ORGANIZATION OF A PHOTOGRAPHIC DATA BASE*

David J. Sykes
Dr. Roland di Franco**
Honeywell Training and Control Systems Division
West Covina, CA 91790

William J. Curran
Naval Training Systems Center
Orlando, FL 32813

ABSTRACT

The increasing emphasis on the use of photographic images in visual simulation has introduced new challenges. In photobased systems that allow full freedom of movement around three dimensional objects, the question of the image data base size and organization becomes a significant system design issue. This paper addresses the topic of storage and retrieval of multiple images that represent three-dimensional generic objects such as trees, buildings, and targets. In order to obtain a smoothly changing view of the object as the eyepoint moves relative to the object, many photographs taken at different aspect angles are used. Determining the actual number of different aspects required and how these should be physically stored is a nontrivial task. Increasing the number of different aspects facilitates the accomplishment of smooth transitions, but at the same time makes real-time retrieval more difficult. This paper describes a method of organizing the image data for a 3D object onto a disk taking the device performance characteristics into account.

INTRODUCTION

The complexity of a visual scene produced by real-time CGI techniques is limited by the number of polygons that can be processed in one frame period.[1] To overcome this inherent limitation, it is customary to use photomapping of texture onto certain polygons.[2,3] This creates a significant improvement in realism without the need for a vast increase in the polygon count. True CGI images are derived from "exact" transformations of 3D models and are therefore correct at any aspect angle. Photographs, on the other hand, are basically 2D in nature and are only strictly correct at the aspect angle and range at which the photograph was originally taken.

If the object is a planar surface such as a runway or a collection of planar surfaces such as a building, it is possible to digitally "warp" the photographic image of each surface. By means of a true perspective transformation, each surface is made to appear correct according to the simulated viewpoint.

In the case of 3D objects such as trees and vehicles, however, it is not so easy. Trees are complex structures with internal parallax and occulting among branches and leaves. A battle tank has many surfaces with protrusions many of which occult each other, either fully or partially. A warping process cannot introduce information that is not in the original image because of occulting, neither can it change parallax. Therefore, in the case of complex 3D objects, it is necessary to have a number of photographs taken at different aspect angles in order to capture the required information. To be strictly correct, it is also necessary to consider the effect of range on intra-object occulting. However, since the detail is only significant at close range, this effect can be ignored if the photographs are taken at close range.

The problem addressed in this paper is how to determine the number of photographs required, the angles at which they should be taken and their organization on the storage medium. Increasing the number of images not only increases the cost of creation and storage of the data base, but it also adds to the difficulty of real-time retrieval. On the other hand, reducing the number of stored images leads to noticeable jumps in the appearance of

* This work was performed under PM-Trade/NTSC Contract Number N61339-83-C-0066.

** Now at the University of the Pacific, Stockton, California

the object. At high angular rates as would occur in a fly-by or fly-over, the problem is not severe. However, at low rates as in the case of helicopter station keeping, masking/unmasking, target acquisition, etc., the situation is critical. In order to provide the subtle visual cues and avoid distracting jumps, it is vital that a smoothly changing view be provided regardless of the direction of motion and absolute position.

In this paper, a mathematical model for 3D object representation is first described followed by a discussion on various ways the photographs can be positioned around the object. Some experimental results are then reviewed which indicate the required angular separation of images. Next, the characteristics of appropriate storage media are summarized indicating a fundamental difficulty in meeting the requirements. A method of interpolation is then described which reduces the number of images while maintaining smooth transitions. Finally, a method of organizing the images onto a disk so as to minimize distance between adjacent views regardless of direction of motion is described.

THE MATHEMATIC MODEL FOR THE ORGANIZATION OF IMAGES

For each three-dimensional object in the simulation, there is a sequence of images of the object which must be obtained from the data base in order to construct the sequence of views which will be projected on the screen.

