VISUAL AND INFRARED SHIP MODELING FOR COMPUTER IMAGE GENERATION*

Richard J. Mitchell
Anthony J. Stenger
Technology Service Corporation
Santa Monica, California 90405

John L. Booker
Naval Training Equipment Center
Orlando, Florida 32813

ABSTRACT

Programs for generating simulated visual and far-infrared imagery of Soviet combatant ships were implemented on a computer system at the Naval Training Equipment Center. These programs were used in a system definition study to determine the characteristics that a real-time computer image generation system must have in order to satisfy trainer performance requirements for processing capacity and resolution. The implemented system consists of data bases at various levels of detail, visual and far-infrared sensor models, and image generation capabilities that include a flexible set of options for various viewing conditions and special features.

INTRODUCTION

The Naval Training Equipment Center (NTEC) is developing the Advanced Visual/Near-Visual Submarine Periscope/Electro-Optic Infrared Sensor Simulation (AVEOSS) prototype trainer. As part of the supporting R&D effort, Technology Service Corporation (TSC) implemented general computer image generation (CIG) programs as part of a system definition study to determine the characteristics necessary for the trainer system. The scenario involves the Osa, Kashin, and Kiev Soviet combatant ships viewed through both visual and far-infrared sensors.

Because the most significant parameters in the definition study were level of detail and resolution at varying ranges and aspects, the CIG programs and models had to allow the user complete operational control. To this end, the geometric ship models were constructed from planar surfaces, ranged in level of detail from 500 to 4000 potentially visible edges, and could be viewed from any aspect angle and range with display formats from 512 to 2048 raster lines per TV frame.

This paper describes the CIG programs and data bases implemented at NTEC in terms of geometric data base modeling, visual and infrared models, and frame generation characteristics. The data bases developed use modular, hierarchical construction so that complex scenes can be processed and sensor-ship interaction can be blended with the hidden-surface and shading requirements for image generation. The visual and infrared sensor algorithms include reflectivity models and more complex components for determining the radiant emittance of a surface in the far-infrared spectrum. The sensor models incorporate the effects of ship surface materials and the intervening (user-selected) maritime environment. The frame generation routine employs a list-directed priority algorithm for determining hidden surfaces. Included along with the user-selected parameters for level of detail, range, aspect, resolution, and magnification are options for edge smoothing and smooth-surface shading.

The CIG methodology used in the system is described next, followed by sample imagery of the Kashin and Kiev models taken from a high-resolution Dicomed image recorder. Recommendations for further efforts in data base modeling are given in the last section.

CIG METHODOLOGY

The system implemented at NTEC (see block diagram in Figure 1) generates shaded computer imagery in three distinct steps. First, a scene (e.g., a ship on the ocean) and the objects composing it are stored in a geometric data base in a meaningful and useful format. Second, the tone (or gray level) of each surface is predicted. Third, the surfaces are projected onto a screen and a shade assigned to each pixel for the surfaces viewable in that pixel.

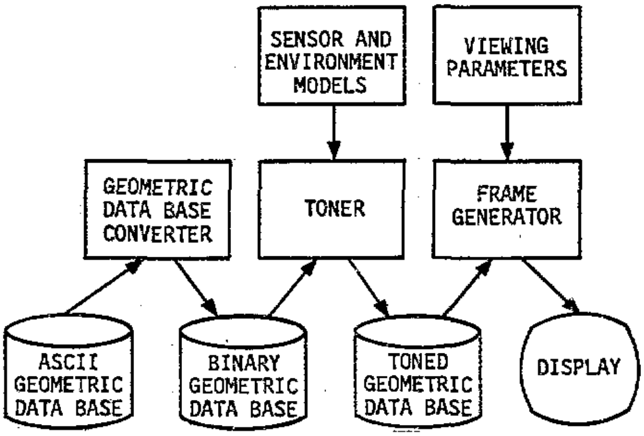


Figure 1. General CIG System and Interface

The geometric data base consists of planar polygonal bounded surfaces defined by vertex points. A material identification code is associated with each surface. Surfaces that are locally coherent are grouped into objects. A list-directed priority scheme employing a tree-structured directory is used in organizing the data base in order to determine hidden surfaces. The data base converter in Figure 1 performs three primary tasks:

---

[*] The work leading to this paper was performed under NTEC Contract N61339-79-C-0151.

1. Convert the data base from ASCII to a binary format.
2. Set up a tree-structured directory with pointers to the surfaces.
3. Compute the outward normal of each surface.

Since these tasks are performed early in the process, they do not have to be repeated for each frame generated. Thus, while subsequent processing is repeated as conditions change, data base conversion is performed only once in the life of a data base.

The toning process for visible and infrared sensors is performed independent of frame generation. The toning routine may be thought of as a "black box" which computes, for a given sensor and set of environmental conditions, a single tone for an entire surface or a tone for each vertex of a surface. Material descriptions are associated with each surface to provide visible reflectance, interior conditions, absorptivity, emissivity, surface orientation, and surface roughness. Environmental conditions include solar and sky loading, air and water temperature, and humidity.

