AD-A101 730 SCRIPPS INSTITUTION OF OCEANOGRAPHY LA JOLLA CA VISA--ETC F/G 9/2
OBJECT SIMULATION.(U)
MAY 81 B L MCGLAMERY, B F FAHY F30502-78-C-0246
UNCLASSIFIED RADC-TR-81-64 NL

1 OF 1
AD
A101 730

END
DATE
FILMED
8-81
DTIC

RADC-TR-81-64
Interim Report
May 1981

OBJECT SIMULATION

University of California

Sponsored by
Defense Advanced Research Projects Agency (DoD)
ARPA Order No. 2646

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-81-64 has been reviewed and is approved for publication.

APPROVED:

JOHN J. VASSELLI
Project Engineer

APPROVED:

FRANK J. REHM
Technical Director
Surveillance Division

FOR THE COMMANDER:

JOHN P. HUSS
Acting Chief, Plans Office

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER RADC-TR-81-64	2. GOVT ACCESSION NO. AD-A101 730
	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) OBJECT SIMULATION	5. TYPE OF REPORT & PERIOD COVERED Interim Report 15 Mar 80 - 15 Sep 80
	6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) B. L. McGlamery B. F. Fahy	8. CONTRACT OR GRANT NUMBER(s) F30602-78-C-0246
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of California/Visibility Laboratory Scripps Institute of Oceanography San Diego CA 92120	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62301E 26460409
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Blvd Arlington VA 22209	12. REPORT DATE May 1981
	13. NUMBER OF PAGES 16
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Rome Air Development Center (OCSE) Griffiss AFB NY 13441	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: John J. Vasselli (OCSE)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Object Simulation
Three-Dimensional Modeling
Computer Graphics

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes a three-dimensional object simulation program which has been developed. Input to the program is a description of the object in terms of polygons, and light source-object-observer geometry. Output from the program is a two dimensional array of numbers representing the brightness values of the object as would be seen by the observer. Shadows are included in the output image. Examples are shown. The procedure for simulation of an optical system using the simulated images

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) as input is also given, along with examples.

SUMMARY

This report describes a three-dimensional object simulation program which has been developed. Input to the program is a description of the object in terms of polygons, and light source-object-observer geometry. Output from the program is a two dimensional array of numbers representing the brightness values of the object as would be seen by the observer. Shadows are included in the output image. Examples are shown. The procedure for simulation of an optical system using the simulated images as input is also given, along with examples.

OBJECT SIMULATIONS

Introduction

A computer program has been developed at the Visibility Laboratory for simulating three-dimensional objects. The program produces a two dimensional array of data values which represents the object's relative brightness or radiance values for a given light source-object-observer viewing geometry. An important and unique feature of the program is that shadows are realistically included in the image. This program has been used to produce images for system simulation and mensuration studies. This report briefly describes the method used in the program and gives examples of both object simulation and system simulation.

Description of Program

The ability to computer simulate three-dimensional objects is not new[1,2]. The distinguishing feature of the program being presented here, called NMMV, is its ability to realistically handle shadows for an arbitrarily placed light source, and its ability to separate the hidden surface algorithm from the picture-making algorithm.

In the previous simulation programs used at the Visibility Laboratory, the light source was assumed to be coincident with the observer. Further, one had to select the picture parameters, such as resolution and region of the object to be displayed in advance, since the Watkins hidden surface algorithm involves a scan line-by-scan line approach. The NMMV method implements the hidden surface algorithm first and uses a separate algorithm to generate the pictures afterward.

The NMMV method is in principle very simple. Objects are defined in a local 3-D coordinate system. The object must be broken into planar polygons of 16 sides or less. For curved objects, the finer the piecing the better the resulting approximation. The polygons are defined by specifying the position of their vertices (referred henceforth as nodes) in 3-space, along with specifying a clockwise connection pattern, which will be made clear by a simple example.

See Fig. 1a. The polygon shown is actually a square. If we are to stand upon the first node and walk along the outside of the polygon in a clockwise fashion (from the observers point of view) we would encounter the nodes of the polygon in the order 1, 4, 3 , 2. Thus we would define the polygon as the connection of nodes 1, 2, 3, and 4 in the order of 1, 4, 3 , 2.