The object in question is assumed to be located at the center of an imaginary sphere called the photosphere. In this analysis, only the top hemisphere will be used. First a base meridian from the North Pole to the edge of the hemisphere is constructed. Then a finite number of angles of elevation are selected. Each of these angles of elevation defines a circle of latitude parallel to the equator as shown in Figure 1.

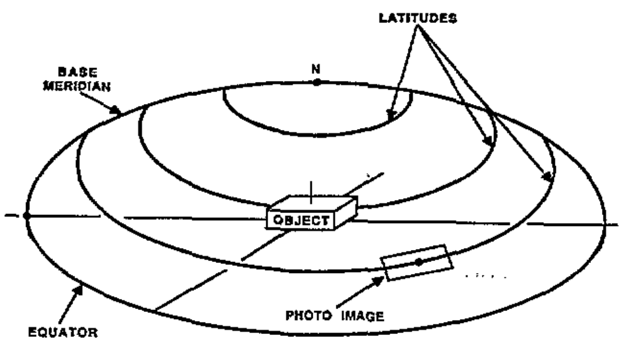

Figure 1. The Photosphere

Along each latitude, a finite number of positions is selected. A collection of photo images is constructed so that the center of each image is located at these select positions on the latitude lines. The images are taken to be tangent to the photosphere. There are several ways to select the angles of elevations and the subdivisions of each latitude.

The decision as to which image should be obtained next from the data base depends on the movement of the eyepoint relative to the object. Since, in general, there is no restriction of the direction of the motion of the vehicle, e.g., a helicopter, the next image needed may be to the left or right or above or below the previous image, I. The set of neighboring images of an image on the photosphere is the set of images which could possibly be needed for display in the next cycle of the system as shown in Figure 2.

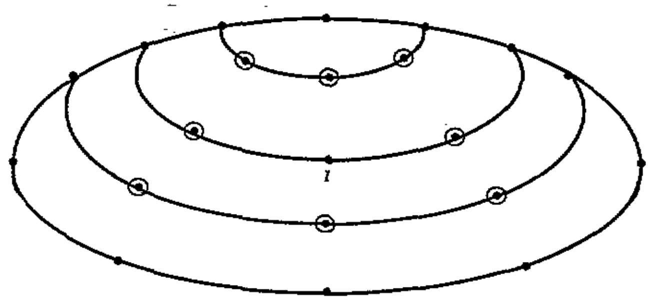

Figure 2. Near-Neighbors of an Image I

Image Patterns on the Photosphere

Three ways of organizing the images of the object on the two-dimensional surface of the photosphere were considered. These are:

1. The Equiangular Pattern
2. The Equidistant Pattern
3. The Equilateral Triangle Pattern

In all three cases, a system of spherical coordinates is used to define the positions on the photosphere. A position can be measured by its angle of elevation ϕ from the horizontal plane and its angle of azimuth θ from the base meridian as shown in Figure 3.

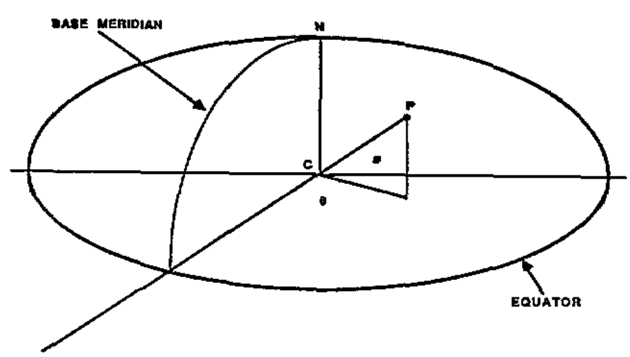

Figure 3. Spherical Coordinates

The equiangular pattern of positions is defined by selecting an angular increment. The range of ϕ from 0° to 90° is subdivided into angles which differ by the increment. The same is done for the range of θ from 0° to 360°. This will produce a grid of latitudes and longitudes on the hemisphere. Positions for the images will be at the intersection of each latitude and longitude as shown in Figure 4.

