DIGITAL VISUAL SPECIAL EFFECTS

By

Frank P. Lewandowski
David Hinkle
William Tucker

Advanced Products Operations
Link Division
The Singer Company
Sunnyvale, California 94086

ABSTRACT

LINK has developed techniques that significantly improve the usefulness and appearance of visual scenes without expending an undue amount of visual system capability. Much of this work used face or object substitution, real-time data base manipulation, and further frame and field logic to produce extremely realistic sea scenes including sea state, bow and stern wakes, and land scenes such as smoke, steam, rotor blades, weapons effects, etc.

These techniques will be discussed and a short movie illustrating some of the most dramatic effects will be shown.

INTRODUCTION

First-time observers of digital visual systems often react somewhat negatively to the scene. *No matter how many edges are employed*, the scene does not contain nearly the amount of detail that is apparent in home TV. Even the most detailed scenes appear sterile and lifeless. To sum up the comments of the observers, the scenes are "cartoonish".

Even at the most complex detail level, a computer generated scene is only an abstraction of the real world. No amount of hardware, texture, faster processors, etc. will alter this fact. That such scenes are abstract, however, should not produce such strong negative reactions, for the eyes and the mind abstract constantly from the real world, selecting information from a shifting complex of images.

Interestingly, it has been universally observed that simulated scenes become much more realistic when a pilot is task-loaded. At our plant, where an interactive air-to-air helicopter combat system is in operation, all of the military personnel who have flown in combat have remarked that there is no scintillation, data base error, truncation, nor any cartoonish quality to the display as soon as the "enemy" fires a rocket past their windscreen. It would appear, then, that as soon as some action or movement engages the viewer, the problem of abstraction is very greatly reduced. With this in mind, we decided to substitute movement for detail as a way to *make simulated scenes more realistic*. By adding dynamic background objects we have made the scenes kinetic instead of static, giving "life" to the displays. The movement keeps the eyes and mind busy and does not permit them to concentrate on the lack of detail; the shifting quality of the scene allows the brain to perform, apparently, in a more normal way, selecting focus objects from a variety of images. We feel that we have made a significant advance in the "state of the art" and that what we have been able to accomplish marks only the beginning stages of a process which may aid computer simulation technology enormously.

Only a few of the many special effects that have been developed will be discussed in this paper. These are dynamic sea state, realistic bow and stern wake, ship motion, and semi-opaque images (suspension bridge and rotor blades).

We will show in detail how each effect was generated and present a movie with each segment, shown first in slow motion and then in real-time. All of these scenes were taken directly from the LINK research simulator in the Sunnyvale Laboratory.

SEA STATE

Static 3D Waves

The 3D waves were modeled as small mounds. These "mounds" were scaled smaller than hills and coded blue rather than green. When the results of the first model were displayed, it became apparent that the problem with 3D waves is identical to the problem with the appearance of low hills in nap-of-the-earth flight. The problem is that there is little difference in the shading of the faces that make up the "mounds". When smooth shading is applied, any resemblance to a 3D feature and any indication of shape is totally washed out (See Figure 1).

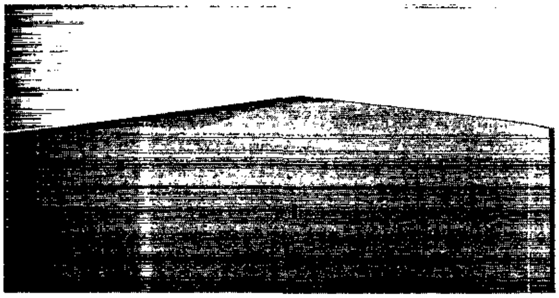


Figure 1 3D Static Waves

On military helicopter programs, we investigated many modeling techniques to alleviate this problem. Among these techniques were overly accentuating sun shading, outlining faces in 3D objects, and placing a checkerboard pattern on 3D features. Whereas these techniques can be somewhat accepted by the viewer in terrain (there are regular field patterns in many areas), they are totally unacceptable in an ocean scene.

Moving 3D Waves

Translating the 3D pattern would not change its appearance (over a static 3D pattern) because the sun angle with respect to the faces would not change. Therefore, several other techniques were implemented. These techniques included rotating the 3D objects, switching 3D objects in and out of the scene (leveling), etc. All were difficult to implement and did not give promise of providing a useful and desirable visual effect even if the implementation were successful.

2D Sea State

The first attempts to develop sea state were made utilizing a static 2D pattern. Several different regular patterns were modeled and displayed. Although these pattern types have been tentatively accepted for nap-of-the-earth flight, their regularity makes them disconcerting for expanses of water. Originally it had been assumed that the patterns could be modeled with subtle differences in intensity so that the appearance would not be objectionable but the eye is much too sensitive to even slight intensity changes, particularly along straight lines between faces. An example of a non-smooth-shaded regular pattern with many intensity levels is shown in Figure 2.