Definition of the polygons with proper clockwise orientation is mandatory. As a rule, if one wishes a polygon to be visible, it must be seen to be a clockwise polygon. Note that an observer from the opposite position in the example above would see the polygon as counterclockwise, and hence for that observer the polygon would appear invisible, by definition. One wishing to define the very thin rectangular panel on Fig. 1b who wants the panel to be visible from all angles would have to define both sides of the panel to be two distinct polygons, e.g., polygon

A = 1, 4, 3, 2 and polygon B = 1, 2, 3, 4. The program interprets polygons which are oriented counterclockwise to be pointing away from the observer. These polygons are treated as if they were invisible to the observer because it is assumed that in simulating solid objects the observer always sees the face of a panel and never the back.

The basic algorithm for taking a polygon-approximated image and determining its 2-D projection with shading is as follows:

(1) The polygon approximation is converted to the coordinate system of the light source. This is accomplished by having the user input azimuth and zenith position angles of the object and light source with respect to an arbitrary coordinate system centered at the observer. The program then performs the mathematical calculations required to express each node of the object in the coordinate space of the light source.

(2) Taking each polygon in turn, the program decides if the polygon is visible, i.e., still clockwise oriented. If so, the program puts it in a visible list and will process it further. Otherwise, it places the polygon in a hole list, and no further processing will be done on it until step

(3). Of the polygons in the visible list, some will be in front of others and either partially or completely prevent light from reaching the more distant polygons. An algorithm is employed which for any two polygons will decide which polygon is in front (if either) and which casts a shadow on the other (if such is the case). The basic method simply involves projecting both polygons on a common plane and testing all segments of one against the other to determine if there are any non-trivial intersections.

If the two polygons do not intersect, both remain in the visible list and testing is carried out on the next pair of polygons (which pair may involve one of the polygons from the previous pair). If an intersection is found, the polygon upon which the shadow is cast is divided into visible and shadowed components. That is, the polygon is subdivided into two or more sub-polygons, each of which have the property of being either completely visible (as far as the shadowing polygon is concerned) or completely blocked.

The sub-polygons which are shadowed are transferred to the hole-list and will not be processed further until step three. The visible sub-polygons are returned to the visible list and are thenceforth treated as any other visible polygon in that list. Thus, they may be tested again against other polygons, though they will not be tested against their siblings nor will they be tested against any polygons their parent polygon was already tested against.

In this manner, all visible polygons are tested against each other until the visible polygon list is purified and there remains in the list only polygons which are completely visible to the light source. Conversely, those polygons in the hole list are either completely shadowed or else were oriented in the wrong direction and would have been shadowed if they had been put to test anyway. This completes step two.

(3) Now the polygons are all put together in a new list, reforming the object more or less. As each polygon is put in the new list it is tagged as having been either visible or not-visible (shadowed). The object is then

redefined in the coordinate system of the observer, by suitable coordinate transformation similar to those used in step (1).

(4) Polygons are now tested to see if they are clockwise. If they are not, they are pointing the wrong way and thus must be blocked by some polygon(s) pointing the correct way. Thus, the observer cannot possibly see them and we put them in a new hole list which is the same as throwing them away since we will never be doing any more processing on them. Those polygons which have a chance of being visible are put in the visible list and will be processed further in step (4). It is important to remember that of these visible polygons, some will have been tagged as lighted and some as unlighted since some may have been invisible to the light source but are visible to the observer. Some will have been visible to the light source but will not be seen by the observer, and some will be visible to both.

(5) It is the purpose of step five to carry out this final partitioning. The method is

exactly analogous to that used in step (2) above, so only the results will be discussed here. That is, step (5) results in a list of pure visible polygons in the sense that all polygons remaining in the visible list after step four are completely visible to the observer. Further, all polygons which are in the hole list are blocked from the view of the observer. The important difference between steps (5) and (2) is that the visible and lighted list and a visible but shadowed list such that all members of the former list are completely visible and completely lighted, and all members of the latter list are completely visible but are not illuminated by the source.

(6) Finally, the polygons are shaded according to their reflectivities and angles between the light source, object and observer. That is, each polygon is assigned an intensity value between one and zero. Shadowed polygons may receive non-zero values if the user has selected an option of diffuse lighting. This allows soft shadows to be produced.

(7) A separate program can then be used which fills in an array of arbitrary size by assigning each pixel in the array the intensity of the polygon (if such exists) which occupies that position in the projected plane perpendicular to the viewing angle of the observer. Thus, once the basic algorithm has been run, the user may choose any size for display, whereas in previous programs, the size had to be fixed in the beginning.

