directions from the current group of viewpoints. This process is shown in figure 4.

10. IMPLEMENTATION

The suggested algorithm has been implemented and tested with several room models that differ in depth complexity and patch size. These models and a colour image of one of them are shown in figure 5. The times of hidden surface computations for five viewpoints in the viewpoint bounding box are shown in the following tables. The results show that the overhead of the suggested algorithm is low. It also indicates that the computation time glows very slowly with respect to the increase of depth complexity when the algorithm is used.

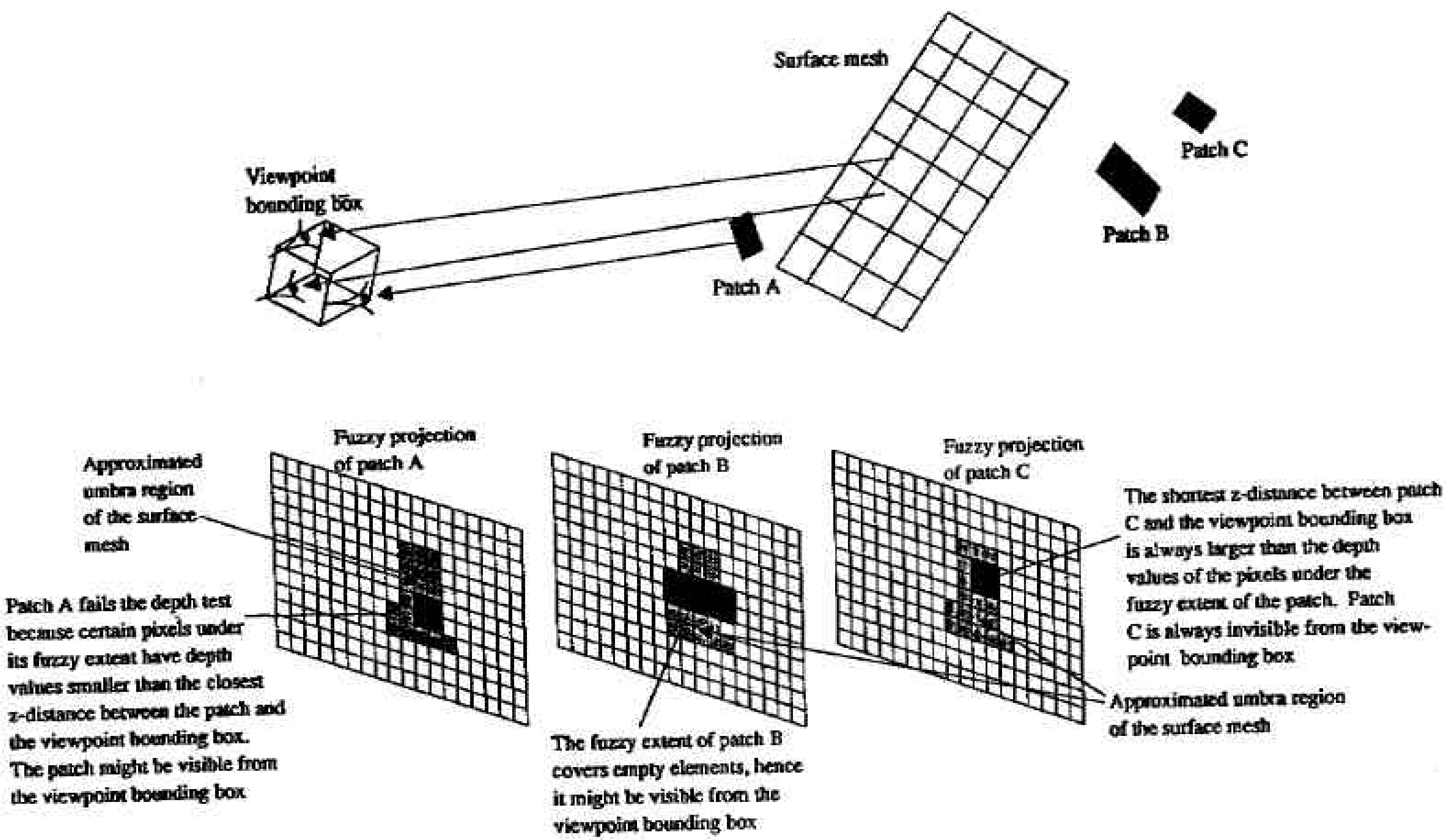
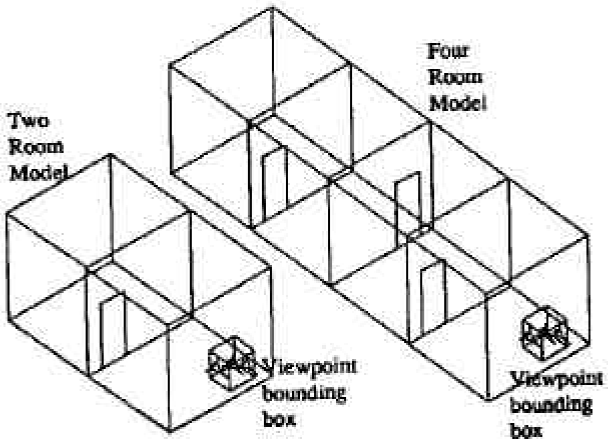