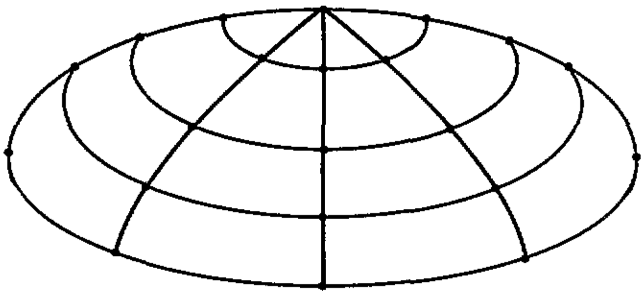

Figure 4. Equiangular Pattern

The equiangular pattern uses the same number of points on each latitude. Hence, the number of images is computed as the product of the number of latitude lines and the number of points on each latitude line. Table 1 shows the number of images as a function of angular increment.

Table 1. Number of Views in an Equiangular Pattern

Angular Increment	Number of Views in Equiangular Pattern
5.0°	1,296
2.5°	5,185
2.0°	8,101
1.0°	32,401
0.5°	129,601
0.25°	518,401

Since this pattern uses the same number of positions on each latitude line, there is a greater overlap of information in the images on a latitude near the North Pole than there is on a latitude near the equator. This redundancy of information makes the equiangular pattern undesirable.

The equiangular pattern can be improved upon by a method in which a fixed increment of arc length is used to space the center points of the images. The construction of this pattern, called the equidistant pattern, begins by subdividing the equator of the photosphere into a number of arcs of equal length. This can be accomplished by specifying an angular separation of points on the equator. This increment of arc is used as a measure to lay off segments of equal arc length along the base meridian. Circles of latitude are constructed at each point of subdivision on the base meridian. The increment of arc is then used to lay off arc segments of equal length along each latitude as in Figure 5. Since the length of each latitude is not necessarily an even multiple of the increment of arc used as measure, the last and first points on that latitude may be closer together than the other points on that latitude. This produces a pattern called the "equidistant" pattern. Careful analysis shows that most points are equidistant from one another along latitude lines, but only approximately equidistant from one another when on different latitude lines.

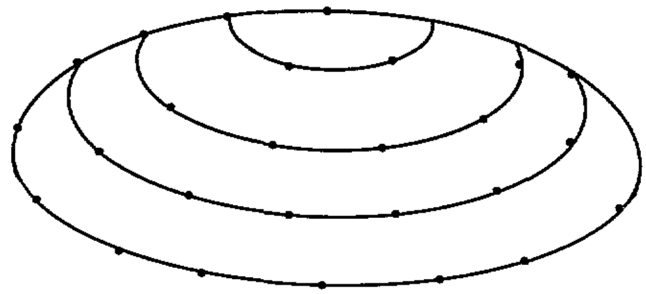

Figure 5. Equidistant Patterns

Observe that latitudes decrease in circumference when moving from the equator to the North Pole. Consequently, the number of views decreases with upward movement on the sphere. Table 2 shows how the number of views in the pattern changes as a function of the number of equidistant views at the equator.

Table 2. Number of Views in the Equidistant Pattern

Number of Views at the Equator	Separation at the Equator (degrees)	Number of Views on the Hemisphere
72	5.0	846
90	4.0	1,348
120	3.0	2,369
144	2.5	3,344
180	2.0	5,225
360	1.0	20,764
720	0.5	82,480
1,440	0.25	330,617

Note that the number of views is approximately 64 percent of the corresponding number of views using the equiangular distribution in Table 1. However, for angular separations of less than two degrees, the number of different images to be stored and accessed is incompatible with the size of storage devices which have acceptable retrieval time for photographic images.

The third image pattern, the equilateral triangle, was devised for the purpose of generating interpolated images. The topic of interpolation will be discussed later. The equilateral triangle pattern is constructed as follows.

First, divide the hemisphere into six regions called hexants as shown in Figure 6. Subdividing the base meridian into n equal arcs provides n points on the meridian through which n latitude lines are constructed.