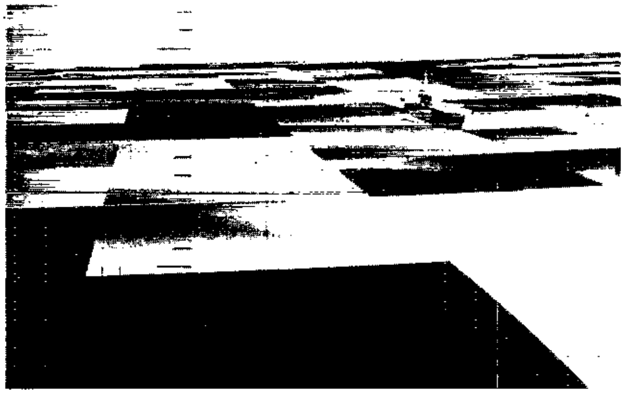


Figure 2 Ocean Surface
(Without Smooth Shading)

These patterns were also modeled so that they could be smooth shaded. That is, the intensity between faces was extrapolated along each display scanline to eliminate the sharp discontinuity in intensity from one face to the next. It was presumed that this "smoothing" in a 2D pattern would be sufficient to give an adequate visual impression of ocean surface.

The pattern of Figure 2 was smooth shaded and is shown in Figure 3. This is obviously an improvement over a non-smooth-shaded sea state. However, the scene appeared much too static and lifeless to properly convey an ocean surface. For this reason, it was decided that a method of changing the apparent intensities of the ocean patterns as a function of time would be necessary.

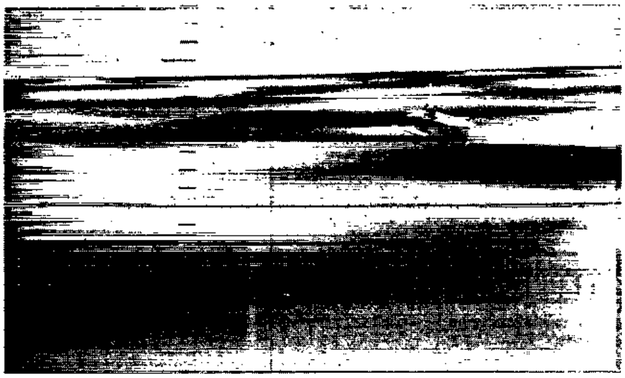


Figure 3 Ocean Surface
(With Smooth Shading)

Dynamic 2D Sea State

The basic approach in generating a dynamic sea was to model a pattern of smooth-shaded faces with modified vertex normals. Since the cross-product of the vertex normals and the sun vector controls the shading of the faces, many different patterns can be generated by modifying vertex normals while the contrast of these patterns can be increased by moving the sun downward from the zenith. The latter is shown clearly in Figures 4, 5, and 6. Figure 4 was taken without a sea pattern and Figures 5 and 6 show the contrast of the smooth-shaded sea pattern at two different sun angles.

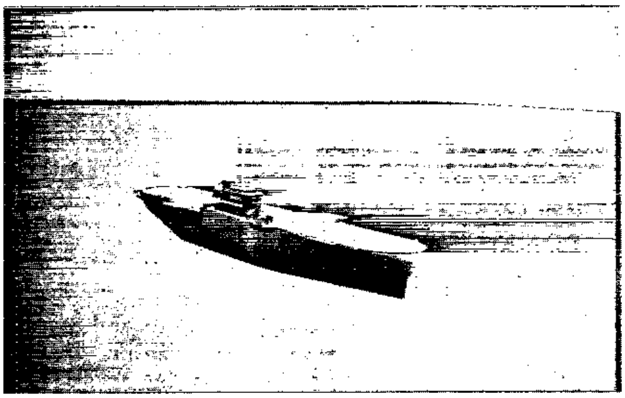


Figure 4 Sea State 1
(No Pattern Contrast)

The initial implementation of this approach was to modify the vertex normals in the object descriptions of the sea pattern in a predetermined sequence. This would cause the faces to change in a cyclic sequence. It was determined that it was necessary to update the normals at display frame rates to prevent noticeable stepping between patterns.

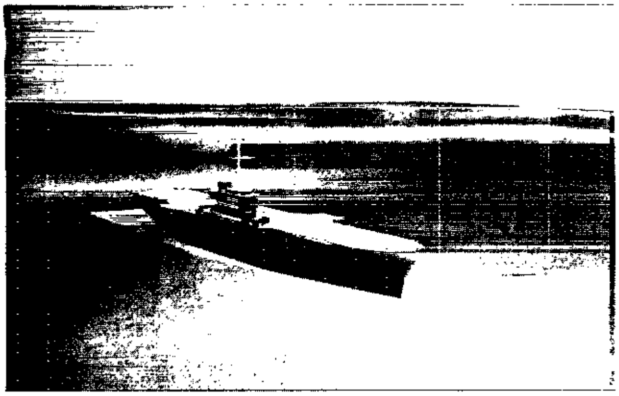


Figure 5 Sea State 1-2
(Moderate Pattern Contrast)

