

NASA Technical Memorandum 88324

(NASA-TM-88324) A COMPUTATIONAL METHOD TO
MODEL RADAR RETURN RANGE IN A POLYGONALLY
BASED, COMPUTER-GENERATED-IMAGERY SIMULATION
(NASA) 22 p CSCI 08B

N87-12967

Unclas

G3/43 44662

A Computational Method to Model Radar Return Range in a Polygonally Based, Computer- Generated-Imagery Simulation

F.J. Moran and J.D. Phillips

July 1986

A Computational Method to Model Radar Return Range in a Polygonally Based, Computer- Generated-Imagery Simulation

F. J. Moran,
J. D. Phillips, Ames Research Center, Moffett Field, California

July 1986



National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

SYMBOLS

b	two-dimensional array designating the intercept of a line segment of a mountain for each mountain and line segment of that mountain (projection in the x - y plane)
b_R	intercept of the radar line-of-sight (LOS) equation in x - y plane
C, c	two-dimensional array designating the intercept of a line segment of a mountain for each mountain and line segment of that mountain (projection in x - z plane)
D_{LOS}	distance from x_c, y_c to radar LOS
D_{NLOS}	distance from a point to the line which is normal to the radar LOS and passes through the point defined by the vehicle position
e	end vector
H_s	height of the terrain above the CGI reference (CGI reference is $z = 0$)
L	one-dimensional matrix containing total number of line segments associated with each mountain
M	two-dimensional array designating the slope of a line segment of a mountain for each mountain and line segment of the mountain (projection in x - y plane)
M_{ij}	vector associated with mountain i and line segment j
M_R	slope of the radar LOS equation in the x - y plane
N, n	two-dimensional array designating the slope of a line segment of a mountain for each mountain and line segment of the mountain (projection in the x - z plane)
N_{LI}	peripheral line segments associated with index i stored in matrix N
s	start vector
V	vertex of a mountain
x_c, y_c	coordinates of center of circle being checked (major or minor circle)
x, y, z	coordinates of a point on the CGI data base
x_1, y_1, z_1	start points of a line segment on the CGI map

x_2, y_2, z_2 end points of a line segment on the CGI map

$X_{A/C}, Y_{A/C}$ aircraft x and y position in the CGI map

X_{CGI}, Y_{CGI} coordinates of the intersection formed by the vertical plane defined by the radar LOS and the lines defining mountains on the CGI data base

ψ_E, ψ_S two-dimensional matrix designating the angular bearing of a line segment of a mountain (projection in x-y plane) referenced to a horizontal line containing the start or end point of the line segment

ψ_L angular bearing of a line segment of a mountain; it is either ψ_S or ψ_E

ψ_R radar heading angle (in x-y plane)

SUMMARY

This paper describes a method for modeling a ground-mapping radar system for use in simulations where the terrain is in a polygonal form commonly used with computer generated imagery (CGI). The method employs a unique approach for rapidly rejecting polygons not visible to the radar to facilitate the real-time simulation of the radar return. This rapid rejection of the nonvisible polygons requires the precalculation and storage of a set of parameters that do not vary during the simulation. The calculation of a radar range as a function of the radar forward-looking angle to the CGI terrain is carried out only for the visible polygons. This method was used as part of a simulation for terrain-following helicopter operations on the vertical motion simulator at the NASA Ames Research Center. It proved to be an efficient means for returning real-time simulated radar range data.

INTRODUCTION

The increasing need of the military to fly low-level helicopter missions at night or in adverse weather has led to the development of terrain-following radar systems (ref. 1). These systems have been employed for use with the H-53 PAVE Low III helicopter (ref. 2) and are presently being considered for the HH-60 helicopter. Systems of this type are also being considered for the LHX helicopter (ref. 3). These types of rotorcraft systems when used at low altitude must rely on forward-looking radars, forward-looking infrared (FLIR) systems, and other devices which will allow them to avoid terrain and other obstacles in their path.

The presentation of the necessary information to the pilot and to an automatic control system in a terrain-following simulation requires an efficient algorithm for generating real-time, simulated, radar returns (refs. 4 and 5). The model described here was required in real time to return a range-altitude profile of approaching computer-generated imagery (CGI) terrain. This profile was then used for terrain-following guidance. The profile was updated every 0.2 sec along the helicopter flightpath vector.

COMPUTER-GENERATED-IMAGERY TERRAIN DATA BASE

Figure 1 is a projection onto the x-y plane of the CGI terrain data base which was developed for nap-of-the-earth (NOE) simulations. The CGI terrain consists of a detailed modeled area of approximately 9 km². The terrain is composed of hills or mountains which in turn are composed of polygonal surfaces. The altitudes