The frame generation routine is central to the system. In general, all other elements of the system support the frame generator. With a toned data base and viewing parameters as input, the frame generator outputs a synthetic image via two subprocesses:

1. Geometric processing, in which the surfaces of the data base are ordered, translated, and rotated relative to the viewer, and then clipped against planes bounding the field of view.
2. Raster processing, in which each raster is generated by determining the viewable surfaces in each pixel and then assigning the appropriate shade for each pixel.

Geometric Data Base Modeling

Basic Geometric Considerations. The basic primitive used to build the geometric model is a one-sided polygonal bounded surface that can be convex or concave. It is defined by a list of vertex points, ordered counter-clockwise from the viewable side. Since each surface is planar, it possesses a unique outward normal. Locally coherent surfaces are grouped together to form an object. At the object level, a number of useful operations may be performed, most notably redefinition or specification of the level of detail and rapid elimination of an object's surfaces from the field of view.

Many of the planar surfaces in a ship data base are patchwork approximations to curved surfaces. To perform curved-surface shading and make the surfaces appear smooth, the surface normal must be supplemented by defining an outward normal for each vertex. Although in some cases, such as the ship's hull, the actual contour cannot be determined, selected points on the contour can be located. In such a case, a vertex normal can be approximated by averaging the normals of surfaces that share the common vertex. In many cases, simple geometric entities such as spheres, cylinders, and cones (either complete or truncated) can be originally specified for objects. The objects can then be transformed into a patchwork of surfaces.

The use of geometric entities which can be parametrically specified during data base development impacts CIG in two significant ways: 1) the contours are known exactly and, consequently, so are the vertex normals; 2) the level of detail needed to represent an object at various ranges is easily adjusted. As the chosen level of detail for a data base increases, so does the fineness of the patchwork used to approximate the curved object. For example, for the tubular superstructure of a tower, cylindrical approximations can be used for high detail at close range, whereas only a single strip is needed for lower levels of detail.

Data Base Organization. Local surfaces listed by their vertices constitute the basic representation of an object in a geometric data base. However, the overall design of a geometric data base can significantly impact CIG system capabilities. In other words, while the basic geometric data are necessary for generating computer imagery, the organization of the data base may help to solve problems such as hidden-surface determination as well as allow the structured, modular development of complex scenes.

The hidden-surface problem encountered in the frame generation process is essentially solved a priori by using a list-directed priority scheme when developing the geometric data base.[1,2] The drawbacks of this solution are that it somewhat restricts the arrangement of surfaces and objects and increases the cost of data base development. On the other hand, it decreases the cost of frame generation because surface masking priority is already determined and thus does not have to be repeated in other CIG routines such as specular reflection and shadow generation.[3]

The list-directed priority scheme uses two distinct methods to establish the priority list. First, all surfaces comprising an object are listed according to their inherent masking priority within the object such that no surface may mask any other surface listed before it, regardless of the viewing position. Inherent masking priority restricts surfaces from intersecting each other anywhere but at their edges. For a convex object, any surface order is acceptable because no surface of a convex object will ever be masked by another surface. (Note: Since surfaces are one-sided, surfaces facing away from the viewing position are not considered masked.) For a concave object, inherent masking priority may not exist, in which case the object can be divided into smaller objects to solve the masking problem.

The second method used to establish the priority list of all surfaces is to subdivide the scene into objects. Once the surfaces within each object have their inherent ordering established, the remaining task is to sort the objects. To do so, a tree structure was implemented which can modularly create a complex scene by combining simple objects to form more complex objects. Or, as illustrated in Figure 2, a complex scene (such as ships at sea) can be broken down into simple objects and geometric primitives.

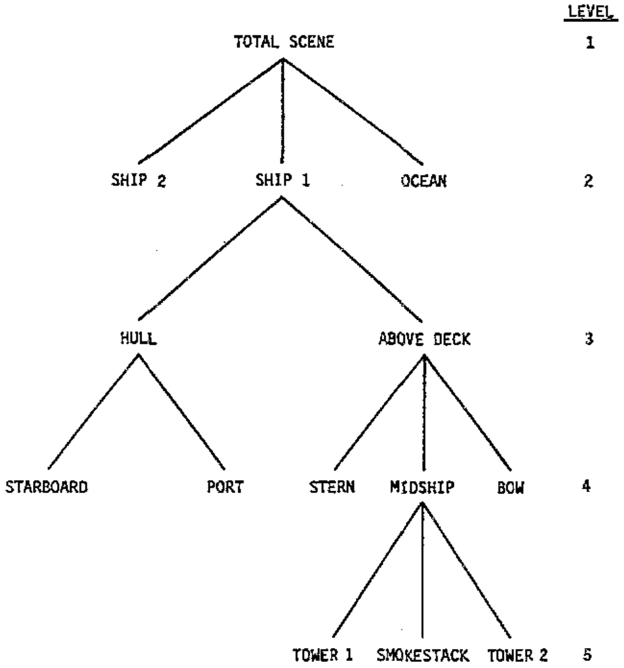
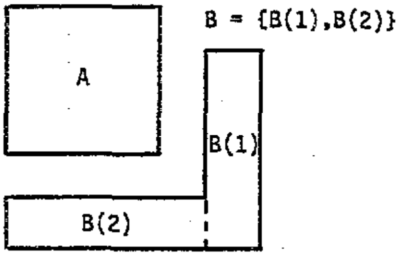