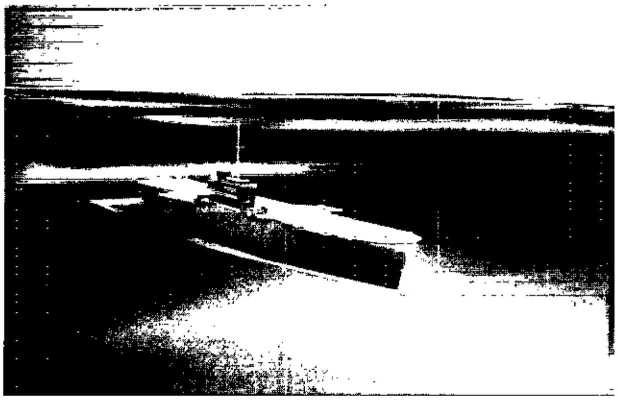


Figure 6 Sea State 2-3
(High Pattern Contrast)

Updating the normals at frame rates proved to be an extremely difficult software task and the time necessary for its execution exceeded the available Central Processing Unit (CPU) time. A simpler method of implementation was then developed as follows (See Figure 7). A single sun vector is normally sent each frame for all objects in the data base. By separating the objects for the sea pattern, it was possible to assign a second sun vector for this particular group of objects. Computing only a new sun vector each frame for the entire sea pattern made it possible to achieve the same visual appearance as the original attempt at implementation (updating each of the vertex normals), but with considerably less real-time software. The single sun vector computation for the sea state required less CPU time in the real-time software than in the initial vertex normal implementation. With only one vector to change, it was also possible to use a more complex vector modification algorithm.

The sun vector, when rotated in a circular motion, causes the sea patterns to change realistically. Changing the rotation rate simulates various agitations of ocean surface. Figures 8 and 9 show how the intensity of each square in the sea pattern changes as the sun alters its position as a function of time. The smooth shading was turned off for these pictures

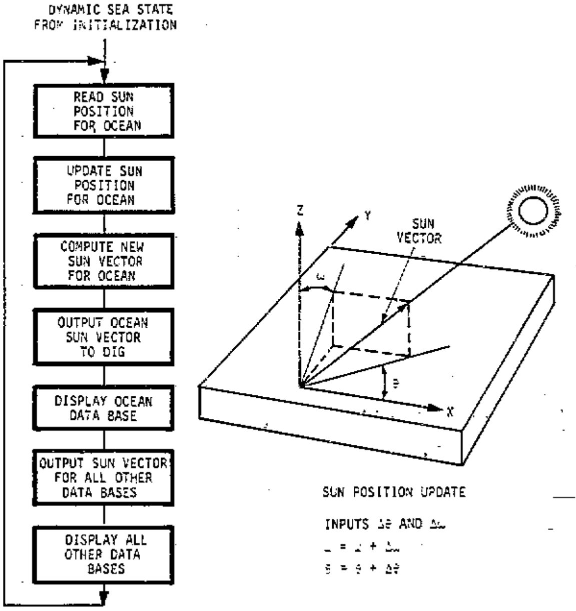

DYNAMIC SEA STATE
FROM INITIALIZATION
READ SUN POSITION FOR OCEAN
UPDATE SUN POSITION FOR OCEAN
COMPUTE NEW SUN VECTOR FOR OCEAN
OUTPUT OCEAN SUN VECTOR TO DIG
DISPLAY OCEAN DATA BASE
OUTPUT SUN VECTOR FOR ALL OTHER DATA BASES
DISPLAY ALL OTHER DATA BASES
Z
Y
X
SUN VECTOR
ω
θ
SUN POSITION UPDATE
INPUTS Δθ AND Δω
ω = ω + Δω
θ = θ + Δθ

Figure 7 Dynamic Sea State Techniques

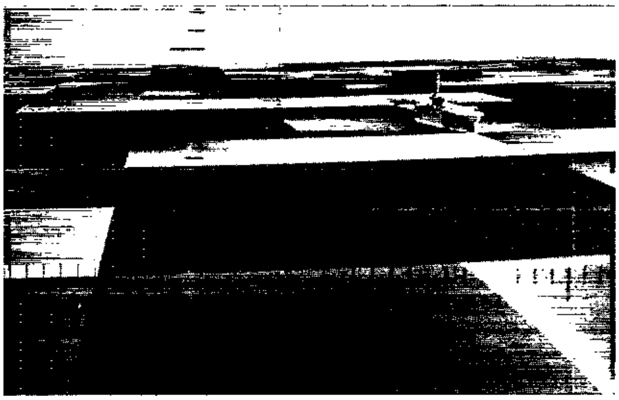


Figure 8 Ocean Surface Time $=t_1$
(Without Smooth Shading)