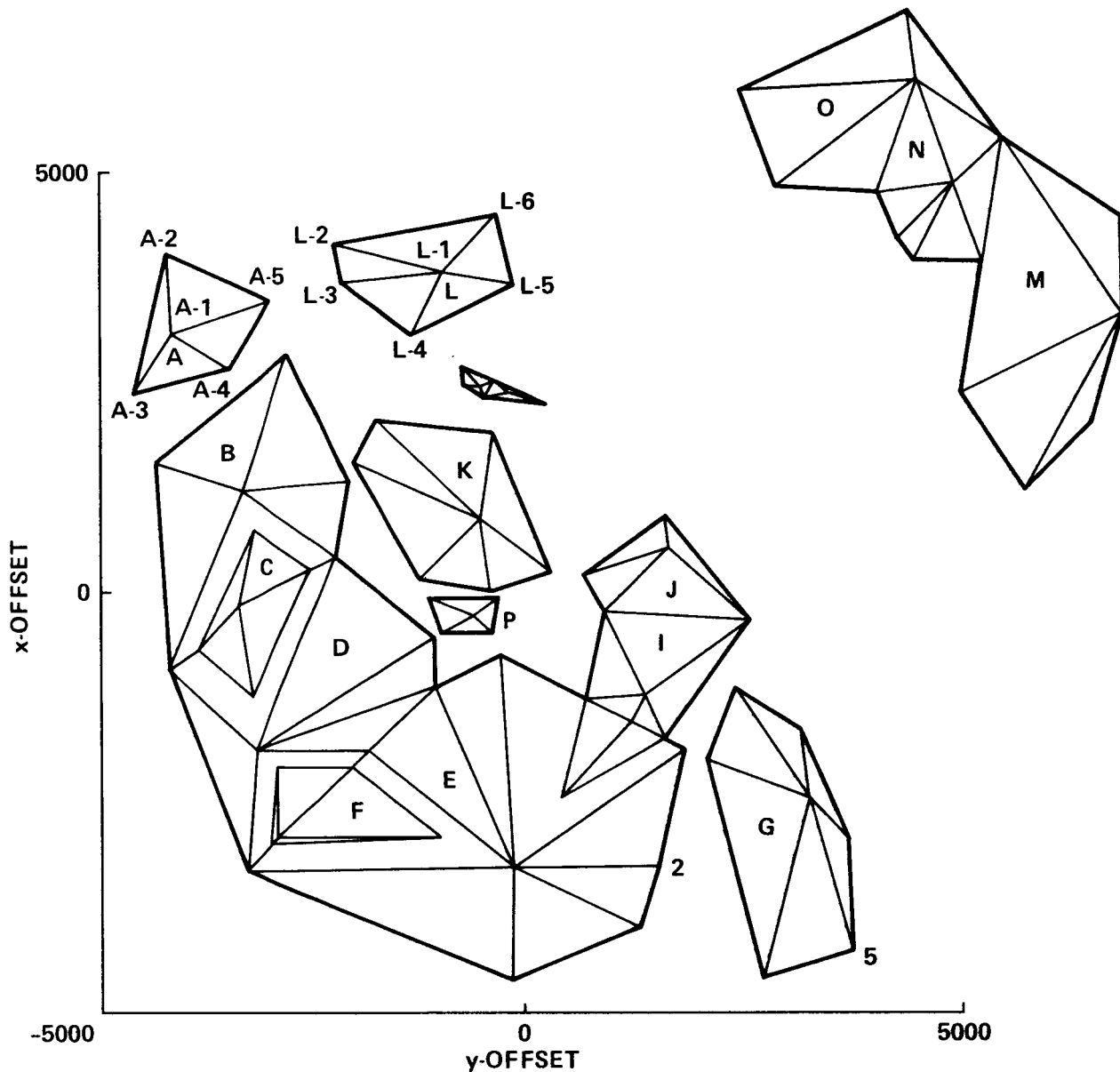


Figure 1.- CGI terrain data base.

of these mountains range from 0 to 1000 ft. Each mountain is identified by a letter and each end point of the line segments where the edges of the hill meet has associated x , y , and z values. The terrain also contains trees and buildings that did not produce returns in the radar model. The CGI system creates a scene by storing coordinates and numbers representing the light-intensity values of the simulated world. These values are retrieved for any viewing point in the data base and the visual scene is reconstructed. The scene details are then displayed to the pilot on a cathode ray tube (CRT). Smoothing and shading techniques are used to create a realistic image.

MODEL REQUIREMENTS

A real-time model of a forward-looking, ground-mapping radar is required which will produce a range-altitude profile of the CGI terrain along the aircraft's present flightpath vector (FPV). The range-altitude profile, to be discussed later, is defined as the profile determined by the intersection of the vertical plane containing the propagated FPV with the surface of the CGI terrain, given the simulated aircraft's x, y, and z position, and the coordinates of the CGI polygon vertex points. The simulation that required the radar model was for terrain following only. The guidance computations used the information from only one profile per computation cycle rather than from several profiles per computation as might be required in terrain avoidance.

METHOD

The development of the forward-looking radar model is a two-part procedure. (1) A set of parameters is calculated prior to the real-time simulation and stored. (2) The second part uses the algorithmic method for determining the altitude of the terrain above the CGI surface along the radar line of site (LOS). The altitude above the surface is defined as a function of the horizontal radar range in front of the simulated aircraft:

$$H_s = f(\text{horizontal radar range}) \quad (1)$$

where H_s is defined as the height of the terrain above the CGI reference value (the CGI reference is given by $Z = 0$).

The horizontal radar range is given by

$$\sqrt{(X_{A/C} - X_{CGI})^2 + (Y_{A/C} - Y_{CGI})^2} \quad (2)$$

where $Y_{A/C}$, $X_{A/C}$ is the aircraft's x and y position on the CGI map and Y_{CGI} , X_{CGI} are coordinates of the intersection formed by the vertical plane defined by the radar LOS and lines defining mountains on the CGI data base. An example of a range-altitude plane is shown in figure 2.

IMPLEMENTATION

The real-time flight computation initially determines which mountains and mountain edges are not traversed by the aircraft's projected horizontal FPV. A typical projected FPV might traverse 6 mountains with a total of 18 mountain edges out of a total of 38 mountains with about 450 edges. Thus the method chosen was one capable of rapid rejection of edges which were not traversed by the radar.