Figure 2. General Tree Representation of Ship Geometric Data Base

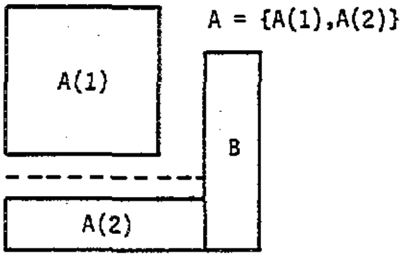
To fully understand the tree structure used, definitions of some terms and relationships are necessary. A node is any point on the tree and consists of one or more objects grouped together. A node's descendants (subnodes) are all nodes subordinate to it. The node's closest descendants (i.e., those in the next immediate level) are designated as the node's children. A node is considered a parent to its children and an ancestor to all its subnodes/descendants. Nodes sharing a parent are siblings. Thus, in Figure 2, the node TOTAL SCENE is an ancestor to all nodes of the data base and the parent of its children SHIP 1, SHIP 2 and OCEAN, which are themselves siblings. No other direct relationships are defined and no significance is attached to the relative levels of two nodes except as to their being descendants, ancestors, or siblings. For example, it is of no consequence that the nodes for the PORT side of the hull and the MIDSHIP section above the deck are both at level 4. The number of levels, nodes of the tree, and children of any node are unlimited. A node with no subnodes has only the geometric primitive surfaces as its children.

For each node, priority tests determine the ordering of its children. Once the children (siblings) have been ordered, the ordering of all their descendants (with respect to the siblings) is established by their transitive properties. For example, if SHIP 1 masks SHIP 2 for a particular eye position, so do all descendants of SHIP 1. For this ordering scheme to be correct, all siblings must be pairwise linearly separable by a separating plane. Stated another way, a node may not intersect the convex cover of any node other than its ancestor or descendant. If such an intersection does occur, the masking relationship between a node's siblings and its descendants is not transitive and thus the masking relationship between unrelated nodes is not known. For some situations, this nonintersection rule may be relaxed during data base development if it is known that the eye will be restricted to certain regions of space, such as the upper hemisphere (i.e., above the ground or ocean).

Figure 3a illustrates a simple, 2-D case in which the nodes have been incorrectly formed and thus are not linearly separable. B(1) and B(2), separated by a dashed line, have been combined to form their parent node B. However, since this parent is not separable from node A by a plane, from certain viewing positions part of B will mask A and A will mask some other part of B. In Figure 3b, A(2) (which is B(2) in Figure 3a) has been combined with A(1) to form parent node A. In this case, nodes A and B are linearly separable and thus the masking relationships are transitive. Hence, planes to separate the nodes may be defined within the data base and used to order objects once viewing position has been specified. During frame generation, A and B will be ordered, followed by A(1) and A(2). If further subdivision takes place, ordering continues until all objects are ordered. At that point, the masking problem will be solved because each object's surfaces were previously listed by their inherent masking priority.



a. Nonlinearly Separable Nodes



b. Linearly Separable Nodes

Figure 3. Node Formation

The geometric data base is stored as a separate directory and a surface list. The directory represents each node by a unique number and contains general, relevant information associated with each node (such as its children's node numbers and separating planes as well as the node's local origin and bounding box). The surface list contains each object's surfaces in sequence, along with vertex lists, surface and vertex normals, material descriptors, and an associated object node number. Whereas the surfaces of an object must be listed sequentially, the objects themselves can be listed in any order. When the data base is converted from ASCII to binary format, pointers from the directory to the first surface of each object in the surface list are established to provide random access to any object.

The tree-structured directory offers more than just a solution to the hidden-surface problem. For instance, any portion(s) of the data base can be viewed by specifying desired nodes. By default, the total scene is displayed. In addition, the data base can be significantly modified through the directory. For example, by changing the local origins, objects or large nodes (such as an entire ship) can be moved. However, when modifying local origins, the separating planes between sibling nodes may have to be redefined, and motion must be restricted so that nodes do not intersect.

Visual and Infrared Toning

Toning is the CIG phase in which electromagnetic models are implemented to predict a gray level for a surface, or a portion of it, based on the sensor model, environmental conditions, and surface characteristics. The toning models are essentially independent of each other and of frame generation. For the visible band, gray levels correspond to the percentage of total light reflected. For the infrared band, gray levels correspond to the emitted and reflected infrared flux density. The predicted gray level replaces the material code for each surface or each vertex of a surface in the geometric data base.

The current implementation for the visual model is based on Lambertian diffuse reflection. For this model, the tone computed is independent of eye position and is based on the cosine of the angle between the surface normal and a light source (e.g., the sun). An additional component is included to account for the sky as a light source.

The infrared model represents a more complicated situation in which material type, internal temperature loads, air temperature and humidity, and the contributions of surrounding bodies are major factors in predicting gray levels. A further complication is that, while the model is diffuse and thus independent of viewing aspect, attenuation due to range can be significant. The following discussion explains the infrared model in more detail.