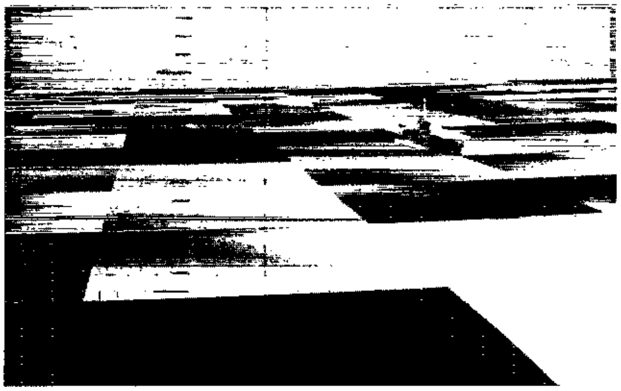


Figure 9 Ocean Surface Time $=t_2$
(Without Smooth Shading)

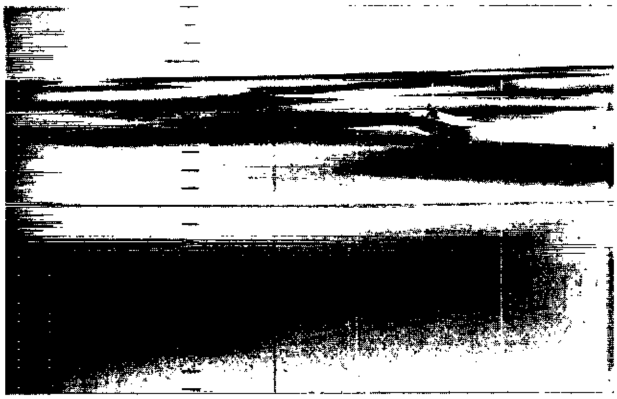


Figure 10 Ocean Surface Time $=t_2$
(With Smooth Shading)

to make the technique obvious. Figure 10 is the sea pattern shown in Figure 9 with the smooth shading turned on. This effect, coupled with the fact that the angle of the sun vector controls the contrast on the pattern (as described previously) is useful in generating a number of different sea states.

Three different patterns of vertex normals were programmed and displayed:

1. When the vertex normals were assigned in a random basis, the resulting dynamic pattern was that of a confused sea.
2. When the vertex normals were computed as a sine wave on odd rows and its mirror image on even rows, the pattern looked unrealistically distinct.
3. When the vertex normals were computed as a sine wave on all rows, the pattern appeared as a series of bands moving in parallel, not unlike ocean swells.

Of the three patterns mentioned, the random pattern appeared to be the most acceptable visually and was incorporated into the NASA /Ames Digital Visual Simulator.

BOW AND STERN WAKE

Face Substitution

In an attempt to generate moving objects such as the ship bow wake, many different schemes were considered. Usually these amounted to a wake moving object which traveled with the ship. Efforts were made to make the complex bow-wave motion using the simplest real-time motion possible. Initial attempts made the moving object in the shape of a cam. As the object rolled about its axis (as it moved with the ship), it would continuously present a different shape (and height above water) to the eye. The implementation of this scheme proved to be quite laborious and required two real-time moving objects (one for each side of the ship) involving many scene edges.

Attempts to simplify the "moving object" procedure indicated that a method would have to be developed that was a radical departure from the straight-forward approach. The movie industry animation methodology was employed and a series of sequenced faces were used to make an object enlarge and decay in a programmed fashion. A procedure was developed whereby all the sequential faces were made part of the moving object with which they were associated--in our case, with the ship moving through the sea. This procedure is shown graphically in Figure 11. Four actual frames taken from the display are shown in Figure 12. As the moving object was transferred from the active data base, one of the bow-wake faces was transferred with it and subsequently displayed. After a fixed number of frames, a pointer would substitute a second face for the first and subsequently display it with the ship. The rate at which the subsequent faces were transferred and the difference between subsequent faces produced the dynamic movement. The selection of these two parameters along with the selection of various colors can be used to produce a wide range of effects.

This technique of effectively modifying the digital data base in real-time gave rise to many of the other effects discussed herein.

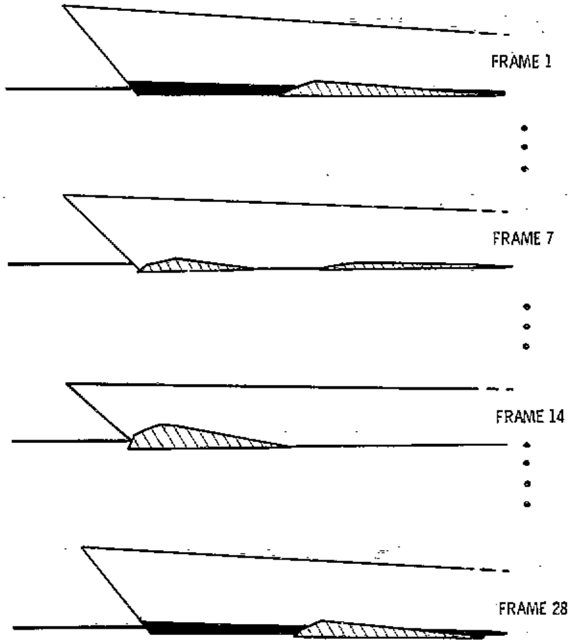


Figure 11 Moving/Growing Objects
(Bow and Wake)