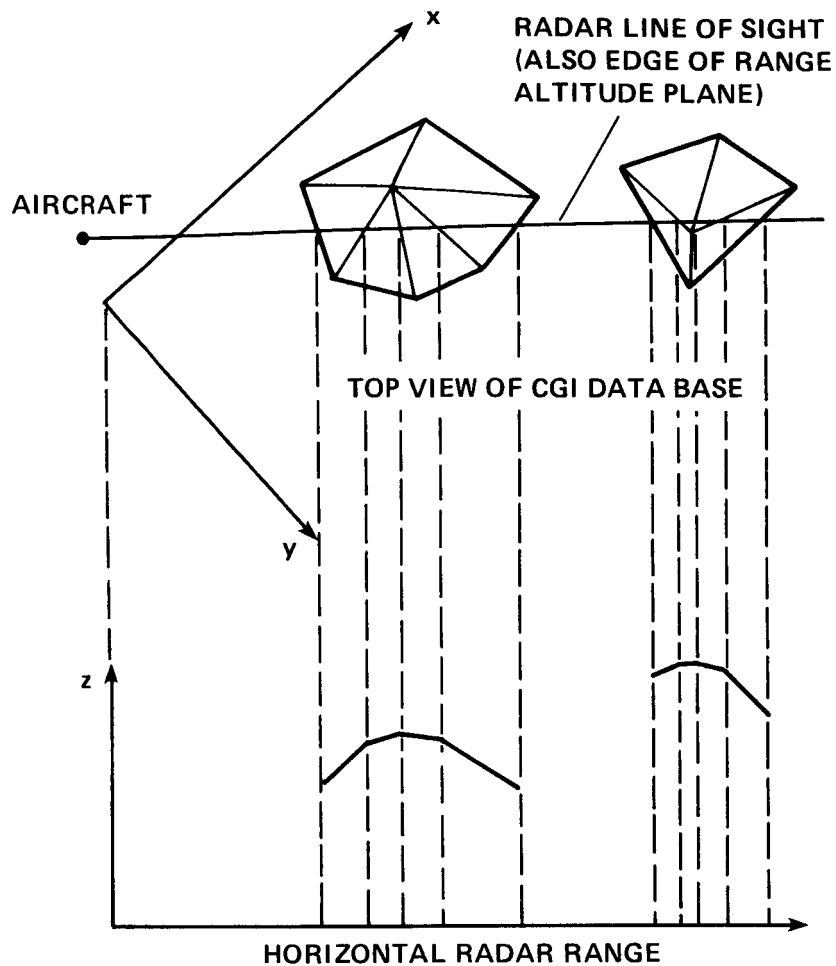


Figure 2.- Example of a range altitude plane.

Calculation and Storage of Nonvarying Parameters

As previously stated, the first part of the method requires the precomputation of certain nonvarying parameters (preflight calculations) which are stored prior to beginning the real-time computations. The procedure is as follows. All of the labeled vertices of the CGI system shown in figure 1 are stored. First, each mountain in figure 1, designated by a letter, i.e., A, B, C, and so on, where $A = 1$, $B = 2$, $C = 3$, and so on, is composed of line segments. V designates a vertex point of a mountain and i designates the mountain, i.e., $1 = A$, $2 = B$, etc. Next the start and end points of all the line segments are determined.

Consider figure 3 which shows mountain A in figure 1. Each line segment is numbered as shown. The number of peripheral line segments for each mountain is stored in L_i . Thus for mountain A in figure 3, $L_1 = 4$ where the index $i = 1$ stands for mountain A. Then the total number of line segments associated with each mountain is stored in L_i . Thus for figure 3, $L_1 = 8$ where the 1 stands for mountain A.

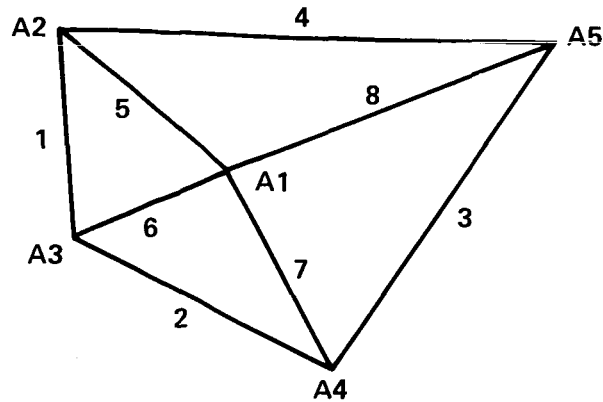


Figure 3.- Mountain A of CGI data base.

Next, the start and end points of the line segments are defined for the CGI data base. The start and end points are stored in two three-dimensional vectors, S and E, where the elements of S_{ijk} designate the start position of a line segment, i.e., the minimum value of the y coordinate, and the elements of E_{ijk} designate the end position of a line segment, i.e., the maximum value of the y coordinate for that line. For each matrix, i designates a mountain, as stated before, i.e., 1 = A, 2 = B; j designates a line segment of mountain i; and k, the start- or end-point coordinate of line j, that is:

- k = 1 -- x coordinate
- k = 2 -- y coordinate
- k = 3 -- z coordinate

Several more quantities are required during the simulated flight. These quantities are determined as follows. The equation of a line in the x-y plane in slope-intercept form is

$$x = my + b \tag{3}$$

The parameters m and b in equation (3) are required for all line segments on the CGI map. These parameters are easily evaluated using the start and end vectors, S and E. The slope array, m, and the intercept array, b, are computed and stored as two-dimensional vectors, M_{ij} and b_{ij} , where i designates the mountain and j designates the line segment of mountain i.

Other quantities required are the angular bearing of all line segments (projection in the x-y plane) as referenced to the start point of the line and also as referenced to the end point of the same line (180° difference). The angular bearing of a line segment as referenced to the start point of the line is simply the arc tangent of the M_{ij} vector. Thus, two vectors $\psi_{S_{ij}}$ and $\psi_{E_{ij}}$ (see fig. 4) are generated such that ψ_S designates the angular bearing of all line segments (with the projection in the x-y plane) referenced to a horizontal line containing the start point of the line segment and ψ_E designates the bearing of all line segments (with the projection in the x-y plane) referenced to a horizontal line containing

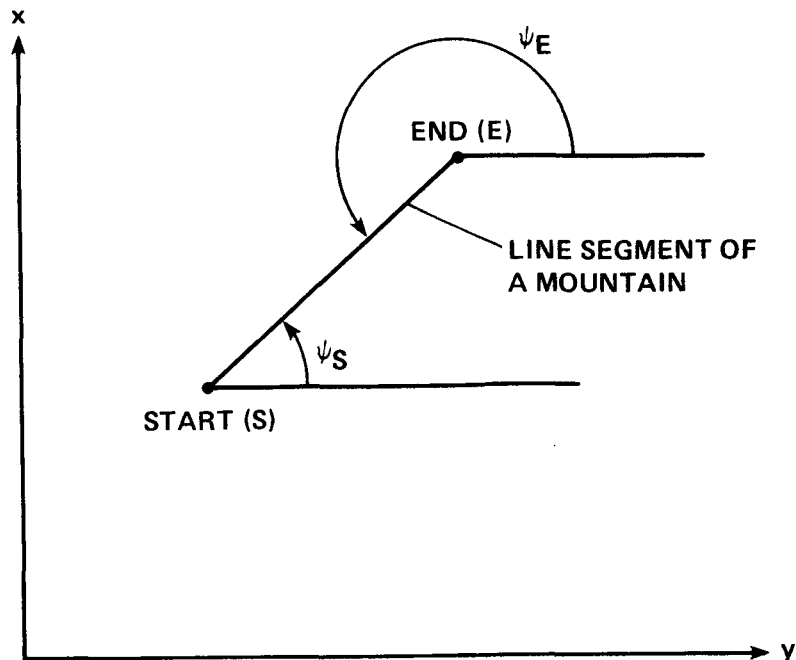


Figure 4.- Geometry defining the angular orientation of lines in the CGI data base.

the end point of the line. The index i designates the labeled mountain and the index j designates the line segment of mountain i . Similarly, the equation of a line in the x - z plane in slope-intercept form is

$$z = nx + c \quad (4)$$

where the parameters n and c in equation (4) are required for all line segments on the CGI map. Again these parameters are easily evaluated using the start and end vectors, S and E . As before, a slope array, n , and an intercept array, c , are computed and stored as two-dimensional vectors n_{ij} and c_{ij} , where i designates the mountain and j designates the line segment of mountain i .

