CIG TRANSLUCENT FACE SIMULATION PROVIDES MULTIPLE BENEFITS

DR. W. MARVIN BUNKER
General Electric Company

ABSTRACT

Military training trends are placing increasing emphasis on use of simulators for full-crew, full-mission training. Visual scene simulation must provide effective visual cues with a high degree of realism and a minimum of distracting effects.

The relationships between objects and their shadows, and the changes in these relationships as the observer moves in the gaming area, have been demonstrated to be extremely effective in contributing to the mental correlation process by which an observer extracts knowledge of the world from visual observations. Computer Image Generation (CIG) applied to visual scene simulation has always had the capability to validly portray shadows modeled as part of a fixed environment -- but has had difficulty coping with changing illumination or shadows of moving objects.

The optical laws which apply to transmission of light through translucent faces are quite simple, and it was readily demonstrated that such faces could be used to provide excellent simulation of shadows. This provided the incentive to devise algorithms for such simulation which would be feasible for implementation in real-time hardware.

After the capability to simulate translucent faces was developed, ideas arose for their use in applications other than shadows. They were used for windows, with very realistic results. Overlapping translucent spheres and ellipsoids were used for smoke and cloud simulation. For this application to be satisfactory, the processing must be modified from that process conforming to the laws of physics.

An extremely fruitful use for translucent faces is in implementation of gradual transition between versions of three-dimensional models. This application is expected to be of even greater significance than the use for which they were developed. As in the case of cloud simulation, processing must depart from physical laws for graudal transition -- but in a different manner than for clouds.

In summary, techniques have been developed for simulation of translucent faces. Three modes of operation apply different rules when more than one translucent face is imaged on the same portion of a view window. This allows extended use of the techniques in providing improved CIG effects.

SHADOW CUES

As motivation for the discussion on techniques for simulating shadows, we will first illustrate their effectiveness in providing cues which contribute to proper interpretation of scenes. Figure 1 shows a block sitting on a featureless surface. Figure 2 is the same block, several feet above the ground. Only by knowing the numbers used in the computation could this knowledge be obtained. It cannot be derived from the visual information in the scenes. Figures 3 and 4 show the same scenes, but with the addition of valid shadows. Now the relationship between the block and the ground is immediately obvious.

CIG systems can simulate shadows such as those shown by defining on the ground surface the face or faces representing the shadow. Standard processing then produces the correct portrayal. Where the object or the illumination vector is moving, it is not difficult to dynamically redefine the vertices of the "shadow faces" for valid displays including the movement.

Figure 5 shows a building setting on a pad, with the shadow extending beyond the pad to the background ground surface. To produce this effect with standard CIG systems, we must define faces representing the non-shadowed pad, faces representing the portion of the pad in shadow, and faces defining the shadowed region of the background surface -- each set with the proper computed color or tone. For a stationary data base, this computation is part of the off-line, nonrecurring data base preparation task. With large numbers of objects and shadow-faces in a dynamic simulation, the computational load rapidly becomes intractable. A partial solution has been to make the shadows black, completely obscuring all portions of faces which they cover. This reduces the real-time computational load to that required for situations such as illustrated by Figures 3 and 4 -- but at the expense of a significant reduction in realism. The promise of translucent faces applied to shadows is that a single face could be defined for the