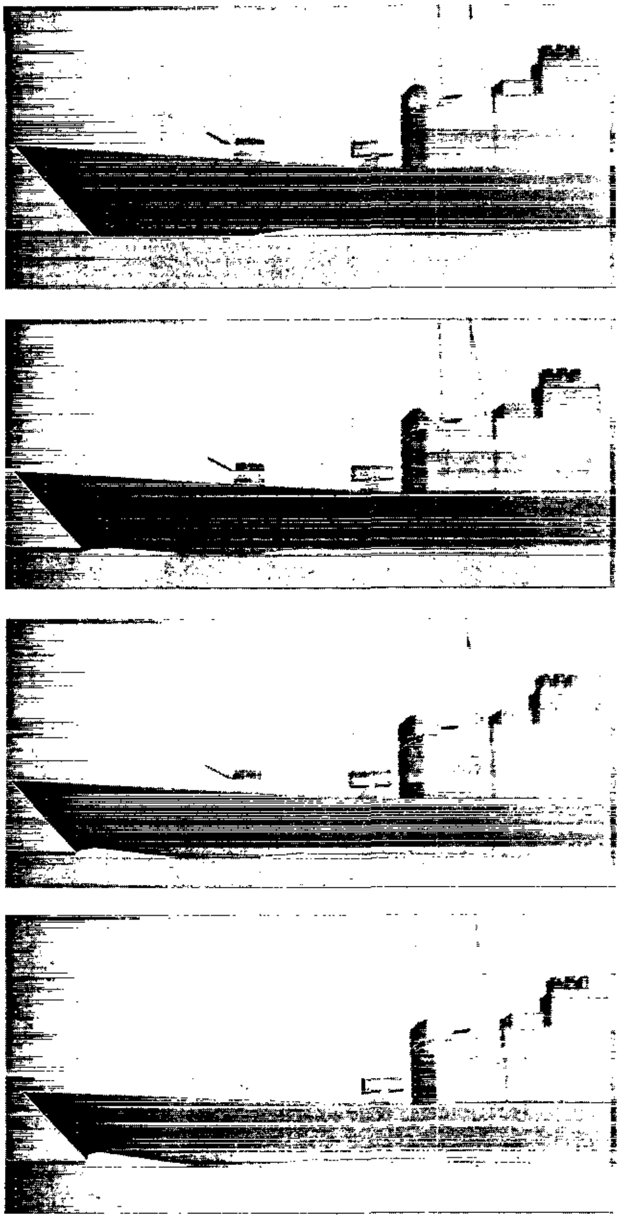
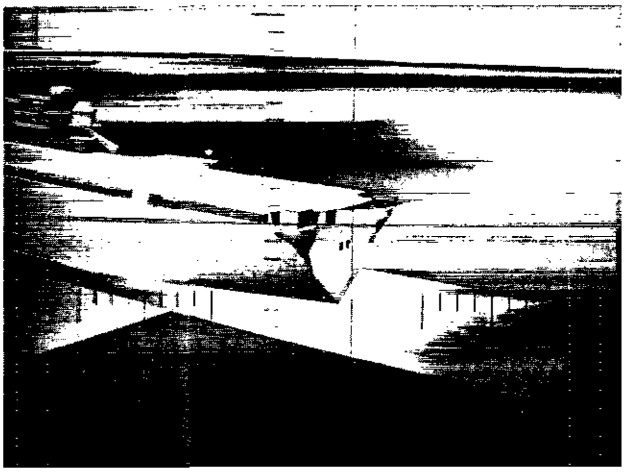


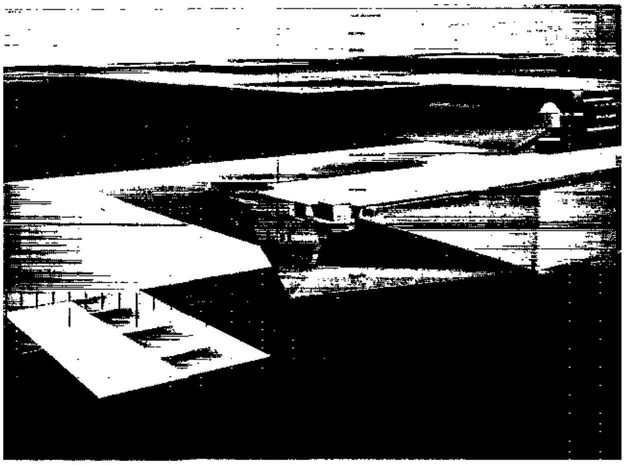
Figure 12 Four Faces (Not Contiguous) of the 28 Faces Comprising the Dynamic Bow-Wake Pattern

Use of Modified Sea State Pattern

The bow wake generated was extremely realistic when viewed from an eye position close to the surface of the sea. However, when observed from near vertical the faces comprising the bow wake become almost invisible (the faces have no thickness). Techniques to make the bow wake a 3D pattern were complex. It was decided that the design approach used to implement the sea state pattern could also be used for bow and stern wake. Several patterns, pattern colors, pattern sizes, sun vector angles, and sun vector speeds were modeled and displayed on the research simulator. After several iterations when the speed of the sun vector rotation was increased substantially, a "churned water" effect was created. Figure 13 shows the bow and stern wake taken from the research simulator.