Major Circles

Major circles are used to quickly eliminate from search as many mountains as possible of the CGI map. A major circle encloses three or more mountains. These circles are visually and analytically determined with the use of a scaled CGI map such as that used in figure 5. They are defined by (1) the coordinates of the center, (2) the radius of the circle, and (3) the table of the mountains included in the major circle. The selection of how many mountains and which ones to contain in a given major circle is arbitrary. The choice of three or four mountains per major circle should be sufficient for the size of the data base discussed in this report. Major circles should not intersect. A possible selection of three major circles is shown in figure 5.

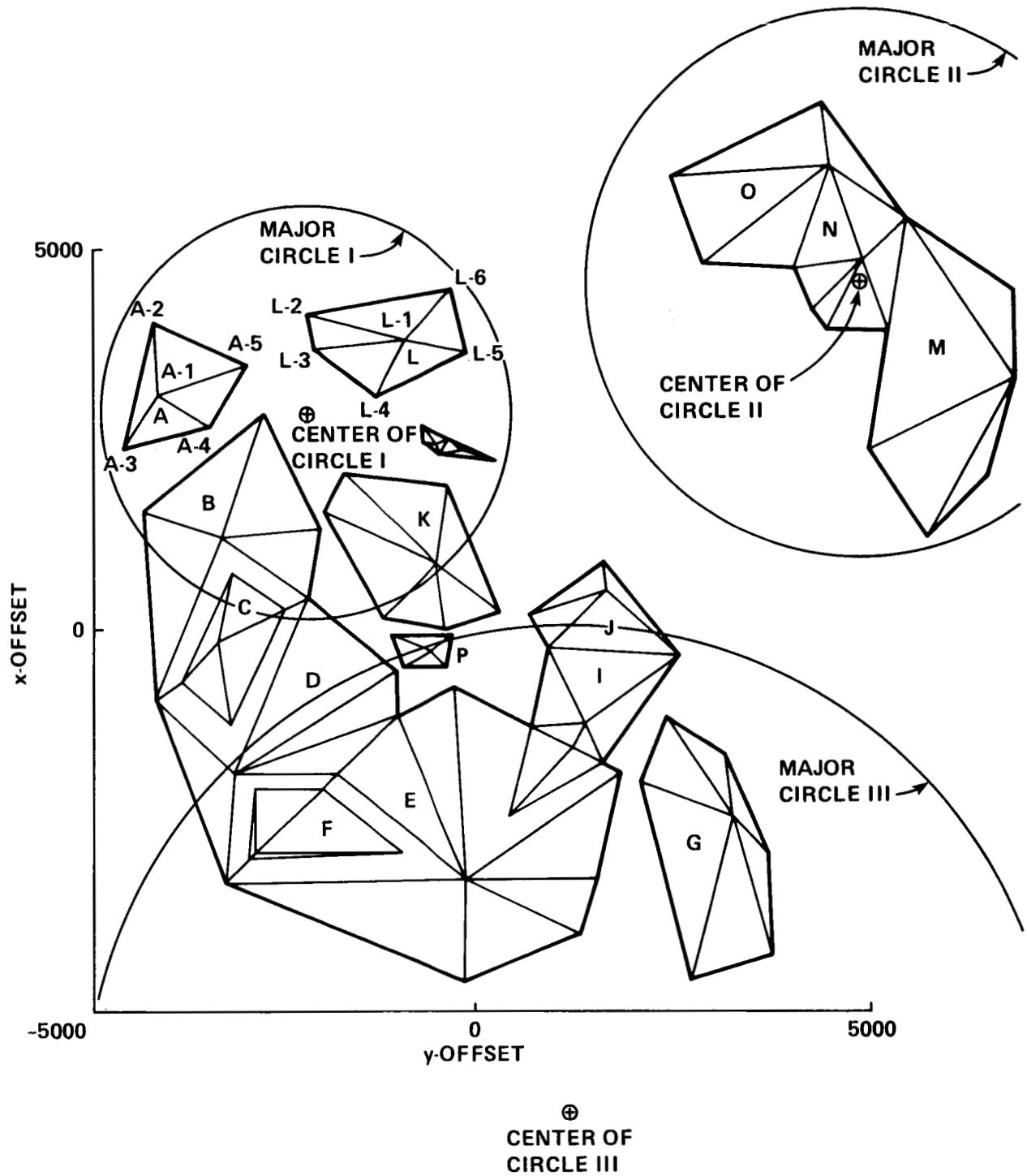


Figure 5.- CGI terrain data base showing location of major circles.

The selection of the mountains and the circles to enclose them is done by trial and error. The circles are first chosen by sight by using a scaled map and a compass. The center and radius can then be determined analytically as follows. Consider circle I in figure 5. It is chosen so as to enclose mountains A, L, Q,

and R. Consider pairs of points near the circumference of the circle that must be enclosed, such as (A2,L6) and (L6,R4). If these points are assumed to lie on the circumference of the circle, then the following is true: (1) Lines A3-L6 and L6-R4 are chords of a circle. (2) The perpendicular bisectors of these lines pass through the center of the circle. Therefore the intersection of these two bisecting lines will be the center of the circle with the radius equal to the distance from this center to one of the extreme points. Points near the circumference of the circle such as A3, A2, L6, and R4 should be tested to verify that they lie within the circumference of the circle determined by this method.

Minor Circles

A minor circle encloses one mountain. These circles are chosen so as to minimize the area which encloses all the vertices of a given mountain. The quantities defining these circles are (1) the coordinates of the center of each circle and (2) its radius.

The method chosen here to determine the circles of minimum area uses all the vertices of a surface. Using all the vertices requires more time than using the peripheral sides only, but time is not critical during preflight calculations.