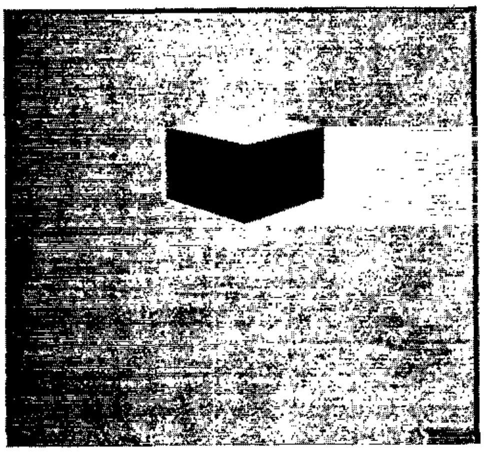

Figure 1. Block on Ground

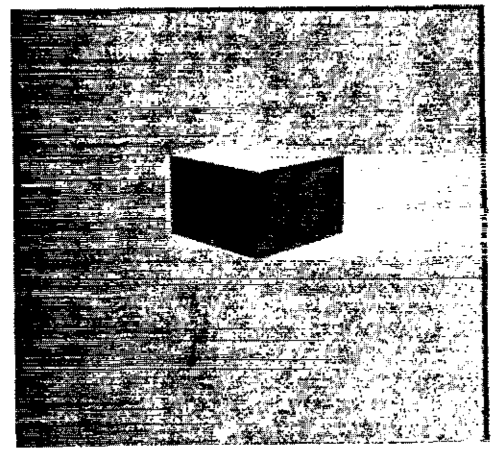

Figure 2. Block Above Ground

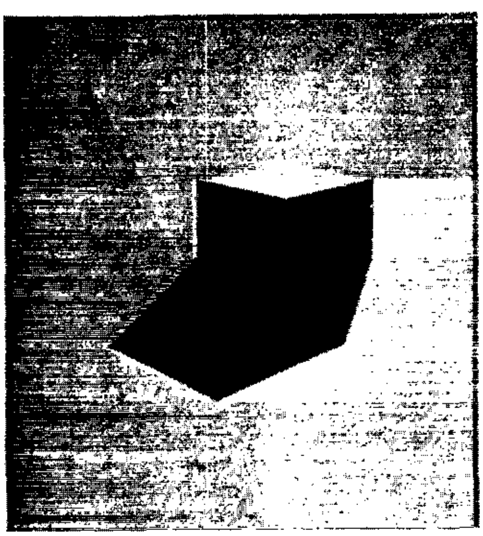

Figure 3. Block on Ground, with Shadow

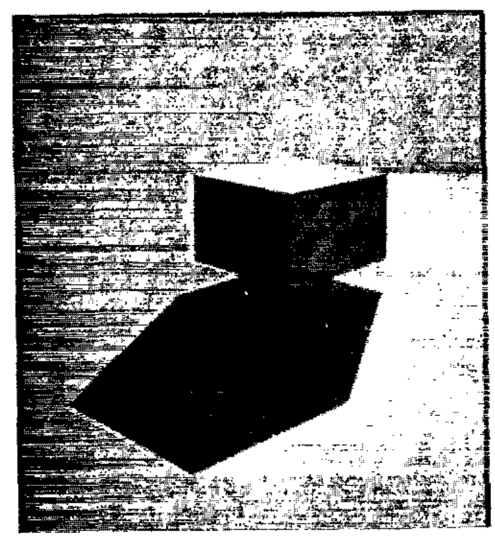

Figure 4. Block Above Ground, with Shadow

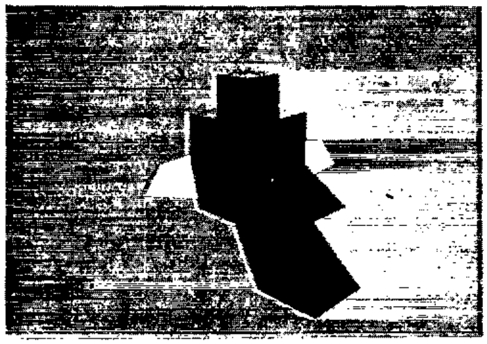

Figure 5. Shadow on Nonuniform Surface

shadow, it could be given color "black," with some defined percent saturation, and all faces and edges covered by the shadow would show through in their correct positions, with realistically modified colors and intensities. The shadow-face computation would be done without regard to what the shadow falls on; yet the result would be fully realistic. Figure 6 is a striking illustration of this realism. It also shows translucent faces used to simulate smoke of variable density.