Bow and Stern Wake - LPH2 (Without Smooth Shading)



Bow and Stern Wake - LPH2 (With Smooth Shading)
Figure 13

SHIP MOTION

Occulting (Ship/Sea)

The realistic portrayal of sea state demands a substantial amount of ship pitch and roll. Other ship movements, such as heave and sway, are important to landing/takeoff simulation but do not make additional demands on the visual system. Discussions with Navy personnel established that landing and takeoff can reasonably be expected to be performed with pitch less than 5° and roll less than 8°.

The worst case for ship/sea occulting would be from a position close to the ocean surface. Viewed from this position, it appeared necessary to model to some distance below the waterline so that the hull can be seen when the ship is pitched up, and conversely, less hull above the waterline can be seen when the ship is pitched down. This causes a mutually exclusive visual object priority problem. If the ship has higher priority than the ocean, the ship can be seen below the waterline to the point that it was modeled (this makes it appear to "float" on the ocean at the lowest point digitized). On the other hand, if the ocean were to have higher priority than the ship, the ship would not appear except for the area above the horizon. Any occulting schemes that involved paths or seams in the ocean were discarded because they would permit only preprogrammed ship motion. The following is a discussion of the experimental methods.

"Moving Box" Occulting Method. Occulting the part of the ship which is below the water can be accomplished by positioning a box (open at the top) around the ship from the waterline down. The box must be the same color as the surrounding water and must have a higher priority than the ship.

The assignment of color, intensity and priority of the box are easily accomplished. Color and intensity are chosen during modeling. The priority of the box is accomplished by making the box a separate object list and setting its priority during initialization of the real-time program to be greater (lower number) than the ship's parts.

The accuracy of this method decreases with increasing altitude of the viewing point. This accuracy can be maximized by making this box as close to the shape of the occulted parts of the ship as possible. The ship then appears to rise from and disappear into the sea when it actually is being occulted by the moving box.

This method was implemented on the research simulator and is extremely effective. However, a simpler method (using no moving object and fewer edges) is more desirable.

Face Substitution Method. It was originally assumed that if ships were to be modeled only to the waterline (effectively "floating" on the ocean surface), an eyepoint close to the water would detect space below the bow and stern when the ship pitched up and down. Before the modeling of a ship was completed, several methods of occulting other than the moving object method discussed in the previous paragraph were investigated. The most promising of these was a modification of the face substitution method used for the bow wake. Basically, it involved using a series of wedge-shaped faces of progressively larger sizes to "fill" the space between the bottom edge of the ship (waterline) and the surface of the ocean as the ship pitched up, and the removal of the wedge-shaped faces as the ship pitched down. These would be inserted (or removed) in real-time as a function of the angle of pitch (See Figure 14).

(Roll is not a problem because the face used on the bottom of the ship appears to be simply the extension of that portion of the ship below the waterline exposed during roll.)

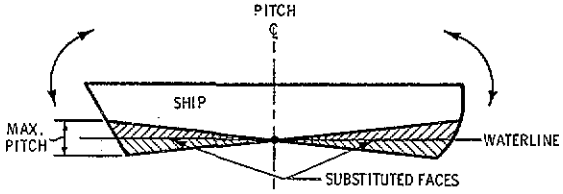


Figure 14 Face Substitution

Necessity of Occulting

When the modeling of a Landing Platform Helicopter (LPH) was completed, it was programmed as a real-time moving object and driven realistically through the ocean data base. Pitch was programmed in as 10° which is double the maximum worst case excursion permissible in landing/takeoff. The LPH in these experiments was modeled only to the waterline. The completed scene was dynamically demonstrated to numerous pilots. At no time during any of these demonstrations was any objection raised, nor was attention called to the fact that when the ships pitched down, the length of the prow did not shorten. In fact, the comments were unanimous that the ships motion was "realistic". The movie that accompanies this paper bears out this contention.

SEMI-OPAQUE IMAGES

Suspension Bridge and Rotor Blades

A technique of substituting visual scene faces in real-time was demonstrated as being feasible and useful in the creation of the bow wake previously discussed. The technique provided a means of generating special effects that achieved a high degree of realism.

The basis of the face substitution procedure was the concept of manipulating the visual data base in real-time. Heretofore, the data base was produced off-line and simply called up when required as various levels of detail (usually as a function of range) or as an entire moving object (aircraft, ship, etc.). The face substitution technique showed how effects that required rapid image movement could be produced easily.