The minimization method described here has been programmed on the VAX computing system and was used to determine the circle of minimum area enclosing a given set of points. It is basically a "shake and shrink" method, that is, whenever the enclosing circle touches a point, the operator will move the circle away from that point and proceed to shrink the circle until another point is touched and so on until further "shaking and shrinking" cannot enclose all the points. A brief description of the method follows.

1. Establish a rectangle (sides parallel to the x and y axes) that minimally encloses all exterior vertices of the polygon FGHJK. In figure 6 the rectangle is ABCD. The rectangle touches the polygon at points F, H, I, and J.

2. Determine the center of the first trial circle by finding the point of intersection of the diagonals of the enclosing polygon (point E). The vertex at maximum distance from the center of the circle (point F) becomes the radius of the starting circle.

3. Gradually move the center (point E) toward the farthest point (point F) until one or more points of the polygon are contacted.

4. If two points that are not on a diameter are contacted, then gradually move the center toward the midpoint of the line segment between the two points until a third point is contacted.

5. If all three points are within one semicircle of each other (i.e., they form an oblique triangle), then move the center toward the midpoint of the line segment between the two points which are farthest apart.

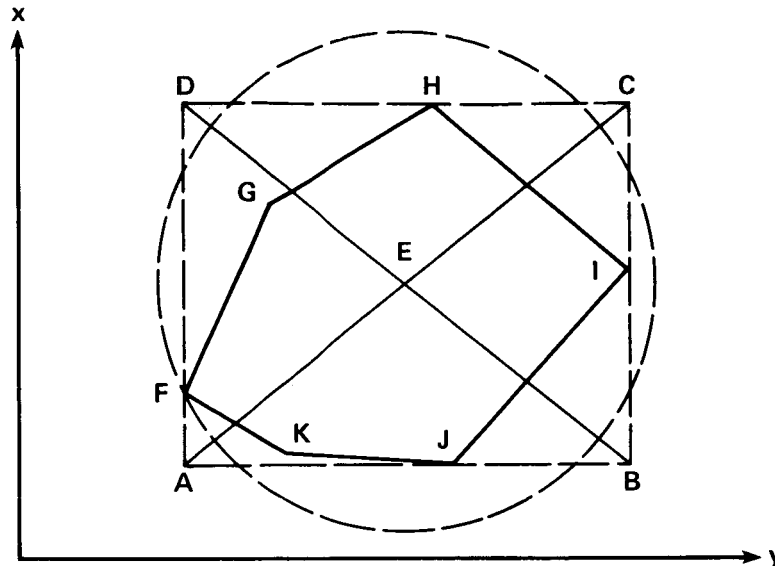


Figure 6.- Determination of initial circle to determine circle of minimum area to enclose a polygon.

6. Repeat 4 and 5 until two points form a diameter or three points form an acute or right triangle. The test for obliqueness is $C^2 > a^2 + b^2$ where C is the longest side of the triangle.

Note that the circles were always chosen to enclose mountains. When other figures such as rectangles, parabolas and ellipses were tried, the number of computations required to determine if the radar LOS intersected these figures would rise dramatically, so they were not used here. This does not necessarily indicate that figures other than circles would not be useful for CGI data bases of different or more extended geometry. However, for the CGI data base considered here, the time efficiency of the circle could not be improved.

COMPUTATIONS DURING SIMULATED FLIGHT

This section describes the determination of line segments of the CGI data base traversed by the radar LOS. This is accomplished primarily by rejecting nontraversed line segments. Initial rejection will occur if the distance from the center of a circle to the radar LOS is compared with the radius of the circle. Rejection occurs if the distance from the center of the circle to the radar LOS exceeds the radius of the circle.

Using the aircraft axes as mentioned earlier (y is the abscissa), the method follows:

1. Given the current position of the vehicle and radar heading, determine the parameters of the radar LOS (projection in the x-y plane).

a. Solve for the intercept (b_R) of the radar LOS with the x -axis (see fig. 7). Use the equation

$$x = M_R y + b_R \quad (5)$$

where x is the ordinate on the CGI map of the line which defines the radar LOS, y is the abscissa on the CGI map of the line which defines the radar LOS, M_R is the slope of the radar heading angle or tangent (ψ_R), and the vehicle position is

$$X_{A/C}, Y_{A/C} \quad (6)$$

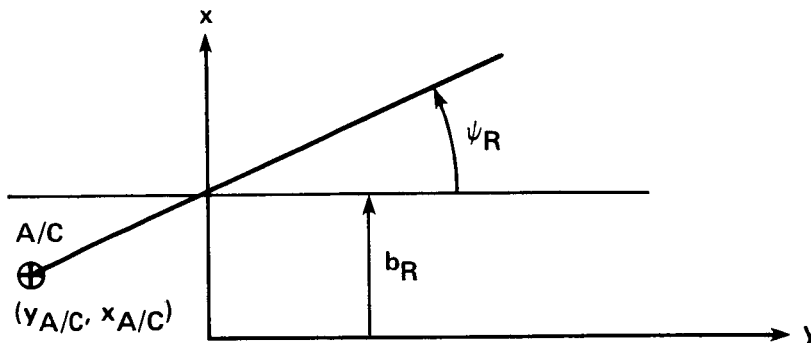


Figure 7.- Determination of parameters of the radar LOS in the x - y plane of CGI data base.

Substitute equation (6) into equation (5) and solve for b_R .

$$b_R = X_{A/C} - M_R Y_{A/C} \quad (7)$$

b. Determine the distance from an arbitrary point in the x - y plane to the radar LOS (fig. 8). From analytic geometry, the equation for this distance is

$$D_{LOS} = \frac{x_c - M_R y_c - b_R}{\sqrt{1 + M_R^2}} \quad (8)$$

where x_c, y_c are the coordinates of the circle being checked. The circles can be either major or minor and D_{LOS} is the distance from y_c, y_c to the radar LOS. As this report is concerned with computer speed, different expressions will be developed for quantities of interest because some variables may be more readily available than others in a program. If equation (8) is used, the square root in equation (4) is calculated only once per given LOS. Note that

$$\sqrt{1 + M_R^2} = \sqrt{1 + \tan^2 \psi_R} = \sec \psi_R = \frac{1}{\cos \psi_R}$$