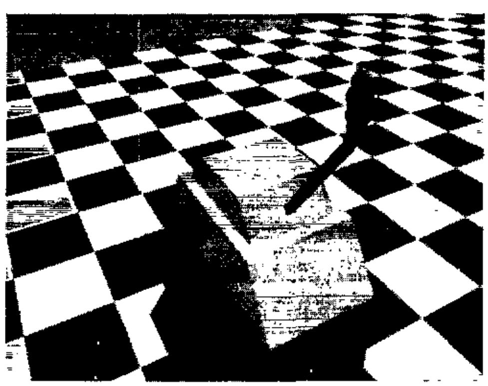

Figure 6. Tank with Shadow and Smoke

TRANSLUCENCE SIMULATION

If we consider a given pixel containing the image of a translucent face with another face (or portions of several other faces) behind it, the processing to determine the proper color for the video applicable to that pixel is quite straightforward. A number of extremely impressive scenes have been produced by applying non-real-time per-pixel processing to simulate translucence. In a typical real-time system, per-pixel time for each channel is 25 nanoseconds, and a number of channels must be processed simultaneously. Thus, for a concept to be feasible for real-time implementation, it must minimize any impact on the per-pixel processing. Ideally it should not impact it at all. The algorithms to be described meet this goal.

Edge Processing

Figure 7 is a partial block diagram showing processing of edges in a CIG system.

The Edge Generator contains information defining edges, circular features, and other entities in their entirety as they appear in the display windows for the current scene. For each scan line, the edge generator determines which edges are "active"; i.e. which appear on the current scan line. It truncates these edges to the top and bottom of the scan line, and outputs them formatted for the current scan line.

The Orderer accepts the set of edge definitions, and orders them in left-to-right order as required for the ultimate scan line video to be generated.

When two or more faces have their images on the same portion of the scan line, the decision as to which is to be shown is made by the next stage, the Priority Resolver. This portion also handles the highly complex and important function of implementing the area-times-color rule to reduce quantization effects.

The Priority Resolver output is routed to the Video Processors -- one for each display channel. These Video Processors combine information from several sources; they implement fog simulation, curvature, and other effects, and output the video to the display devices.

The Priority Resolver and Video Processor functions combined account for the major portion of the hardware in a typical CIG system. Thus, if a new feature can be added in a manner which does not impact these functions, the probability of success will be significantly increased.

Translucence Algorithms

The technique developed for translucent face simulation has the fundamental desirable characteristic defined previously -- it has no effect on any functions of the Priority Resolver or the Video Processor. The added functions take place between the output of the Orderer and the input to the Priority

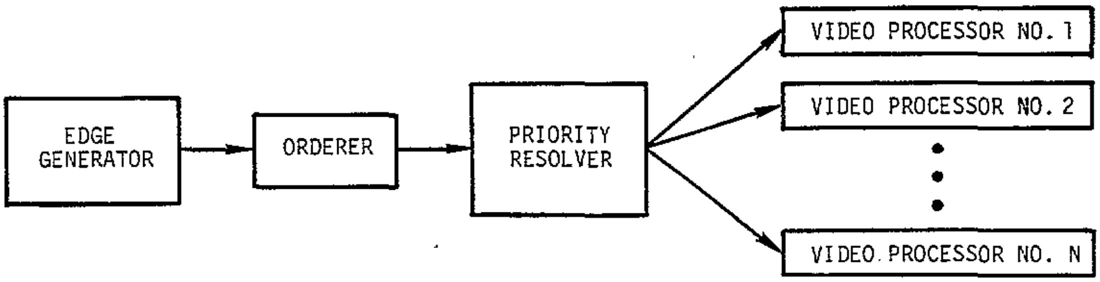

Figure 7. CIG Partial Block Diagram

Resolver. The set of edges from the Orderer is intercepted, modified, and the modified set of edges is then supplied to the Priority Resolver, where standard processing is applied. The result is valid translucent face simulation.

To achieve this goal, two different types of edge modification are required.

Type 1 Edge Modification

In the following, "T" designates tone, or intensity. "0" represents black, "255" is brightest white. "J" has its normal meaning of pixel number along a scan line.

Figure 8 shows part of a scan line. Edge 6 starts face 17, a solid face with a tone of 100 and dT/dJ = 0.5. Edge 7 starts face 4, a translucent face with tone = 20 and dT/dJ = 0. A type 1 modification consists of changing the tone and dT/dJ associated with an edge starting a translucent face. Assume face 4 has saturation of 40 percent. Then the tone associated with edge 7 is changed to (0.4 x 20 + 0.6 x 100) = 68, and dT/dJ is changed to 0.3. In later processing, when the priority resolver encounters edge 7, it will have the values for face 17 seen through face 4. In general, there can be more than one face behind the start edge of a translucent face. The algorithm must use the highest priority face of those behind the edge.