Figure 4: The detection of totally invisible patches on a fuzzy projection face



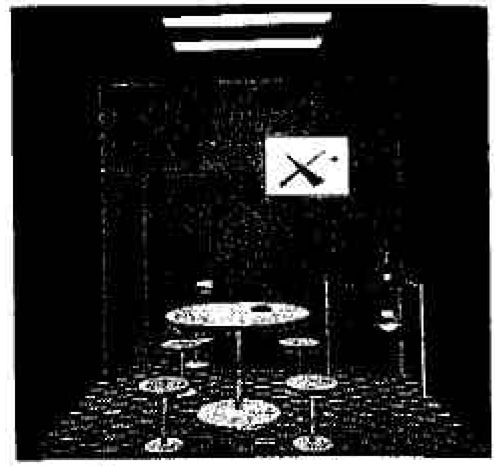


Image of the
2-room model

Figure 5: Display models with different depth complexities

1. Coarse patches, screen resolution: 400 X 400 pixels, resolution of fuzzy projection face : 400 X 400 pixels

	Normal Hidden Surface Removal		Fuzzy Hidden Surface Removal		
Model (number of rooms)	Total patches	CPU time for 5 viewpoints (sec)	Patches invisible to all viewpoints	CPU time for fuzzy computations (sec)	CPU time for 5 viewpoints (sec)
1	8619	30.8	16	0.8	30.9
2	17164	48.8	5505	1.3	34.7
3	25891	62.1	14113	2.1	36.4
4	34508	74.9	22730	2.6	38.6

2. Coarse patches, screen resolution: 800 X 800 pixels, resolution of fuzzy projection face : 400 X 400 pixels

	Normal Hidden Surface Removal		Fuzzy Hidden Surface Removal		
Model (number of rooms)	Total patches	CPU time for 5 viewpoints (sec)	Patches invisible to all viewpoints	CPU time for fuzzy computations (sec)	CPU time for 5 viewpoints (sec)
1	8619	58.1	16	0.8	58.2
2	17164	89.7	5505	1.4	67.8
3	25891	111.2	14113	2.1	68.8
4	34508	131.5	22730	2.6	72.7

3. Fine patches, screen resolution: 400 X 400 pixels, resolution of fuzzy projection face : 400 X 400 pixels

	Normal Hidden Surface Removal		Fuzzy Hidden Surface Removal		
Model (number of rooms)	Total patches	CPU time for 5 viewpoints (sec)	Patches invisible to all viewpoints	CPU time for fuzzy computations (sec)	CPU time for 5 viewpoints (sec)
1	20578	44.0	54	1.5	44.0
2	40801	69.0	12806	2.6	52.1
3	61618	91.7	33341	3.5	53.5
4	82194	110.1	53917	6.1	55.0

4. Fine patches, screen resolution: 800 X 800 pixels, resolution of fuzzy projection face : 400 X 400 pixels

	Normal Hidden Surface Removal		Fuzzy Hidden Surface Removal		
Model (number of rooms)	Total patches	CPU time for 5 viewpoints (sec)	Patches invisible to all viewpoints	CPU time for fuzzy computations (sec)	CPU time for 5 viewpoints (sec)
1	20578	77.1	54	1.4	77.1
2	40801	120.5	12806	2.9	90.0
3	61618	152.1	33341	3.9	92.0
4	82194	182.9	53917	6.1	94.0

11. DISCUSSION AND CONCLUSION

This paper has suggested a new approach to the computation of hidden surfaces. The algorithm applies the concept of fuzzy set theory to the analysis and classification of the fuzzy projections of surfaces. It also uses various strategies to handle uncertainty and enable imprecise computations.

The algorithm does not have the restrictions encountered by the earlier hidden surface algorithms. It does not preclude the presence of intersecting surfaces or patches in the environment. In addition, surface patches may be curved and the scene need not be static. Because it operates on a z-buffer, it is highly efficient and can be easily implemented in hardware.

The algorithm is similar to many techniques in pattern recognition and computer vision. For example, the use of multiple projections on the same plane has been applied in computer vision to compute optical flow (Subbarao 1988, Kanatani 1990) and to analyze dynamic scenes (Aggarwal 1981.) Many vision techniques also apply sophisticated techniques to analyze these mappings (Kanatani 1990, Siedlecki 1988.)