Examples of Object Simulation

Figure 2 shows examples of a simulated object. The light source is behind the observer and below the horizon. The object is shown at azimuth angles of -20, -10, 0, 10, and 20 degrees with respect to a plane through the observer and the sun. Zenith angle of the object is approximately 30°. The object is rotating approximately 20° from one position to the next in a direction the same as the apparent rotation due to azimuth angle. Thus the total apparent rotation to the observer is about 30° from one position to the next. The object is shown with both soft and hard shadows. Hard shadows are those areas that receive no direct illumination from the source. Soft shadows are the hard shadow areas filled in with low brightness values, still proportional to the reflectance of the panels in the shadowed areas. This serves to show the true geometrical shape of the object. It could also simulate in a crude way illumination of shadowed areas by reflected light within the object or from another object.

System Simulation

The object modeling program provides the input to a system simulator which then applies the characteristics of a given sensor system. One such system which has been simulated is a system employing adaptive optics for atmospheric turbulence compensation. (Ref. 3) The following steps are involved in the simulation of a long time-exposure image.

(1) Fast Fourier transform the object radiance map.

(2) Compute the telescope diffraction modulation transfer function.

(3) Compute the modulation transfer function associated with the residual wavefront errors of the adaptive optics system.

(4) Compute the modulation transfer function associated with miscellaneous errors.

(5) Multiply the object Fourier transform by all these transfer functions, and inverse transform to get the degraded image, without sensor effects.

(6) Scale the degraded image spatially and in intensity so that the value at each pixel corresponds to the mean number of photoelectrons per sensor system resolution element per sample time.

(7) Apply a Poisson noise process to the array to simulate the noise effects of the photoelectron process.

(8) Apply resolution degradation of the sensor and its associated electronics by Fourier transform techniques.

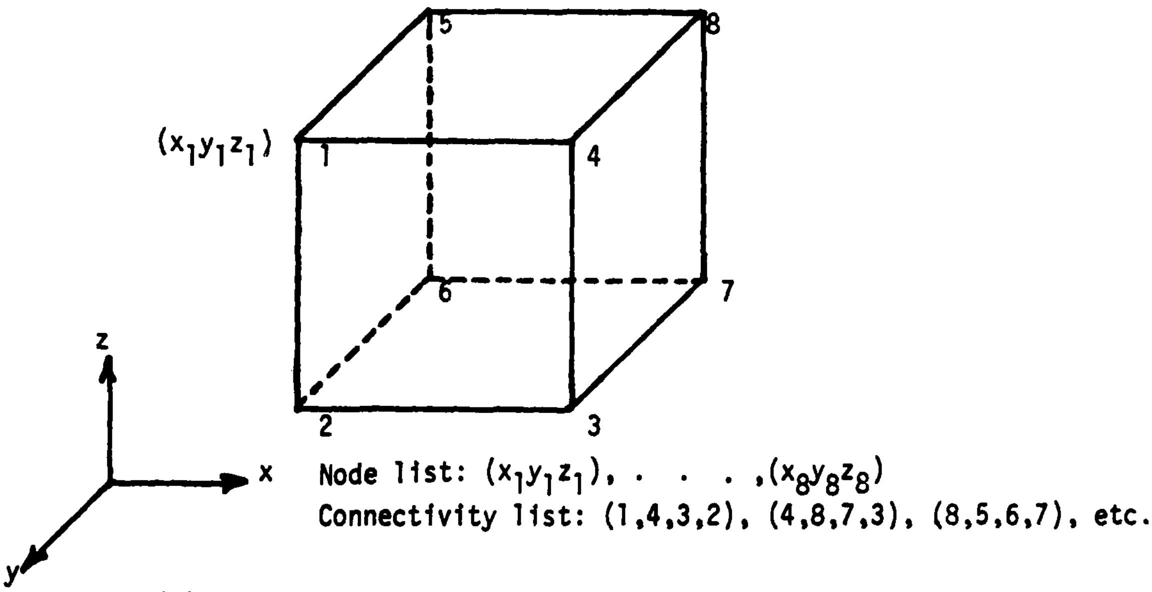
(9) Add additional noise to simulate preamp and other system noise.

(10) Discretize the image data to specific levels to simulate analog to digital conversion. This simulates the final recorded image.