The technique was further expanded to include object substitution as well as substitution of objects/faces on alternate fields or frames to produce a semi-opaque effect. The semi-opaque effect comes about by displaying the entire scene on one field or frame and the object desired to appear semi-opaque on the next field or frame. Since the priority of the semi-opaque object would obscure the scene behind the object only every other frame, the eye would alternately see both (See Figure 15). The effect is the appearance of a semi-opaque object. The following describes how the technique is implemented and some of the interesting effects produced.

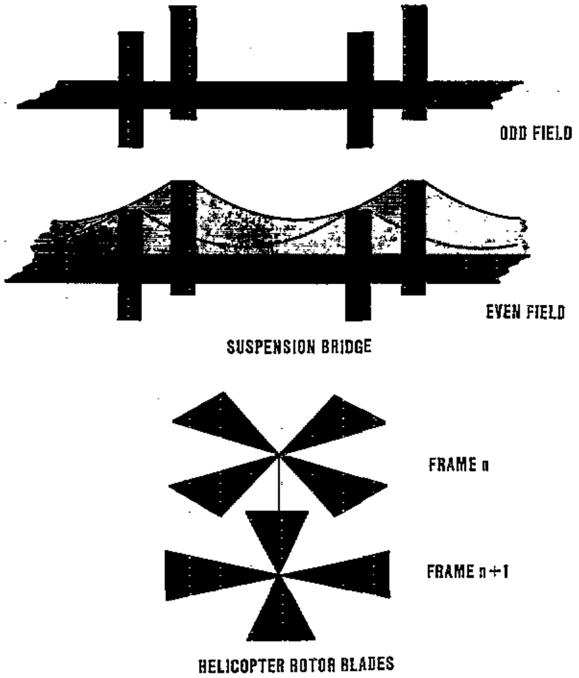


Figure 15 Semi-Opaque Objects

Object substitution is accomplished by modeling the data base with a dummy object in the location of the special effect. This dummy object is not displayed but is used for generating planes so that the special effect will have the correct priority. The desired special effect is modeled as a series of objects in the same manner as a cartoon is generated. These objects are then sequentially displayed at object substitution rates dependent upon the particular application.

The suspension bridge is modeled so that during the odd television display fields only the bridge towers and bridge deck are output to the object list (See Figure 16). This permits the scene behind the bridge to be displayed. During the even fields the towers and deck are again displayed (hence appear totally opaque) and the objects that make up the sides of the bridge are also displayed. This gives an excellent impression that the city (or any object) behind the sides of the bridge is only partially obscured by the bridge wires.

An actual image taken from the research simulator display is shown in Figure 17. It clearly shows how effectively this technique can be applied.

There is another factor involved here which may not be obvious at first--the repetition rate of the displayed image. In the case of the bridge, the image cannot appear to flicker unrealistically; hence it must be displayed faster than 20 times per second. To accommodate both the update rate requirement and the alternate image requirement, it was necessary to update the digital visual system 60 times per second.

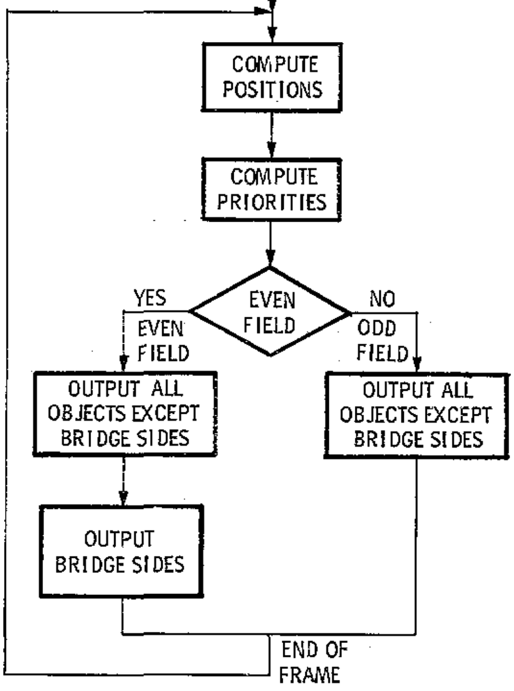


Figure 16 Semi-Opaque Objects (Alternate Field Technique)