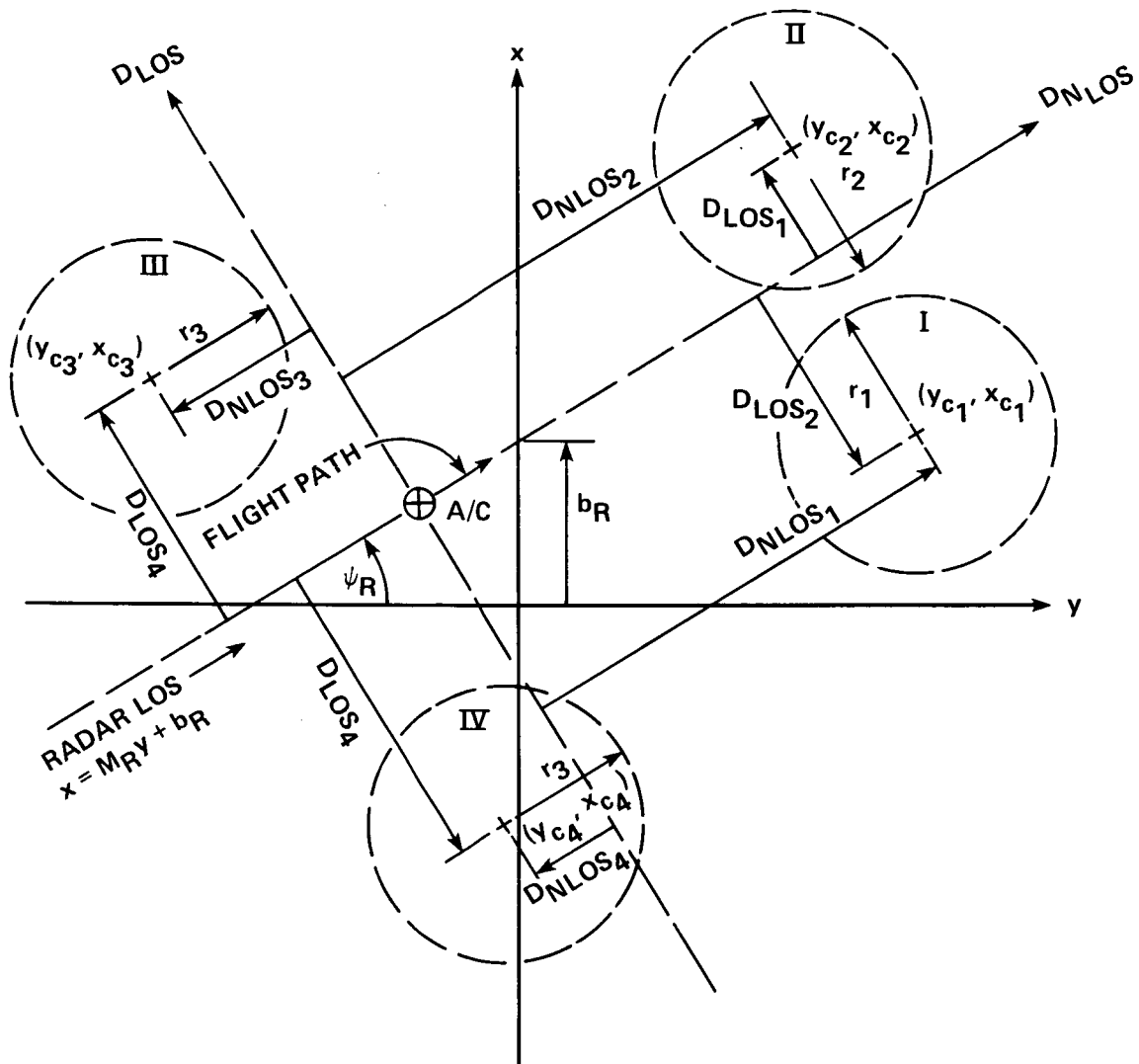


Figure 8.- Interception test of circles.

Therefore, the equation for D_{LOS} can be written as

$$D_{LOS} = (x_c - M_R y_c - b_R) \cos \psi_R \quad (9)$$

c. Determine the distance from any point to the line which is normal to the radar LOS, D_{NLOS} , and passes through the point defined by the vehicle position (fig. 8). The equation for this line is

$$\frac{x - X_{A/C}}{y - Y_{A/C}} = -\frac{1}{M_R} \quad (10)$$

Equation (10) expands to

$$x + \frac{1}{M_R} y - \left(X_{A/C} + \frac{1}{M_R} Y_{A/C} \right) = 0 \quad (11)$$

The distance from a point to a line in equation (7) is

$$D_{NLOS} = \frac{x_c + (1/M_R)y_c - [X_{A/C} + (1/M_R)Y_{A/C}]}{\sqrt{1 + (1/M_R)^2}} \quad (12)$$

Using $M_R = \tan \psi_R$ in equation (12),

$$D_{NLOS} = x_c \sin \psi_R + y_c \cos \psi_R - (X_{A/C} \sin \psi_R + Y_{A/C} \cos \psi_R) \quad (13)$$

Again, circles checked by equation (13) can be either major or minor. A sign is associated with the distance from a point to a line. For D_{LOS} , this sign is not important (see fig. 8). For D_{NLOS} , it will be important because we wish to reject only surfaces to the rear of the origin of the radar LOS. Since D_{NLOS} requires a sign check and D_{LOS} does not, it is faster to reject on the lateral sides of the radar LOS first.

2. Proceed with rejection of mountains and line segments not traversed by the radar LOS. Refer to figure 8.

A mountain or group of mountains contained in a circle whose radius is less than the distance of its center from the radar LOS can be rejected as a candidate or candidates for intersection by the radar LOS. The order of testing for this rejection and acceptance is as follows.

a. If the radii of the major circles is less than the distance of their centers from the radar LOS, the mountains contained in major circles are rejected. This uses equations (4) or (5).

b. Reject any mountains contained in the major circles to the rear of the origin of the radar LOS beyond the range of intersection using equations (12) or (13).

c. Repeat a) and b) with minor circles. The sign of D_{NLOS} is important.

d. Check to determine if the exterior lines of the remaining mountains are intersected by the LOS. If none of the exterior lines of a given mountain intersect the radar LOS, stop checking on that surface; it can be rejected. However, if even one exterior line is crossed by the radar LOS, then all lines of that surface must be checked for intersection.