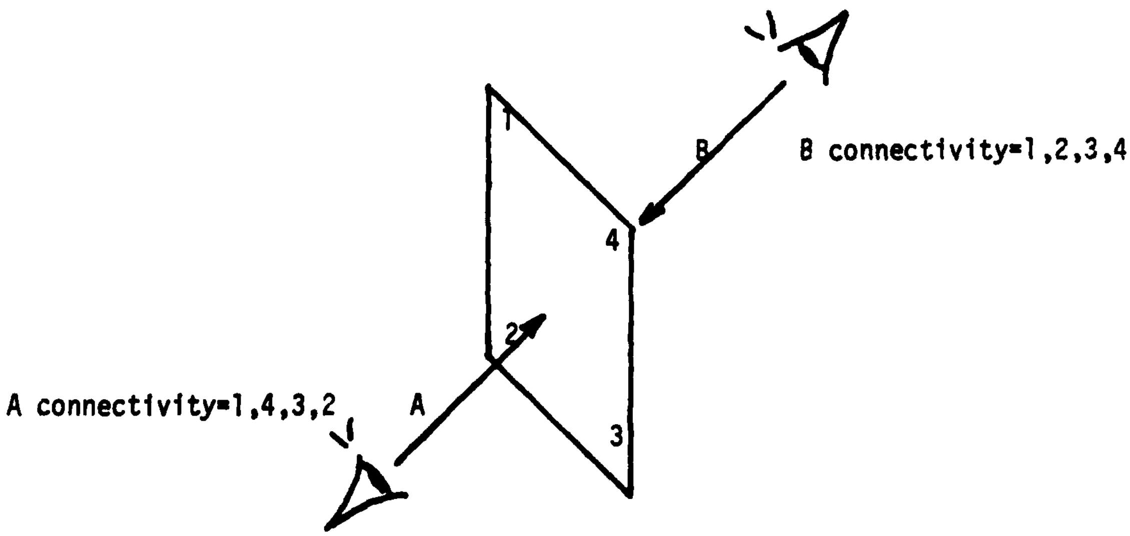
Figure 3 shows the results of a system simulation process for the same images as shown in Fig. 2.

Discussion

The ability to simulate images of three-dimensional objects has been found to be a very useful tool for providing precisely defined and precisely variable images for use in system and mensuration studies. Another application of object simulation which has yet to be fully explored is its use in image interpretation. (Ref. 4) Given the degraded image of a real object, the object and its image would be simulated. The simulation would be based on all previous knowledge of the object, the known lighting and viewing conditions, the characteristics of the imaging system, and information extracted from the actual degraded image. A comparison would then be made between the actual and simulated degraded image. Differences between these images which exceed the noise level of the real image would then indicate changes to be made to the assumed object. An iterative process would allow the true object to be constructed within the limits imposed by resolution and noise in the real image.



(a). Representation of an object by polygons.



(b). Connectivity convention for visible surfaces.