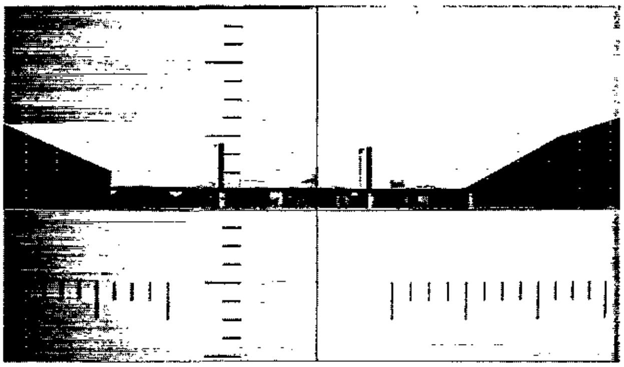


Figure 17 Suspension Bridge

The second example presented here is the helicopter rotor blades. The object substitution technique is used to provide a semi-opaque quality to the spinning rotor blades. It should be noted that in this case two 45° segment German crosses have been modeled so that they are rotated 45° in the data base (providing a complete solid disc) and displayed on alternate frames (rather than alternate fields as in the case of the bridge). The block diagram is shown in Figure 18. This provides the semi-opaque effect as discussed previously. In addition, displaying each set of blades on alternate frames rather than alternate fields as noted previously provided an unexpected effect. The effect of the "flicker" resulting from the 15 image per second update (each blade pattern every other frame) gives a realistic impression of blade movement that was impossible to obtain in any other manner.

In this case, an undesirable attribute of a discrete image system is being used to advantage. *Figure 19 shows the semi-opaque quality of the* blades, but unfortunately only direct viewing on the display CRT or in a movie can show how this simple technique provides realistic blade "motion".

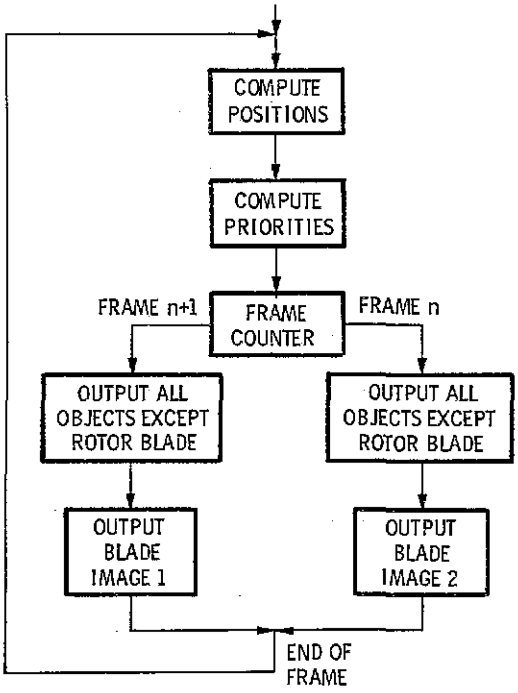


Figure 18 Semi-Opaque Objects (Alternate Frame Technique)

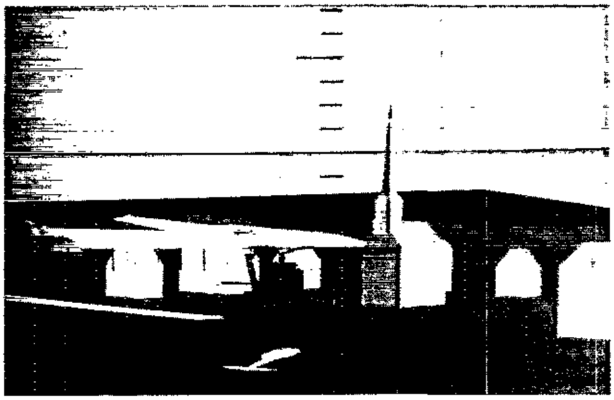


Figure 19 Helicopter Rotor Blades

CONCLUSION

Two important digital visual techniques have been discussed in this paper. The first was the real-time data base manipulation procedures which were used for producing rotor blades, bow wakes, and weapons effects without resorting to the use of moving objects. The second was the separation of the object list in the data base so that separate sun vectors for different sets of objects could be output to the image processor. In the latter case, the appearance of dynamic sea state resulted from the use of a constantly changing sun vector and a randomly modeled blue checkerboard. The results of the application of just these two techniques are an indication of the kinds of visual effects that remain to be implemented on digital visual systems with little more than imagination.

Both of the techniques used can also be considered successful because little additional data base is required, few additional edges have to be processed, and only an insignificant amount of real-time is required. In these particular cases, it can be stated that the visual impact of the resulting scene is far greater than the small amount of visual system capacity that it uses.

ABOUT THE AUTHORS

Mr. Frank P. Lewandowski, Senior Scientist, with LINK Division of The Singer Company. In this position he is investigating the possible range of visual effects in digital image generation systems. Mr. Lewandowski received his Bachelor of Science Degree in Electrical Engineering and in Mechanical Engineering from the University of Illinois.

Mr. William Tucker, Senior Engineer, with LINK Division of The Singer Company. Mr. Tucker has been involved with the LINK Digital Visual System since its inception in 1972. He has been instrumental in many of the design innovations which have improved the imagery generated by the DIG. He has also participated in many of the unique data base generation schemes which have resulted in the imagery included in this paper.

Mr. David Hinkle, Senior Programmer/Analyst with LINK Division of The Singer Company. Many of the data base generation and manipulation programs described in this paper are the result of Mr. Hinkle's efforts. Since he joined LINK, Mr. Hinkle has been involved in studies to achieve real-time data base management and new methods for creating edge efficient data base object models.