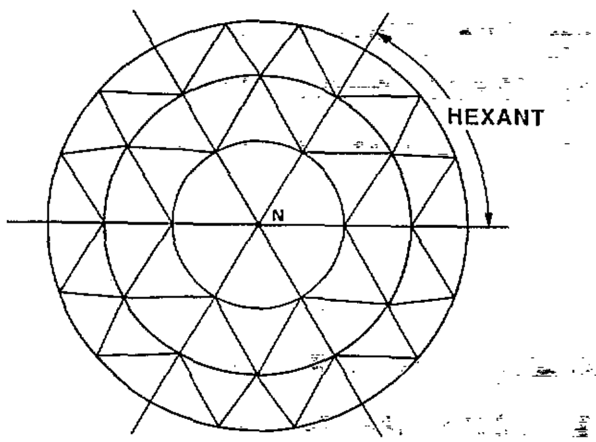

Figure 6. Top Down View of Equilateral Equilateral Triangle Pattern on the Hemisphere

Note that the first latitude as measured from the top makes a set of six curvilinear triangles with the North Pole as a common vertex. The segment of the kth latitude in each hexant is subdivided into k equal arcs. The k+1 points of subdivision on the kth latitude segment are connected with subdivision points on the (k-1)st latitude segment to create the equilateral triangle pattern as shown in Figure 7.

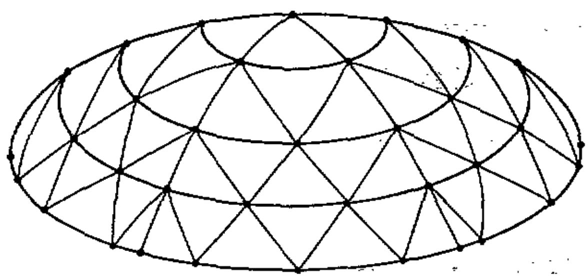

Figure 7. Equilateral Triangle Pattern on the Sphere

If there are n latitude lines, the equilateral triangle pattern will have 6k images on the kth latitude. The total number of views is given by:

$$V(n) = 3n^2 + 3n + 1$$

The total number of triangles is:

$$T(n) = 6n^2$$

A choice of 33 latitude lines leads to a total of 3,367 views. In the vertical direction, the angular increment for latitudes is $\Delta \phi = 2.72^\circ$. At the equator, the increment is $\Delta \theta = 1.82^\circ$ and there are 198 views on the equator. There are 6,534 triangles in the pattern.

STORAGE DEVICES

Two types of devices are considered for storage of the large number of images in the data base. These are the analog video disk and the Parallel Data Transfer Disk, PDTD (a high performance Winchester disk). Optical digital disks are not considered because of their low data transfer rate and long access time.

With the analog video disks, one image is stored per frame in a linear string. Accessing a near-neighbor image is accomplished by jumping over adjacent frames. Because of limitations of the "seek-time" of the video-disk and the cycle time of the total system, the maximum allowable jump is 200 frames per video period. This means that any near neighbor image must be within 200 frames of the current frame. With this limitation the large capacity of the video disk (54,000 frames) cannot be used for real-time retrieval. Nevertheless, if only a small percentage of the capacity can be used for a single 3D object, the low device cost makes the video disk attractive.

In the case of the PDTD, the capacity is much lower and the cost considerably higher than the video disk and the storage is digital as opposed to analog. The data for a single 512 x 512 pixel image requires approximately 53 kilobytes of storage if color encoding and data compression are used. On this basis, a 650 MB PDTD will hold about 12,000 images. Unlike the video disk, however, the PDTD can access most images stored in the device within one video frame period.

EXPERIMENTAL RESULTS

In order to determine the necessary angular increments for smooth transitions at low angular rates, the following experiment was performed. Using a video camera and a scale model tree, images were captured and digitized at 1°, $1/2^\circ$, and $1/4^\circ$ intervals over 40° in azimuth. Three video sequences were then generated to simulate the view of flying around a tree with 1°, $1/2^\circ$, and $1/4^\circ$ steps. The same angular rate of 3.75 degrees per second was maintained in all three cases by repeating each image for the appropriate number of frames.