FIGURE 1

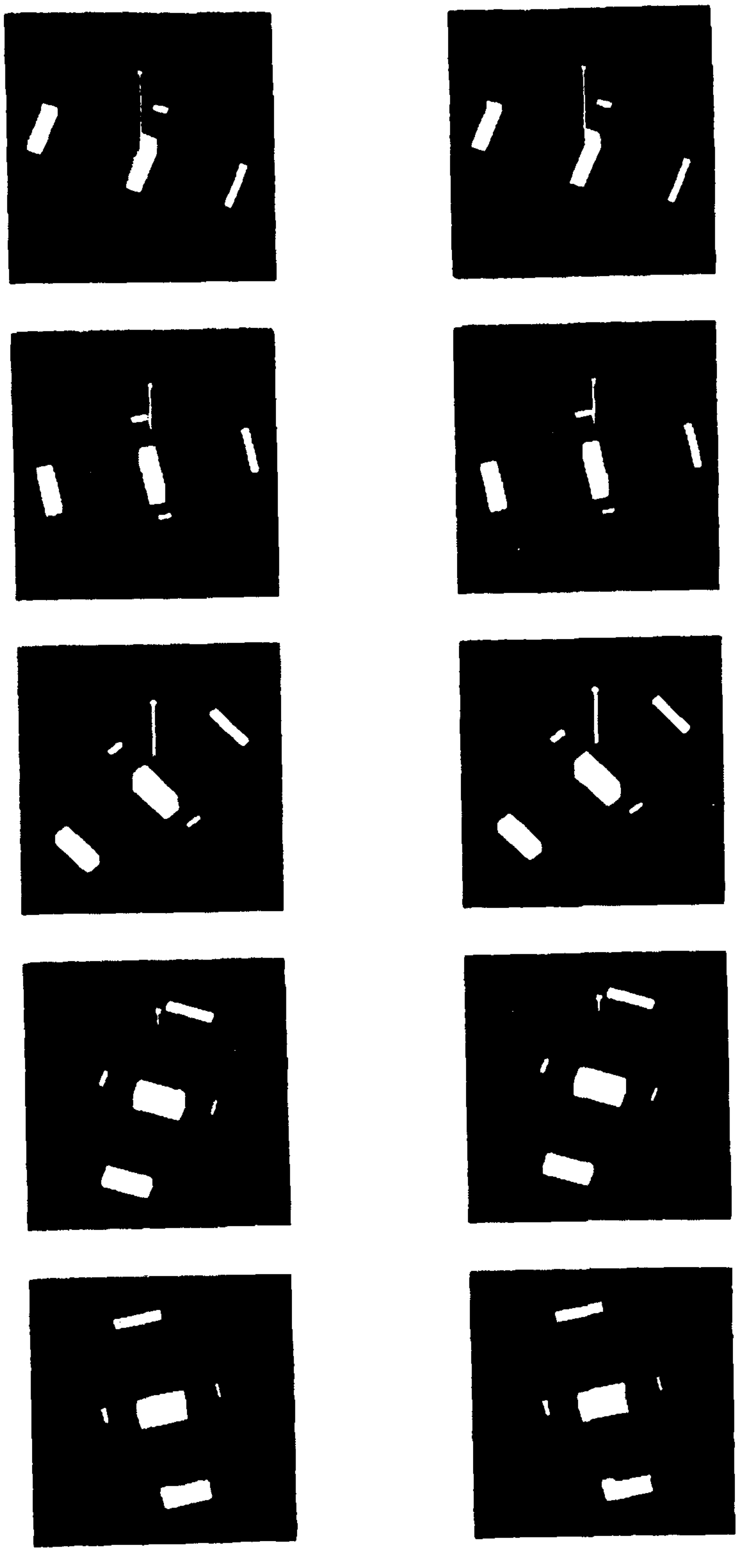


Fig. 2 Computer generated objects with shadows. Sun is behind observer and below the horizon.
Left column: 'soft' shadows. Right column: 'hard' shadows.

Fig. 3. 'Hard' shadow images of Fig. 2 with atmospheric and sensor effects.

REFERENCES

1. G. S. Watkins, "A Real Time Visible Surface Algorithm", UTEC-CSc-70-101, Computer Science, University of Utah, Salt Lake City, Utah 84112, June 1970.

2. Numerous papers from Computer Graphics, Quarterly Report of SIGGRAPH-ACM, Vol. 11, Number 2, Summer 1977.

3. B. L. McGlamery, "A Summary of Simulation Techniques for Compensated Imaging Systems", RADC report, RADC/OCSE, Griffiss AFB, NY 13441, Aug. 1980.

4. B. L. McGlamery, "Preliminary Requirements for a CIS Image Processing Facility", RADC report, RADC/OCSE, Griffiss AFB, NY 13441.

addresses	number of copies
John J. Vasselli RADC/OCSE	10
RADC/TSLD GRIFFISS AFB NY 13441	1
RADC/DAP GRIFFISS AFB NY 13441	2
ADMINISTRATOR DEF TECH INF CTR ATTN: DTIC-DDA CAMERON STA BG 5 ALEXANDRIA VA 22314	12
Visibility Laboratory Scripps Institute of Oceanography San Diego, California 92120	5
Space Division P.O. Box 92960 Worldways Postal Center Los Angeles, CA 90009 Attn: YNCS/Capt Jackson	1
HQ ADCOM Peterson AFB, CO 80914 Attn: INAY	1
HQ ADCOM Peterson AFB, CO 80914 Attn: XPDO	1

Itek Corp. 3
10 Maguire Rd.
Lexington, MA 02173
Attn: Mr. Paul Mailhot

FTD 2
Wright Patterson AFB, OH 45433
Attn: ENHP/Mr. C. Chapin

FTD 2
Wright Patterson AFB, OH 45433
Attn: Mr. A. T. Larson

RADC/OL-AB 1
c/o AVCO Everett Res Lab
P.O. Box 261
Puunene, Maui, Hawaii 96784
Attn: Capt Gary Dahlen

Aerospace Corp. 1
P.O. Box 92957
Los Angeles, CA 90009
Attn: Dr. Ira Gura

Aerospace Corp. 1
P.O. Box 92957
Los Angeles, CA 90009
Attn: Dr. Turner

AVCO Everett 2
P.O. Box 261
Puunene, HI 96784
Attn: Program Manager

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

DAT

ILM