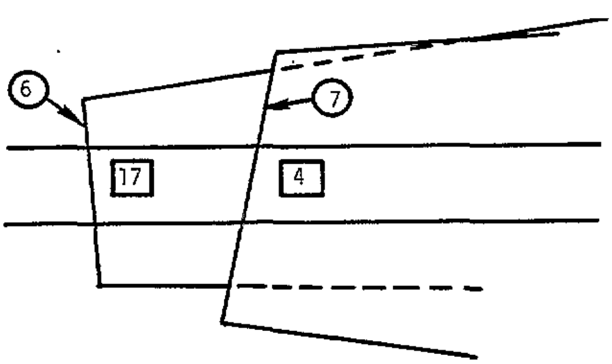

Figure 8. Configuration Requiring Type 1 Edge Modification

Type 2 Edge Modification

In Figure 9, edge 8 starts face 15, which has a tone of 200. At this point, the tone on the scan line must change to 128. However, we cannot just change the tonal information associated with edge 8. Note that at edge 9, we "fall off" the translucent face, and edge 8, with its unchanged original information, will be needed by the Priority Resolver to determine the fallback information. One approach would be to modify edge 9 in this pre-priority-resolver edge modification function, so it would contain the fallback data. Such thinking could make this edge modification function as complex as the Priority Resolver itself -- this we want to avoid.

The type 2 modification, applied to an edge which has a translucent face in front of it, involves creation of an additional edge for the scan line, spatially identical to the edge initiating the modification, with tonal information reflecting the combination of the translucent face and the face to the right of this edge, and with face-left number and face-right number both that of the translucent face.

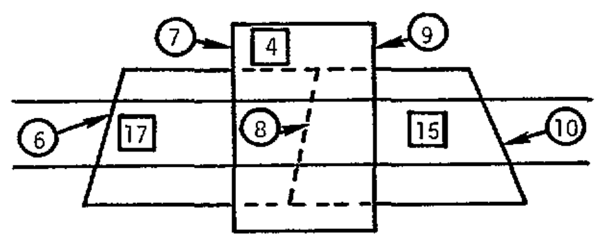

Figure 9. Configuration Requiring Type 2 Edge Modification

Results

Figure 10 shows an algorithm verification test scene. The two vertical

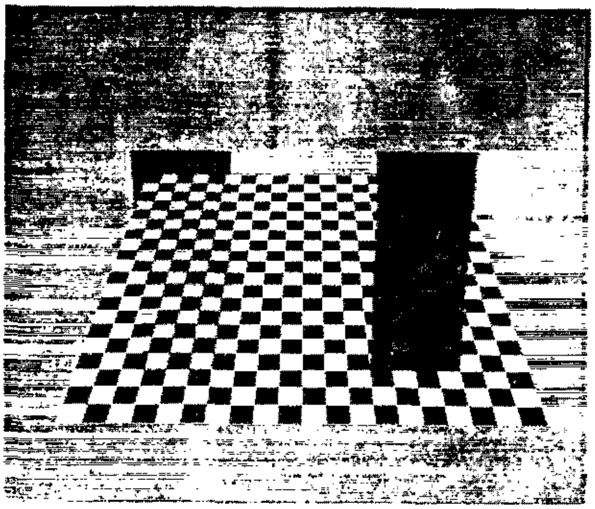

Figure 10. Translucent Face Simulation

faces are assigned the same tone, but different degrees of transparency of saturation. Every combination of edges was validly handled by the algorithm based on the edge-modification concept described.

Figure 11 is a scene more typical of those that might be expected in simulation. Translucent faces are used both for the shadow of the tower, and for the windows.

CLOUDS AND SMOKE

In the simulation of clouds and smoke, it is frequently desired that they be translucent rather than opaque, as in previous CIG systems. Highly realistic results have been produced in the past by using overlapping or intersecting spheres and ellipsoids for cloud and smoke simulation. It seems quite straightforward to replace the opaque sphereoids with translucent ones to achieve the desired result.