Observation of the video (sequences with the tree full size on the screen) revealed that:

a. 1° increments were very noticeable
b. $1/2^\circ$ increments were noticeable
c. $1/4^\circ$ increments appeared smooth

The experiment was repeated using a model tank and similar results were obtained, although the side views were not as sensitive as the front views because of the gun. However, the use of $1/4^{o}$ angular increments requires such large numbers of images (see Tables 1 and 2) on a photosphere that it is clearly impractical for data base generation, storage, and retrieval. Some other solution is necessary which provides smooth transitions without an excessive number of images. One approach to the solution is the use of spatial interpolation between images. This is described next.

SPATIAL INTERPOLATION

The basic idea of spatial interpolation is to blend together several images in order to create a desired intermediate image. This necessitates the development of a pattern which allows for a smooth transition of images. Simple linear interpolation in a single dimension is accomplished by using a weighted average of intensity data from two images, I_0 and I_1 to create an intermediate image I_r, as follows:

$$I_r = rI_1 + (1 - r)I_0$$

Here r is the fraction of the distance from I_0 to I_1, $(0 < r < 1)$.

Although computationally simple, linear interpolation produces a noticeable artifact. When rotating around an object from image I_0 to I_r to I_1, the image changes from a sharp image at I_0 to a less sharp image at the interpolated image I_r to a sharp image again at I_1 as shown below:

I_0 ————	I_r ————	I_1
SHARP	LESS SHARP	SHARP

This artifact can be eliminated by using a second level interpolation scheme as follows. The original images I_0, I_1, ... I_n are used to create a new data base of interpolated images J_0, J_2, ... J_m. For example, J_0 and J_1 might be obtained from I_0, I_1 and I_2 linear interpolation as follows:

$$J_0 = (I_0 + I_1)/2$$

$$J_1 = (I_1 + I_2)/2$$

Intermediate images J_r are then created by linear interpolation using J_o and J_1 as follows:

$$J_r = r\, J_1 + (1-r)\, J_0$$

Using these second order interpolates, experiment has shown that rotating around an object using J_0, J_r, J_1 produces images which exhibit a limited range of frequencies which are far less distracting to the viewer yet contain sufficiently sharp images for visual discrimination.

In two dimensions, linear interpolation of images can be accomplished using three images which are located at the vertices of a triangle. See Figure 8.

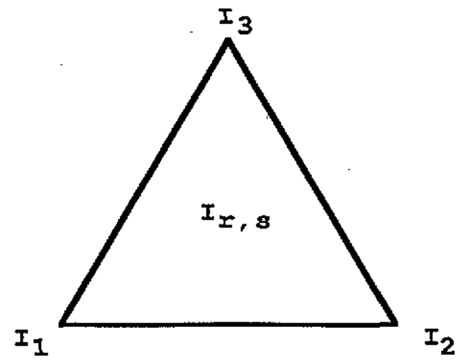

Figure 8. Triangular Pattern for Spatial Interpolation

The general interpolation formula for three images at the vertices of a triangle is given by

$$I_{r,s} = \sum_{i=1}^{3} a_i\,(r,s)\, I_i$$

Here the functions $a_i(r,s)$ are general weighting functions.

The utilization of a triangular pattern for interpolation necessitates a pattern of curved triangles for images on the photosphere. The equilateral triangle pattern described earlier was developed to meet the needs of this formula for interpolation. Every location on the photosphere lies within some triangle of the equilateral pattern. An image centered at a point on the photosphere can be obtained by interpolating from images at the corners of the curvilinear triangle.

As with one-dimensional interpolation, utilization of direct photographic images in the data base would create the same artifact described earlier. Hence the vertices in the equilateral triangle pattern represent positions of first order interpolated images.