Thermal Emission. The radiant emittance (the power per unit area of a black-body emitter) with a spectral band bounded by minimum and maximum wavelengths λ_1 and λ_2, respectively, is given by

$$W = \int_{\lambda_1}^{\lambda_2} \frac{c_1}{\lambda^5} \frac{\lambda d}{\left(e^{c_2/\lambda T} - 1\right)}$$

where T is the absolute temperature of the body, and c_1, c_2 are constants. For surfaces at or near room temperature, a substantial amount of thermally emitted radiation is in the 8 to 12 μm spectral band. In addition, a surface reflects thermal radiation from other, nearby emitters such as the sky and water. Thus, the total emitted energy from a surface is approximately the sum of its own thermal emission and the reflected energy of its surroundings.

Infrared Gray Level Prediction. The gray level of a surface is directly proportional to the surface's radiant emittance. To predict a surface's radiant emittance, its absolute temperature at any arbitrary time must first be predicted. Surface temperature is a function of the thermal properties of the material, the instantaneous temperature of the surroundings, and the previous temperatures of the surroundings.

In the infrared toning model, an environmental file is set up for the scene. This file contains parameters such as longitude, latitude, day of the year, data base orientation with respect to North, and hourly air temperatures for the 24 hours prior to the "day" the ship imagery is being simulated. From these parameters, the radiative and convective heat load on the exterior surface at any time can be determined. The interior surface is assumed to be exposed to a constant interior ambient temperature. The conductive path between the exterior and interior surfaces can be modeled as an equivalent RC electrical network. The surfaces in the ship data base are assumed to be metal plates, and the equivalent RC network model contains eight nodes, as shown in Figure 4.

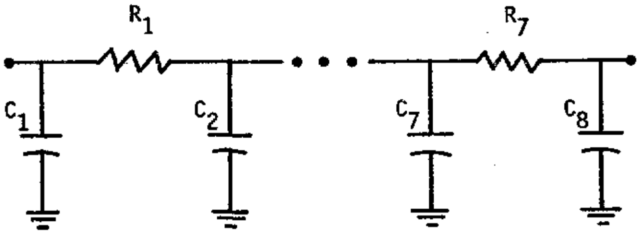


Figure 4. RC Network Model for Thermal Conduction through a Metal Plate

To determine the precise temperature at each node requires an eighth-order differential equation. An approximation is obtained by solving a system of eight first-order linear difference equations. Thus, a reasonably accurate calculation can be obtained for the exterior temperature of all surfaces in the data base.

Gray levels are predicted in two steps. First, the environmental data are used to calculate the exterior temperature of metal plates whose normals are oriented in twelve standard directions, and a look-up table is generated. Second, each surface in the data base is addressed and its normal vector is compared to the twelve standard directions to interpolate the surface temperature and add in the reflected infrared energy.

Atmospheric Attenuation. Before the gray level is assigned, the effect that atmospheric attenuation of surface emittance has on the gray level is calculated. Atmospheric attenuation depends on the field-measurable parameters of temperature, relative humidity, range, and visibility. A scaled-down version of the LOWTRAN 4 atmospheric transmission model developed at the Air Force Geophysics Laboratories is used. The maritime aerosol model and a horizontal transmission path are assumed. The gray level assigned is based on the attenuated emittance "seen" by the sensor.

Frame Generation

User-Selected Parameters. A number of parameters/options are available to the user for generating imagery in many formats and for many conditions. The user can select:

1. Data base and associated level of detail
2. Eye position in x,y,z scene coordinates
3. Aim point (i.e., image center) in x,y,z scene coordinates
4. Frame size--number of rasters, number of pixels per raster
5. Field of view--horizontal and vertical in degrees
6. Rotation in degrees
7. Special features for earth curvature, smooth-surface shading, and edge smoothing.

The frame size and field of view chosen dictate the sampling resolution per pixel. Field of view is also inversely proportional to magnification. For a given frame size, objects remaining in the field of view will subtend more pixels as the selected field of view decreases. For the periscope modeled, magnification of 1X corresponds to a 48° field of view, 6X to 8°.

Geometric Processing. Geometric processing is a preprocessing step to raster processing. It comprises all operations by which the potentially viewable surfaces are ordered and oriented relative to the eye.

As explained earlier (see Geometric Data Base Modeling), objects are ordered through tree traversal, during which sibling nodes are compared via separating planes. For a pairwise comparison, the node on the same side of the separating plane as the eye has priority over the other node. Since each object's surfaces are presorted, all surfaces are ordered when all objects have been ordered.

All surfaces are then clipped by the planes of the field of view using a reentrant polygon clipper.[4] The clipping process can be rapidly accelerated by determining whether the bounding boxes of the objects are totally inside or outside the field of view. If the effects of earth curvature are to be simulated (so that the ship drops off the horizon), the surfaces are also clipped against an ocean plane.

Finally, all remaining surfaces are translated and rotated relative to the eye coordinate system, in which the origin is at the eye and one of the major axes lies along the line of sight. The resulting edges are stored in a general edge list for raster processing.

Note that the surface vertices are not transformed to 2-D screen coordinates (normally a perspective transformation) at this stage. Instead, the transformation is performed during raster processing because a nonlinear projection such as a cylindrical or spherical projection[5] requires the original 3-D information. For the CIG system discussed in this paper, only the perspective projection is used.

Raster Processing. The list-directed priority scheme considerably simplifies raster processing in that no further testing is needed to determine which surfaces may mask other surfaces. Frame generation proceeds one raster at a time, top to bottom. As each raster is processed, the points at which all edges enter and exit and raster are maintained. The edges are then projected onto the screen.