Overlapping Translucent Faces

Assume we have actual overlapping translucent faces shown as A and B in Figure 12. Assume face A transmits 40 percent of the incident light, and that face B transmits 60 percent. Then the region designated A+B will transmit 24 percent. If there is a bright surface behind the two, the A+B segment will appear as a lens-shaped dark region. This is not incorrect -- it is precisely the appearance we would get with two physical translucent discs. However if we wish to use overlapping spheres to simulate smoke or clouds, we do not want the effect described.

Modification for Cloud Simulation

A simple modification in handling of more than one translucent face on the same portion of a scan line results in the desired effect. For the portion of the scan line where both have images, the processing is as though all translucent faces except the one with highest priority have 100 percent transmission. In other cases, such as overlapping windows, it is necessary to simulate true physical translucence. Hence, a type-designation bit is interpreted by the edge-modification logic to determine the type of processing required. The smoke in Figure 6 illustrates the effectiveness of this processing.

GRADUAL LEVEL-OF-DETAIL TRANSITION

Level of Detail

In almost all existing CIG systems, a given feature may be modeled to several levels of detail, with the less-detailed versions requiring fewer edges. When an object is at such a distance that it is very small in the view window, smaller detail could not be seen even if it were computed and displayed, so a lower detail version is used to improve edge utilization efficiency. As the viewer approaches the object, higher detail versions are substituted for computation.

There has been one major disadvantage in the use of this technique. The visual perception system is extremely sensitive to abrupt changes in scene detail, even though they are small. Level of detail transitions are thus detected, and can be quite distracting. If the transition from one version to another is continuous and gradual as the exercise proceeds, and the increase in observable detail occurs in a very natural manner, just at it occurs when actually approaching an object, this distraction is eliminated.

Gradual Transition for Two-Dimensional Features

Such gradual transition is quite easy to implement in cases where each face involved has a background of unchanging known characteristics. Consider markings on a runway, for example. Assume marking color is white, and runway color is gray. During the period when a given runway stripe is undergoing transition, the entire face representing the stripe will be a constant color for each scene. This is computed as a proportional mix of the face color and the