To create this data base it is necessary to develop a pattern for photo images which when interpolated will generate the pattern of interpolated images. Figure 9 shows one way in which the two patterns can be interlaced for a single level. Each interpolated image lies in a curved triangle with photo images at the vertices. The pattern can be extended to as many levels as are needed.

Hence the equilateral triangle pattern provides a pattern of curved triangles which make interpolation of images on the photosphere feasible.

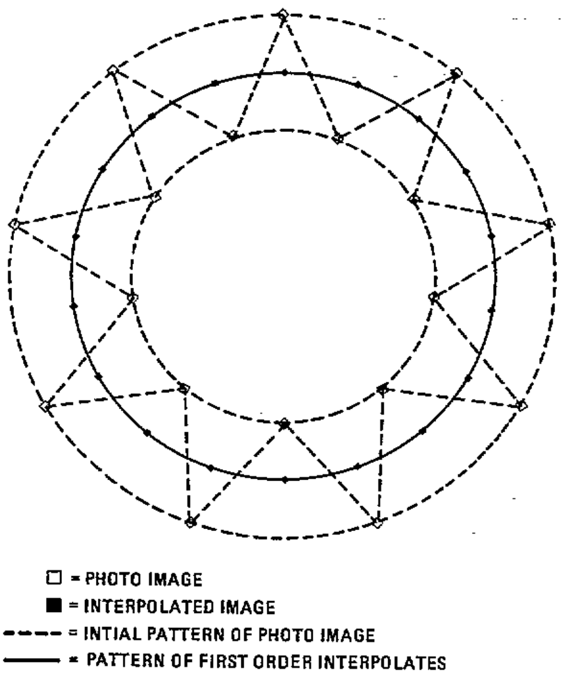

Figure 9. Interlaced Equilateral Patterns of Photo Images and First Order Interpolated Images on a Single Level

DATA BASE ORGANIZATION

The data base organization problem is to determine an efficient way to organize a 2-dimensional collection of images on a storage medium which is basically sequential or one-dimensional. Given a particular image on the photosphere that is being processed by the flight simulator, there are only a small number of possible images that may be needed next. These possible images are designated as near-neighbors of that image being processed. The near-neighbor images must be located in sequential memory close enough to the image in process so that they can be obtained within the cycle time of the simulator.

The organization of this data base is modeled by considering it to be a mathematical structure known as a graph. A graph consists of a collection of vertices and a collection of edges which connect certain pairs of the vertices. To build a graph-theoretical model of the data base, each image on the photosphere is defined to be a vertex of the graph. Two images are connected by an edge if they are near-neighbors of each other; i.e., the images could possibly be used on consecutive frames in the visual system. The resulting graph is called the near-neighbor graph of an object. Figure 10 shows an example of a near-neighbor graph on a photosphere.

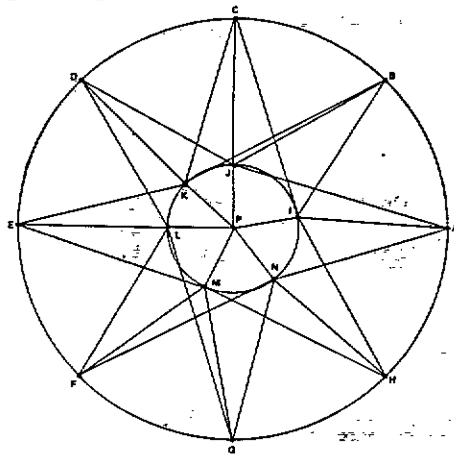

Figure 10. Example of a Near Neighbor Graph

Bandwidth of an Enumeration for a Graph

The arrangement of images in the sequential memory is accomplished by choosing an enumeration of the images or vertices of a graph. Denote by E(I) the positive integer associated with the image I. The jump in the enumeration between images I and I' is $|E(I) - E(I')|$. The number $m_E(I)$ will denote the maximum jump $|E(I) - E(I')|$ between the image I and any near-neighbor image I' to I. The bandwidth of the enumeration E for the graph is called B_E and denotes the maximum values of $m_E(I)$ over all images of the graph.