Each raster is generated pixel by pixel, left to right. For each pixel processed, a stack is maintained and updated to indicate which surfaces in the pixel, delimited by their edges in the raster, are potentially visible. If no edges of the top surface in the stack lie in the pixel, the pixel is completely shaded by that surface. If the surface has a uniform shade, the predicted surface tone is used. If the surface shade varies between vertices, Gouraud shading[6] is applied.

Whenever edges are visible in the pixel, aliasing effects are reduced by an edge-smoothing technique which assigns a shade to the pixel based on the relative amount of area of each surface in the pixel. The area of each surface is calculated by a simple summation of triangles and rectangles outlined by the surface edges and borders of the pixel. This technique does not reduce aliasing effects as completely as Catmull's integrator algorithm[7] because it only estimates portions of any surface other than the top surface that are actually visible in the pixel. However, this approach requires less computation and significantly reduces the stairstep effect observed in raster graphics.

SHIP MODELS AND IMAGERY

The major cues for recognizing ships are the edge content and distinguishable shapes produced by contrasts in gray shades. Although texture is an important recognition cue in many simulations, it is of little importance for ships because they usually have dull gray or black overcoats of paint. The ship's silhouette provides the primary sources of contrast; however, certain areas of the ship provide other contrast sources (e.g., hull concavities and "hot spots" in infrared simulations).

The overriding goal of data base modeling was to produce the ship's general outline along with those portions of the superstructure which add detail to the outline. In addition, we modeled major objects which blended in with the rest of the ship from a broadside view but were prominent from a quartering aspect. In line with the hierarchy illustrated in Figure 2, data bases were developed by dividing each model into its major components and then subdividing until all objects of interest were defined and their surfaces could be listed by inherent masking priority.

Emphasis was also placed on modeling objects so as not to display unwanted edges, such as those generated by planar patches used to simulate curved surfaces. Smooth shading was attempted on selected portions of the ship to demonstrate its effects.

Data bases for each ship range in level of detail from approximately 500 to 4000 potentially visible edges. Data bases of high level of detail were developed first, followed by data bases of progressively lower levels. This approach was based on the assumption that it is easier to remove than to add detail. The three methods implemented to achieve the lower levels of detail were 1) planar patches were reduced in number and increased in size so that less or no curved surfaces were simulated, 2) objects or surface detail not contributing to the broadside silhouette or other major areas of contrast were eliminated, and 3) small objects were either removed or grouped to form major, simplified objects.

Figures 5 through 8 and Figures 9 and 10 are, respectively, visual images of the Kashin and Kiev high-level-of-detail models. These images illustrate the complete 3-D nature of the models. The Kashin model consists of 63 objects and 1150 surfaces; the Kiev model of 103 objects and 2300 surfaces. The images were generated on the VAX-11/780 computer at the Experimental Computer Simulation Laboratory of NTEC, and were photographed with a Dicomed D-47 image recorder. Options for edge smoothing and smooth shading were used.

For the Kashin, the midsection was emphasized because of its prominent structures such as the smokestack and radar towers. In Figure 8, the main radar tower's complex tubular structure illustrates the detail that can be portrayed. In addition, the hull was emphasized because of its distinctive shape and the line that runs along much of it, both of which are significant recognition cues. (The hulls of U.S. destroyers are flat from bridge to stern.)

For the Kiev,* importance was placed on modeling the dominant superstructure and the irregular, asymmetric hull with its concave sections. As was done for the Kashin, radar antennas were modeled because they are distinguishable features. The top-sail radar (dark object above the bridge) was modeled with especially high detail using a gridwork of tubes.

RECOMMENDATIONS

Whereas the CIG software at TSC needed only minor modifications to handle the large geometric data bases and the maritime environment, the geometric data base modeling required much effort because of the manual, labor-intensive techniques used. Ship measurements were made from line drawings and entered into the geometric data base via keyboard. Most of the validation consisted of visually inspecting sample computer-generated imagery. Tasks such as ordering surfaces by their inherent masking priority and choosing the detail to be used for modeling objects were generally performed by the modeler.

An interactive data base editor with the proper graphics hardware could significantly reduce the throughput time for developing and validating a geometric data base. Data could be entered via digitizer tablet or keyboard and displayed on a vector graphics terminal. With cursor control and a convenient set of command capabilities, the modeler could manipulate the data on a vertex, surface, or node level to achieve any desired geometric configuration. Operations useful to the modeler could range in complexity from simple translation and rotation routines to automatic surface ordering within an object, and even to automatic object subdivision to resolve surface masking-priority conflicts.

ACKNOWLEDGMENTS

Acknowledgments are extended to Warren Stone of TSC for his work on data base modeling, Jeri O'Donnell of TSC for her editorial work on this paper, and Dennis Mistoler of NTEC for generating the ship imagery.