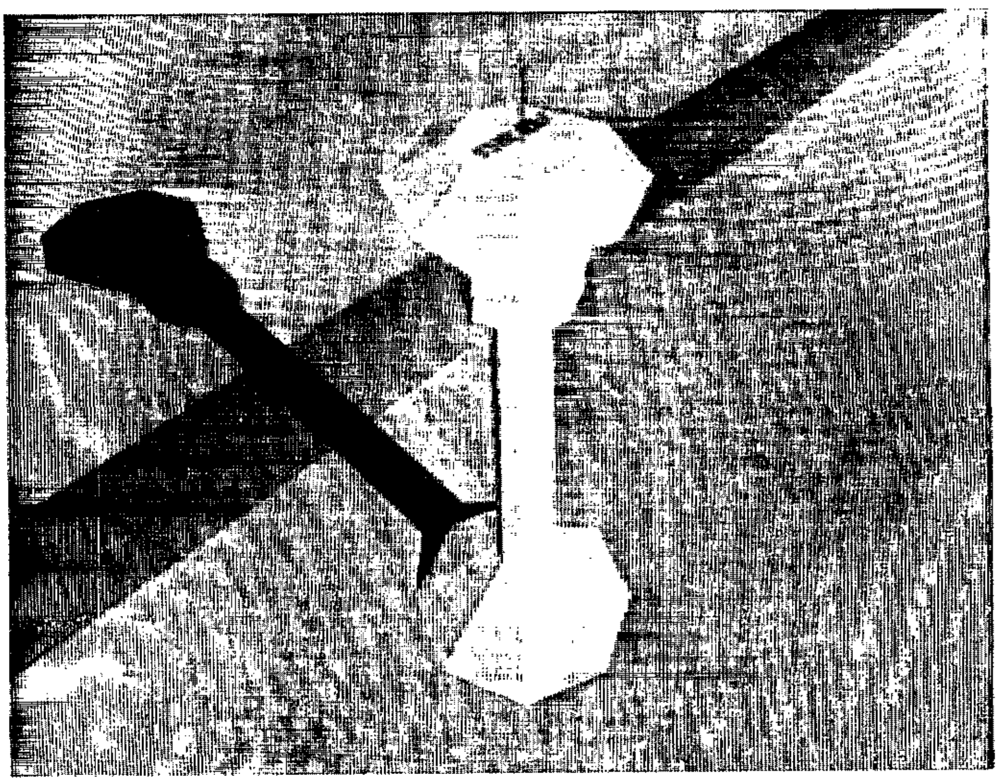

Figure 11. Tower with Shadow and Translucent Windows

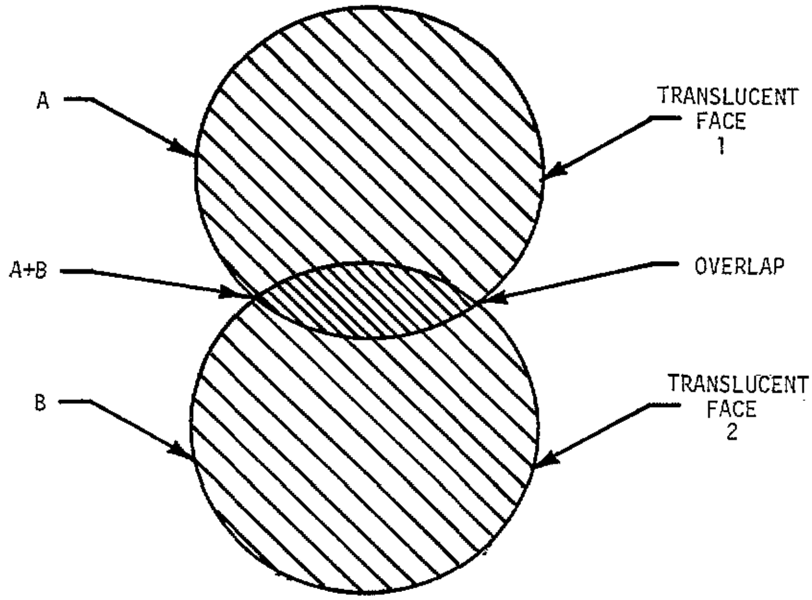

Figure 12. Effect of Translucent Face Overlap

background color, based on position in the transition region. This will change from scene to scene, thus achieving the gradual transition. The proportionality computation can be performed in an early stage of the processing. The modified face color can be computed at the same point in the processing, or the transition factor can be carried along and used later, depending on details of implementation. This capability is present in recent real-time systems, and has proven extremely effective.

3D Gradual Transition

Gradual transition is also required for three-dimensional objects. The above approach is not applicable for such 3-D objects. Not only does the background for such an object vary from scene to scene, it will generally change from one scan line to another, and even when partially through a face on a given scan line. Thus, no scheme of changing a face color applicable to an entire scene can be expected to work.

As part of a study contract for AFPHRL, WPAFB, gradual transition evaluation scenes were produced using a conceptually simple algorithm. The entire scene was generated to both levels of detail involved in the transition. Transition scenes were then produced by combining these two versions proportionally, one pixel at a time. This demonstrated the effectiveness of the transition concept for features with the background changing along a face. This algorithm however, could certainly not be considered a candidate for real-time implementation.

Translucent Faces for 3D Gradual Transition

Consider a point in a mission when a model of a ship is to be in transition between a low level of detail, and the next higher level of detail. Assume we are at the 25 percent point of the transition -- one-quarter of the way from low to high. An approach we might consider is the following. Process both versions of the ship, but *designate all faces of the low-detail version* as translucent with 25 percent transmission (or 75 percent saturation), and designate all faces of the high-detail version as translucent with 25 percent saturation. Will this achieve the desired goal? Not quite. Even where images from both versions are present (such as the A+B portion of Figure 12), we will see through the ship to whatever is behind it. What is needed here is a third technique for handling multiple translucent faces, in which the saturation is handled in an additive manner. This can readily be implemented by a third mode in the edge modification function following the Orderer.