Figure 11 shows an enumeration of the previous near neighbor graph. In this case the vertices are enumerated in a clockwise order. The bandwidth B_E is 13 between vertices 1 and 14.

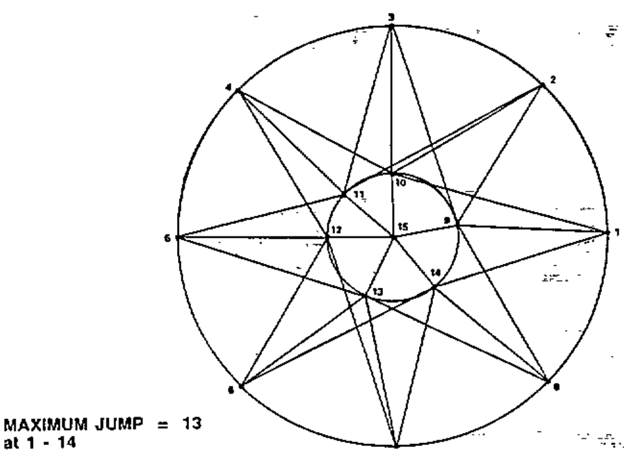

Figure 11. Enumeration of a Near-Neighbor Graph

Bandwidth Reduction

The images of an object are arranged in memory in the order determined by the enumeration E. It must be possible to move from a given image I to the next needed image I', a near-neighbor image, and transfer the image data for I' within one cycle of the visual system. Hence, it is advantageous to choose an enumeration with the lowest possible bandwidth. The bandwidth here represents the greatest jump that could occur between an image and a near neighbor image throughout the whole collection of images.

While the determination of an enumeration which produces the minimum bandwidth is still an unsolved problem, there exist algorithms for determining enumerations with reduced bandwidth. The Gibbs-Poole-Stockmeyer Algorithm given in Reference 4 was implemented to reduce bandwidths for enumerations on near-neighbor graphs. Application of this algorithm to the enumeration of the graph in Figure 11 resulted in the enumeration shown in Figure 12. The bandwidth has been reduced from 13 to 9 between vertices 4 and 13.

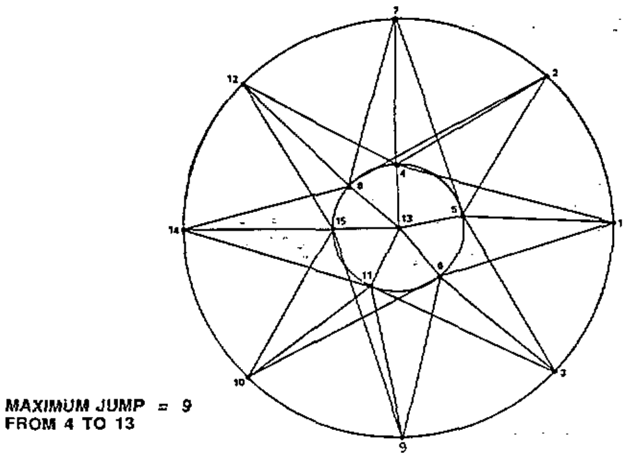

Figure 12. New Enumeration for the Near-Neighbor Graph

The bandwidth reduction algorithm was applied to a number of equidistant patterns of images on the photosphere. The results are shown in Table 3.

Table 3. Results of the Bandwidth Reduction Algorithm on Equidistant Patterns

Number of Views at the Equator	Number of Views on the Hemisphere	Bandwidth
37	231	38
46	353	45
59	590	65
72	846	57
90	1,348	101
120	2,369	128
144	3,344	130

The last entry in Table 3 represents a situation where image spacing at the equator is 2.5 degrees and the bandwidth is 130. This is well within the jump capability of a video disk.