REFERENCES

1. Sutherland, I. E., R. F. Sproull and R. A. Schumacker, "A Characterization of Ten Hidden-Surface Algorithms," Computing Surveys, Vol. 6, No. 1, March 1974, pp. 1-53.
2. Schumacker, R. A., et al., Study for Applying Computer-Generated Images to Visual Simulations, AFHRL-TR-69-14, U.S. Air Force Human Resources Laboratory, September 1969.
3. Mitchell, R. J., "Surface-Shadow Generation for the Preparation of Synthetic Imagery," Digital Processing of Aerial Images, SPIE Proc. Vol. 186, May 1979, pp. 249-253.
4. Sutherland, I. E., and G. W. Hodgman, "Reentrant Polygon Clipping," Comm. of the ACM, Vol. 17, No. 1, January 1974, pp. 32-42.
5. Sutty, G. J., "Nonplanar Projections in Image Simulations," Image Understanding II, SPIE Proc. Vol. 205, August 1979, pp. 153-159.
6. Gouraud, H., Computer Display on Curved Surfaces, UTEC-CSC-71-113, University of Utah, June 1971.
7. Catmull, E., "A Hidden-Surface Algorithm with Anti-Aliasing," SIGGRAPH '78 Proc., Vol. 12, No. 3, August 1978, pp. 6-11.

---

*Although the figures show the Kiev as equal in size to the Kashin, the Kiev model is actually twice as large. The field of view for the Kiev images was halved.