Figures 13, 14, and 15 show three stages in the transition of a simple model of a ship: low detail, half-transition, and higher detail. These were produced using translucent faces in the mode described. When a video

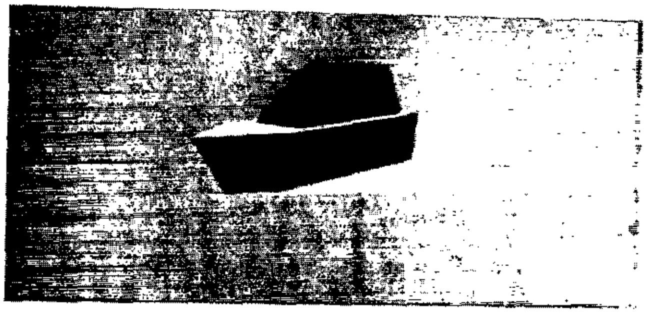

Figure 13. Ship at Low Detail

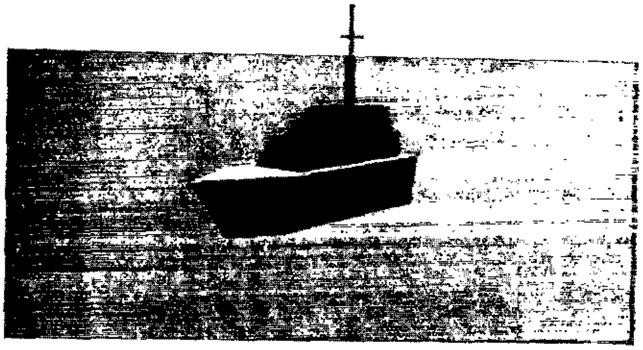

Figure 14. Ship Midway in Transition

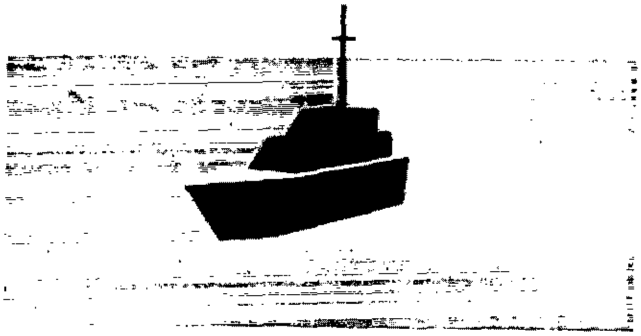

Figure 15. Ship at Higher Detail

tape of this transition is viewed, the effect is very natural and unobtrusive.

Edge Utilization Efficiency

The effect on edge utilization efficiency of simultaneously computing two versions of a model must be considered. As a first comment, even if greater edge capacity is required for a given mission, the elimination of distracting effects could well justify the additional processing. Secondly, assume the transition to higher detail starts at the point where a standard system would abruptly change to the higher detail. The increased edge processing burden would thus be the number of edges in the lower detail version, which will typically be one-fourth to one-half the high-detail edges. Finally, there is a high probability that the natural, nondistracting nature of the gradual transition will mean that level of detail increases can be delayed until the viewer is closer to a model, thus reducing edge processing requirements on a typical scene.

Summarizing the above considerations, it seems highly probable that there will be improved performance with no increase in edge processing capacity required.

CONCLUSION

A technique for simulation of translucent faces has been developed and validated by producing scenes. This approach has characteristics which indicate it should be feasible for efficient real-time hardware implementation. It makes possible realistic simulation of a variety of new features and effects in CIG systems. Probably of even greater importance, it facilitates implementation of gradual level of detail transition for three-dimensional objects, thus increasing realism and adding to edge utilization efficiency.

ABOUT THE AUTHOR

DR. W. MARVIN BUNKER is presently a Consulting Engineer in Advanced Technologies Engineering at General Electric Company in Daytona Beach, Florida. He is currently active in research and development on conceputal, mathematical, and hardware aspects of simulation systems. This applies to perspective display systems such as electro-optical viewing systems, visual display systems, and radar display simulation. He has taught engineering and mathematics at several universities and is a member of the Board of Visitors of Embry-Riddle Aeronautical University. He received a B.S.E.E. degree from the University of Oklahoma, and an M.E. and Ph.D. in electrical engineering from the University of Florida. Dr. Bunker has authored papers in the areas of simulation, instrumentation, computer techniques, and circuit theory.