The algorithm also have many similarities with fuzzy computer vision and pattern recognition techniques. For example, its classification of regions is similar to Ruspini's fuzzy classification of images into regions based on the possibility values of these regions (Ruspini 1969.) The analysis of the surface occlusion relationship under imprecision also bears many resemblance to the fuzzy analysis of the scene in several computer vision and pattern recognition techniques (Jain 1983, Ligomenides 1986.) Similar to the suggested algorithm, these techniques also use approximations to reduce the complexity of the scene and enable its description.

This paper has demonstrated the advantage of exploiting the inherent imprecision in graphics computations. More advanced use of the fuzzy set theory could be possible in the radiosity method and motion blur techniques, as their results are intrinsically fuzzy.

ACKNOWLEDGEMENT

The author is grateful to Dr. Don Herbison-Evans, Mark James and the conference referees for their reviews of this paper. The author would also like to thank Basser Department of Computer Science, University of Sydney and Institute of System Science, National University of Singapore for their assistance and support.

REFERENCES

Aggarwal J. K. (1981), "Dynamic Scene Analysis," in "Image Sequence Processing and Dynamic Scene Analysis," Huang, T. S. (Eds,), Springer-Verlag, 40-73.

Appel A. (1967), "The Notion of Quantitative Invisibility and the Machine Rendering of Solids," Proc. of the ACM National Conference, Thompson Books, Washington DC, 387-393.

Catmull, E. (1975), "A Subdivision Algorithm for Computer Display of Curved Surfaces," PhD Thesis, Report UTEC-CSc-74-133, University of Utah.

Cohen, M. F., D.P. Greenberg (1985), "The Hemi-Cube: A Radiosity Solution for Complex Environment," Computer Graphics, 19(3), 31-40.

Crocker, G. A. (1984), "Invisibility Coherence for Faster Scan-Line Hidden Surface Algorithms," Computer Graphics, 18(3), 315-321.

Fuch, H., Kedem, Z., Naylor, B. F. (1980), "On Visible Surface Generation by A Priori Tree Structures," Computer Graphics, 14(3), 124-133.

Gordon, D.(1991), "Front-to-Back Display of BSP Trees," IEEE CG&A, September 1991, 79-85.

Hornung, C. (1984), "A Method for Solving the Visibility Problem," IEEE Computer Graphics and Applications, July 1984, 26-33.

Hubschman, H., Zucker, S.W. (1981), "Frame-to-Frame Coherence and the Hidden Surface Computation: Constraints for a Convex World," Computer Graphics, 15(3), 45-54.

Immel, D. S., Cohen, M. F., (1986) "A Radiosity Method for Non-Diffuse Environments,", Computer Graphics, Vol. 20(3), 133-142.

Jain, R., Haynes S. (1983), "Imprecision in Computer Vision," in "Advances in Fuzzy Sets, Possibility and Applications," Wang, P. (Ed), Plenum Press, 217-236.

Kanatani, K. (1990), "Group-Theoretical Methods in Image Understanding," Springer Verlag.

Kaufmann, A. (1975), "Theory of Fuzzy Subsets, Vol. 1, Fundamental Theoretical Elements", Academic Press.

Ligomenides, P. A. (1986), "Modeling Uncertainty in Human Perception," in "Uncertainty in Knowledge-Based Systems," Bouchon, B., Yager, R. (Eds), Springer Verlag, 337-346.

Lim, H. L. (1987), "Fast Hidden Surface Removal Through Structural Analysis and Representation of Objects and Their Contours," CGI'87, Japan, 75-88.

Ruspini, E.H. (1969), "A New Approach to Clustering," Information Control, Vol.15, 22-32.

Siedlecki, W., Siedlecka, K., Sklansky, J. (1988), "Mapping Techniques for Exploratory Pattern Analysis," in "Pattern Recognition and Artificial Intelligence," Gelsema, E. S., Kanal L. N., (Eds), Elsevier Science, 277-299.

Subbarao, M. (1988), "Interpretation of Visual Motion: A Computational Study," Pitman, London.

Sutherland, I. E., Sproull, R. F., Schumacker, R. A. (1974), "A Characterization of Ten Hidden Surface Algorithms," Computing Surveys, Vol 6., No. 1, 1-55.

Teller, S. J., Sequin, C. H., (1991), "Visibility Preprocessing for Interactive Walkthrough," Computer Graphics, 25(4), 61-69.

Zadeh, L. A. (1965), "Fuzzy Sets," Information and Control, Vol. 8, 338-353.

Hong Lip Lim has a BSc in Physics from National University of Singapore and an MSc in Computer Science from University of Essex, England. His research interests are computer graphics, virtual reality, fuzzy set theory and computer vision. He is currently with the Basser Department of Computer Science, University of Sydney, Australia.