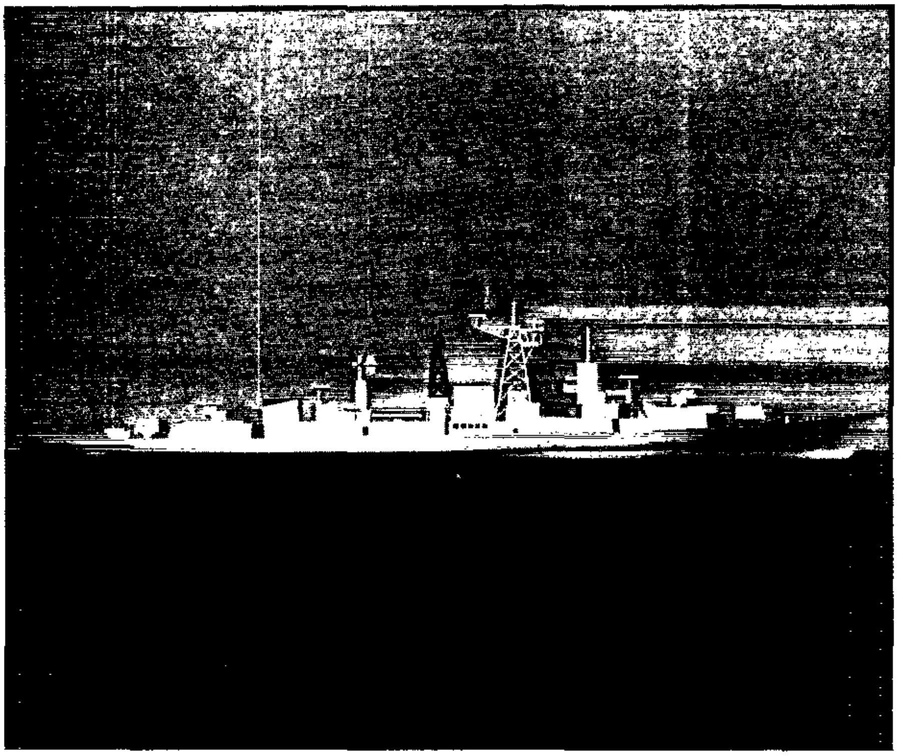


Figure 5. Broadside View of Kashin

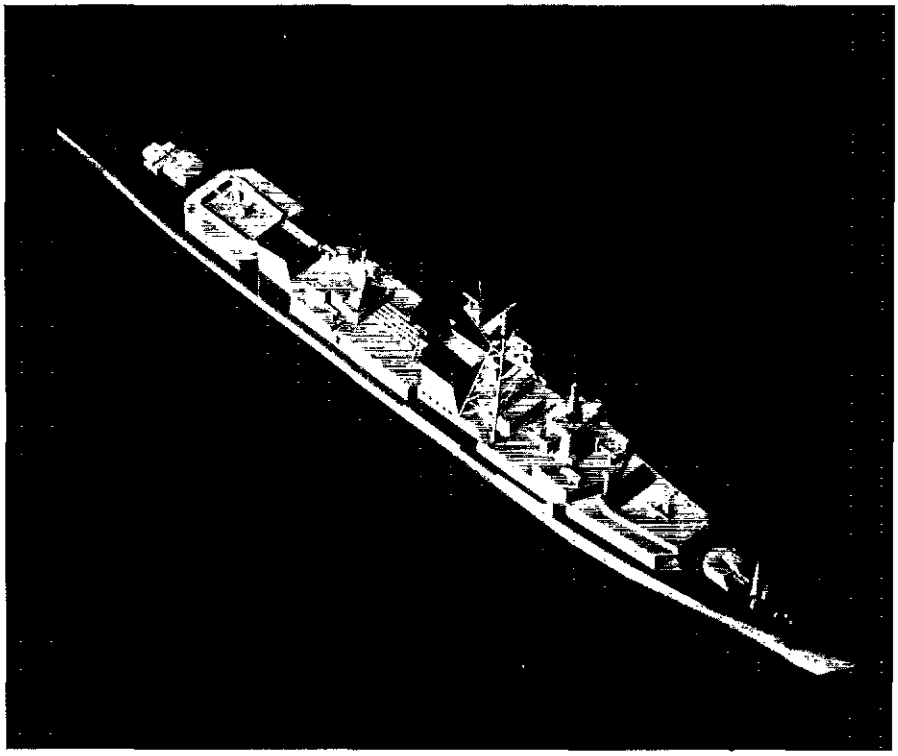


Figure 6. Aerial View of Kashin

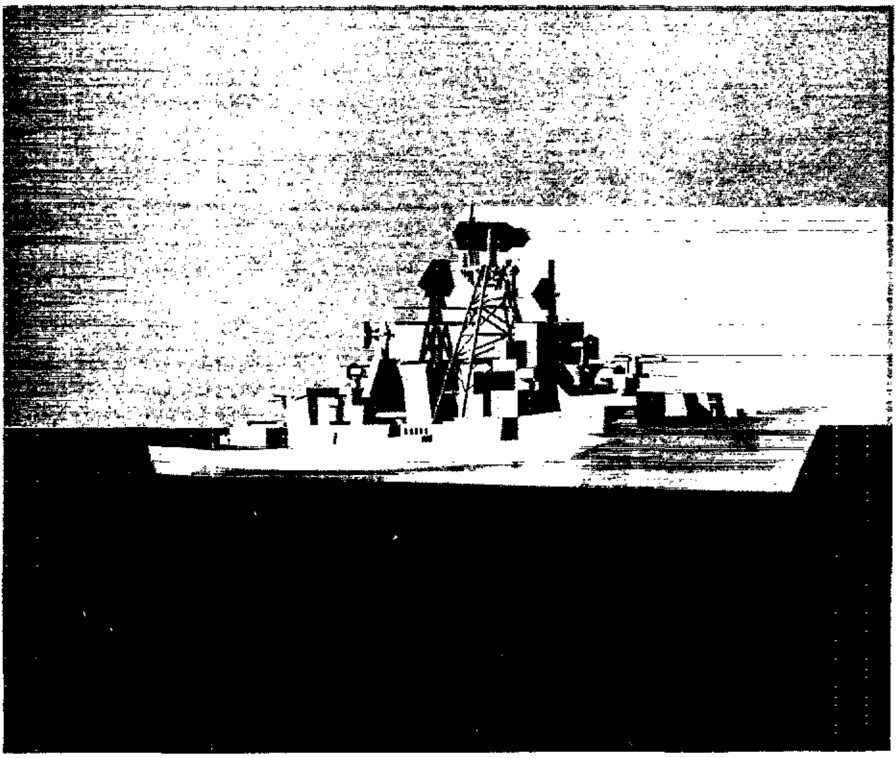


Figure 7. Quartering View of Kashin

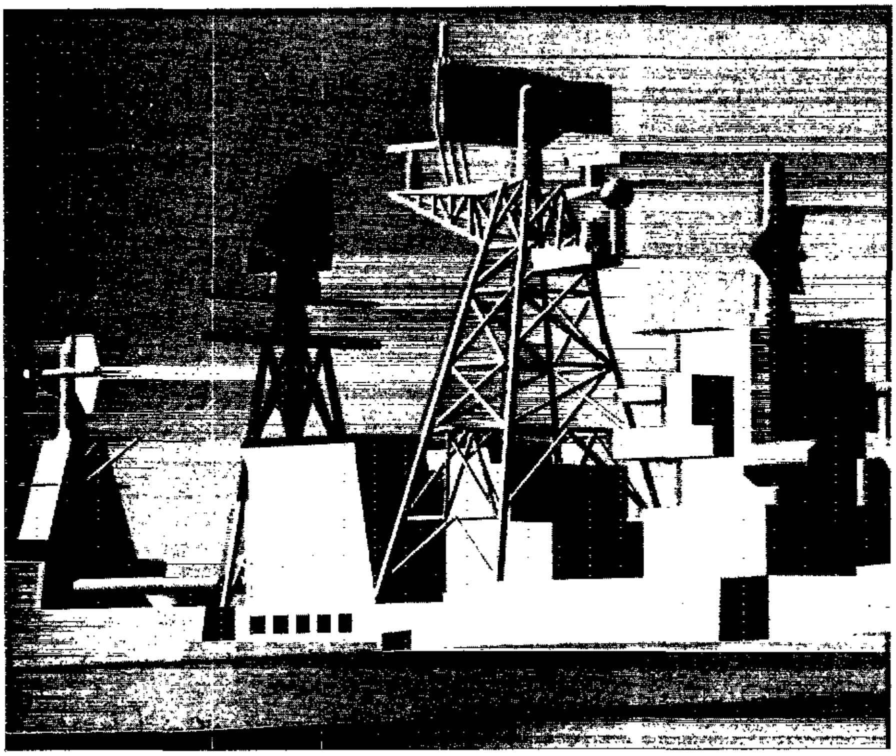


Figure 8. Close-Up of Kashin Midsection

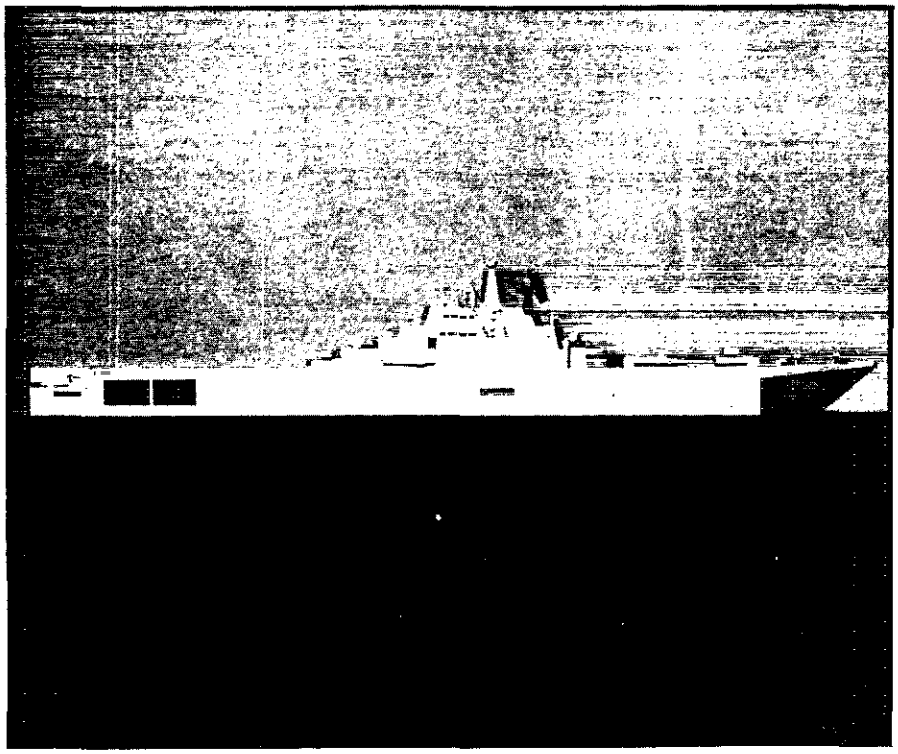


Figure 9. Broadside View of Kiev

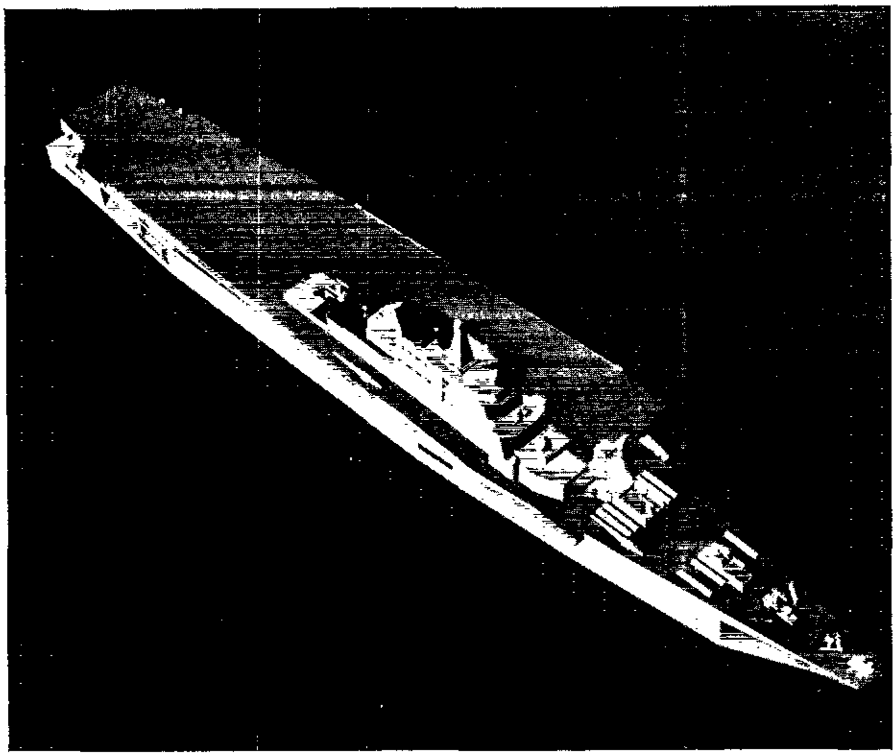


Figure 10. Aerial View of Kiev

ABOUT THE AUTHORS

Richard J. Mitchell is a Member of the Research Staff at Technology Service Corporation. His major work has been on general and specialized computer simulation techniques for training and correlation guidance systems involving many sensor types and formats. Mr. Mitchell has a B.S. in Engineering Science and an M.S. in Electrical Engineering, both from the University of Tennessee.

Anthony J. Stenger is the Manager of the Simulation Systems Department at Technology Service Corporation. His department is responsible for modeling and simulating sensor systems, their targets, and the environment. Mr. Stenger has a B.S. from the University of Detroit and an M.S. from the University of Southern California, both in Electrical Engineering.

John L. Booker is a Project Engineer in Research and Technology at the Naval Training Equipment Center. His research includes computer-generated displays both in CIG and interactive man-computer interface, real-time computer software, and computer systems architecture for training simulators. He formerly worked on logic and computer interface design for the Martin Company. Mr. Booker has an A.B. in Journalism from the University of North Carolina, a B.S. in Electrical Engineering from North Carolina State University, and a Master of Engineering degree from the University of Florida.