Interpolation and Bandwidth Reduction on the Equilateral Triangle Pattern

The choice of an image pattern on the photosphere must have a limited number of images which does not exceed capacity of the storage device. In addition, the enumeration of the associated near-neighbor graph of images must have a bandwidth which is small enough to be within the seek-time of the storage disk. The problem is to meet both these criteria simultaneously.

Interpolation of images at the vertices of a triangle provides a method for reducing the number of images to be stored. The equilateral triangle pattern provides an arrangement of images which is set up for the interpolation scheme. An equilateral triangle pattern with 33 latitude circles requires a total of 3,367 images which is well within the capacity of storage media considered. With an angular increment for latitudes of $\Delta \phi = 2.72$ degrees and an angular increment of $\Delta \theta = 1.82$ degrees at the equator, views are close enough to provide interpolated images with sufficient visual discrimination as determined by experiment.

SUMMARY

Photographic representations of 3D objects are very effective in providing visual cues to helicopter pilots when training for combat missions. To avoid distracting jumps at low speeds, an angular resolution of 1/4 degree is required. Use of individual photographs at this resolution is out of the question because of the resulting data base size and retrieval time. Spatial interpolation provides one solution to the problem of smooth transitions with a moderately sized data base. Images can be organized on either video disks or PDTD's to decrease the separation of near-neighbor images. The combination of interpolation and bandwidth reduction provides a practical solution to the challenge of 3D object photographic data bases. The photographs can be actual photographs of a physical object or can be complex CGI images produced in nonreal-time. Either way, this work represents a major breakthrough in visual systems technology.

ACKNOWLEDGMENTS

The authors would like to acknowledge the contributions made by Dr. Kenneth Lane to the subject matter in this paper. Thanks are due to Dr. Kris Mudunuri for his assistance in generation of this manuscript and to Mr. Miles Christensen who was responsible for the diagrams.

REFERENCES

1. Economy, R. and Bunker, M., "Advanced Video Object Simulation," IEEE 0547-3578, 1984, pp. 1065-1071.

2. Robins, J. and Zimmermman, S., "Exploiting Texture in an Integrated Training Environment," Proceedings of the 7th Interservice/Industry Equipment Conference, November 1985, pp. 113-121.

3. Yan, J. K., "Advances in Computer-Generated Imagery for Flight Simulation," IEEE CG&A, August 1985, pp. 37-51.

4. Gibbs, N., Poole, W., and Stockmeyer, P., "An Algorithm for Reducing the Bandwidth and Profile of a Sparse Matrix," SIAM Journal of Numerical Analysis, Vol. 13, pp. 236-250, 1976.

ABOUT THE AUTHORS

MR. DAVID J. SYKES is a Project Staff Engineer at Honeywell's Training and Control Systems Division. During the past 18 months he has been responsible for the Visual System Component Development Program (VSCDP) contract awarded by PM-Trade. The first phase of this program consisting of technical studies was recently completed and now a prototype system is being implemented. In addition to visual systems, Mr. Sykes is also working on Artificial Intelligence applied to training systems. He holds an MS in Engineering from Arizona State University.

DR. ROLAND DI FRANCO received his Ph.D. in Mathematics from Indiana University in 1965. He has been a National Science Foundation Fellow at the University of California at Berkeley. During the period of the research done for this paper he was a staff member at Honeywell Inc. He has taught applied mathematics at Harvey Mudd College and is currently a professor of mathematics at the University of the Pacific. His current area of interest is image processing and computer graphics.

MR. WILLIAM J. CURRAN is an Electronics Engineer and Principal Investigator with the Naval Training Systems Center, Orlando, Florida. He is currently responsible for the Advanced Simulation Concepts Division's research program in both Advanced Sensor Simulation and Multi-Spectral Sensor Simulation. He is the Department of the Navy's technical point of contact for Inverse Synthetic Aperture Radar (ISAR) simulation for training applications. He has served as a consultant in Anti-Submarine Warfare Systems technology for the past eight years. He holds a Bachelor of Science Degree in Electrical Engineering from the University of Miami and a Master of Science Degree in Electrical Engineering from the University of Florida.