The method for determining whether a line segment is intersected by the LOS follows.

1. Refer to figure 9. Define the start point for a line segment by the end point whose x coordinate is the lesser of the two end points. Define the x coordinate for the other end of the line as the end point of the line segment. These points that start either above or below the radar LOS with bearings heading away from the radar LOS cannot intersect the radar LOS. Compare the y coordinate of a start point to the y value of the radar LOS evaluated at the x coordinate of the start point. Compare the bearing of the line segment from the precalculations to the bearing of the radar LOS.

2. Refer to figure 9. Some lines from 1) will be indeterminate, that is, their start positions and bearing indicate that they could intersect the radar LOS. The procedure used in 1) must now be applied to the end points of these indeterminate lines. The end points and bearings of these lines have also been prestored.

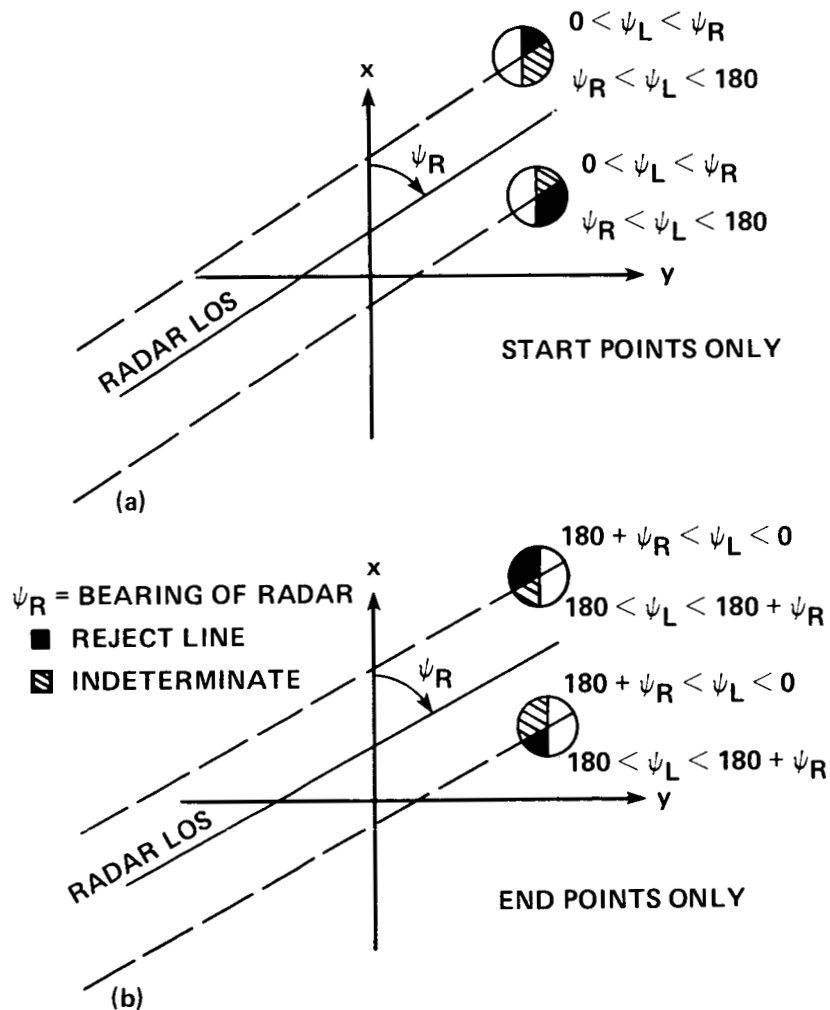


Figure 9.- Determination of intersection of mountain line with radar LOS (end points). (a) Start points; (b) end points.

Figure 10 is a flow diagram of the procedure described in 1). The remaining line segments are intersected by the radar LOS.

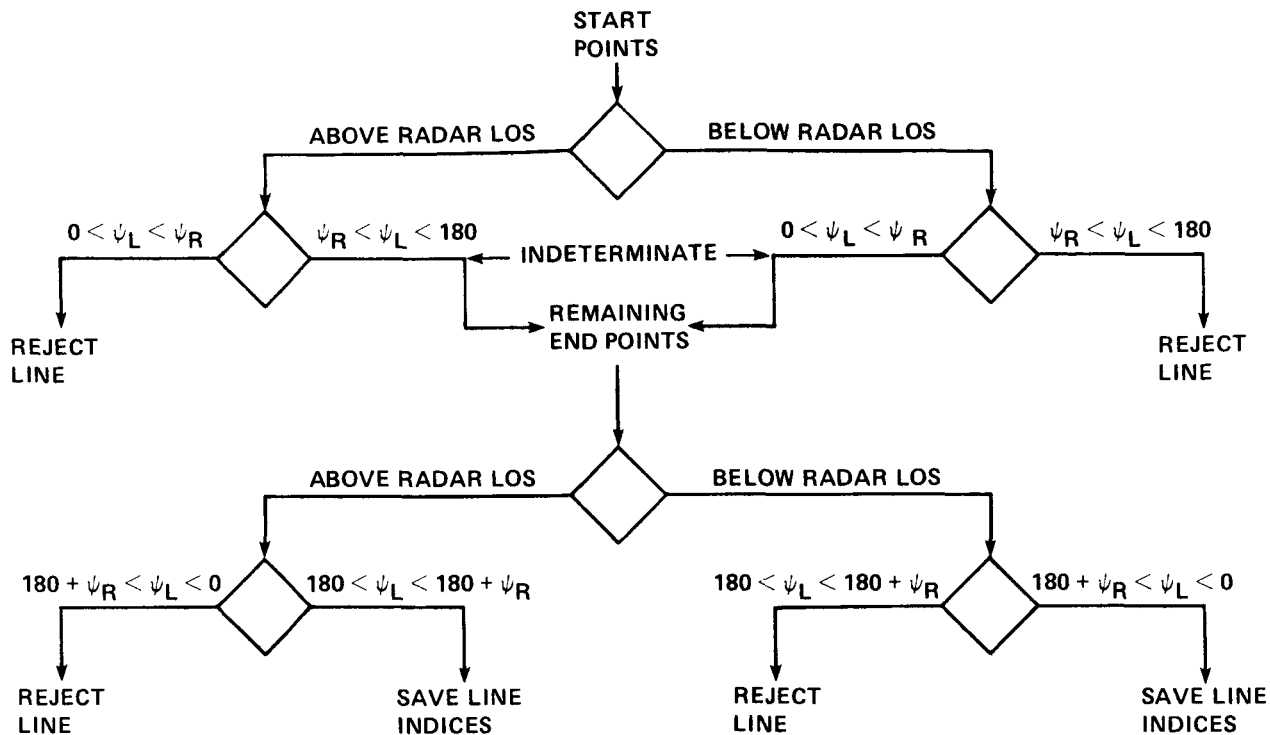


Figure 10.- Flow diagram showing determination of intersection of mountain line with radar LOS.

1. Determine the intersection (x,y) of the radar LOS and traversed line segments. This involves the simultaneous solution of equations (3) and (5) repeated here.

$$x = my + b \quad (3)$$

$$x = M_R y + b_R \quad (5)$$

The m and b in equation (3) have been prestored. The subscripts on x and y in equation (5) have been dropped for clarity.

2. Reject all line segments whose intersection occurs before the source of the radar LOS at the aircraft.

3. The remaining line segments are intersected by the radar LOS starting at its point of origin in the vehicle. Arrange intersections by positive or negative values of x depending on the direction of the LOS. The values of z can be determined at the intersection of the vertical plane containing the radar LOS with the line segments of the CGI map by using the equation for a line through two points in space (x_1, y_1, z_1) , (x_2, y_2, z_2) :

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} = \frac{z - z_1}{z_2 - z_1} \quad (6)$$

where the subscripted quantities are known and the nonsubscripted quantities are unknown.

In equation (6), (x_1, y_1, z_1) and (x_2, y_2, z_2) are respectively the start and end points of line segments on the CGI map. The x and y terms represent the information from 4) above. The z value is to be evaluated.

Consider equation (6) using only x and z terms (the y and z terms could be used instead).

$$\frac{x - x_1}{x_2 - x_1} = \frac{z - z_1}{z_2 - z_1} \quad (7)$$

Solve equation (7) for z :

$$z = \frac{z_2 - z_1}{x_2 - x_1} x - \frac{z_2 - z_1}{x_2 - x_1} x_1 + z_1 \quad (8)$$

or

$$z = nx + c \quad (9)$$

where

$$n = \frac{z_2 - z_1}{x_2 - x_1}, \quad c = z_1 - \frac{z_2 - z_1}{x_2 - x_1} x_1 \quad (10)$$

In equation (9), the quantities n and c have been evaluated and prestored in the two-dimensional vectors n_{ij} and c_{ij} .

We can now compute the quantities required for equations (1) and (2):

$$H = f \text{ (horizontal radar range)} \quad (1)$$

where

$$\text{horizontal radar range} = \sqrt{(X_{A/C} - X_{CGI})^2 + (Y_{A/C} - Y_{CGI})^2} \quad (2)$$

where

H_{surface} = height above CGI data base

H_{surface} = altitude of CGI data base above zero

Equations (1) and (2) give H as a function of horizontal radar range at the intersection of the vertical plane containing the radar LOS with line segments of surfaces on the CGI map.

CONCLUSIONS

The computational method presented here was used in the simulation of a terrain-following vehicle and required 17 msec to execute on a SIGMA 8 computer. A current method required about 120 msec to execute. It should be possible to decrease the execution time significantly in future simulations. It was assumed in using the algorithm for the simulation in reference 4 that the radar gimbals rates were infinite, so the information for the entire line of sight was computed during one or two computer cycles. If realistic finite gimbals rates are assumed, much smaller computation tasks would be required and more prestored quantities could be exploited.

REFERENCES

1. Woodward, Alson C.; and Lagrange, J. Bradner: Terrain-Following Radar: Key to Low-Altitude Flight. National Aerospace Electronics Conference Proceedings (NAECON), May 1979.
2. Clifford, James E., LTC: PAVE Low III, A Long Awaited Capability, Lessons Learned and Projections. NAECON Conference Proceedings, May 1979.
3. Marner, Gene R.; and Pruyn, Richard R.: LHX System Design for Improved Performance and Affordability. J. American Helicopter Society, June 1983.
4. Swenson, Harry N.; Paulk, Clyde H.; Kilmer, Robert L.; and Kilmer, Frank G.: Simulation Evaluation of Display/FLIR Concepts for Low-Altitude, Terrain-Following Helicopter Operations. AIAA Paper 85-3093.
5. Lewis, Michael S.; and Aiken, Edwin W.: Piloted Simulation of One-on-One Helicopter Air Combat at NOE Flight Levels. NASA TM-86686, Apr. 1985.

1. Report No. NASA TM 88324	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle A COMPUTATIONAL METHOD TO MODEL RADAR RETURN RANGE IN A POLYGONALLY BASED, COMPUTER-GENERATED-IMAGERY SIMULATION		5. Report Date July 1986	6. Performing Organization Code
		8. Performing Organization Report No. A-86313	
7. Author(s) F. J. Moran and J. D. Phillips		10. Work Unit No.	
9. Performing Organization Name and Address Ames Research Center Moffett Field, CA 94035		11. Contract or Grant No.	
		13. Type of Report and Period Covered Technical Memorandum	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14. Sponsoring Agency Code 505-66	
		15. Supplementary Notes Point of Contact: F. J. Moran, Ames Research Center, M/S 210-9, Moffett Field, CA, (415)694-5424 or FTS 464-5424	
16. Abstract This paper describes a method for modeling a ground-mapping radar system for use in simulations where the terrain is in a polygonal form commonly used with computer generated imagery (CGI). The method employs a unique approach for rapidly rejecting polygons not visible to the radar to facilitate the real-time simulation of the radar return. This rapid rejection of the nonvisible polygons requires the precalculation and storage of a set of parameters that do not vary during the simulation. The calculation of a radar range as a function of the radar forward-looking angle to the CGI terrain is carried out only for the visible polygons. This method was used as part of a simulation for terrain-following helicopter operations on the vertical motion simulator at the NASA Ames Research Center. It proved to be an efficient means for returning real-time simulated radar range data.			
17. Key Words (Suggested by Author(s)) Radar-return simulation Computer generated imagery Terrain simulation		18. Distribution Statement Unlimited Subject category - 43	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 20	22. Price* A02