A MICROCOMPUTER IMPLEMENTATION OF A SIMPLE VISUAL DISPLAY SYSTEM

JERRY W. CAMPBELL
Florida Power Corporation
and
DR. CHRISTIAN S. BAUER, P.E.,
Department of Industrial Engineering
and Management Systems
University of Central Florida

INTRODUCTION

Computer-driven visual display devices are receiving increasing user acceptance in the development and deployment of simulators and trainers for complex systems. Most of the attention paid these systems has been on the high end of the price/performance spectrum, with full-color, real-time displays for aircraft/spacecraft training simulators becoming the rule and not the exception. There exist other areas of potential application of computer-controlled visual systems, specifically in low-cost trainers designed for initial familiarization or proficiency maintenance on specific areas of systems performance. Such trainers do not need the fidelity or flexibility required of complex systems, but can be optimized for a specific function, such as training pilots for contact approaches to airfields.

This paper discusses some of the basic aspects of implementing "simple" computer graphics and presents an example in the form of simulated night landing displays at Herndon Airport in Orlando, Florida. The three-dimensional projections were calculated and displayed on a video monitor in the form of two-dimensional scenes. Motorola 6800 Microprocessor-based hardware was used as the interface to a video monitor. A description of the various operating parameters for simulated flight landings is presented. Included in the discussion are various pictorial representations of simulated landing approaches at Herndon Airport, and suggestions for improving the performance of the system are outlined.

GRAPHIC PRINCIPLES OF FLIGHT SIMULATION

Since a display screen is a two-dimensional space, one can not display three-dimensional flight perspectives without the use of some mathematical transformations. Three-dimensional coordinate transformation equations must be generated in order to convert flight perspectives, which are three-dimensional, into two-dimensional points.

The three coordinate transformations that must be considered are:

1. Rotation
2. Translation
3. Clipping

For each of these transformations, equations and associated matricies must be established so that the proper perspectives can be generated.

A general rotation in the three-dimensional XYZ-space assumes that the world rotates around the viewer. This in essence is the viewers direction and assumes that the viewers location is at (0, 0, 0) in three coordinate space. A general rotation may be accomplished simply by multiplying a 3 x 3 matrix by a three element vector (1). This matrix only has to be generated once per each viewing direction since it applies to all points for that view.

The rotation or viewer's direction consists of four parameters:

1. Pitch (P)
2. Bank (B)
3. Heading (H)
4. Angle of View (V)

Pitch is a floating point variable that specifies the angle of inclination from which the viewer looks at the scene. Bank is a floating point variable representing the angle at which a viewer's head is tilted sideways when viewing the scene. Heading is the direction the viewer is facing while standing on the XZ plane.

The 3 x 3 matrix, which when multiplied by a three element vector, used to rotate the scene about the origin is shown in Figure 1. This figure shows the rotational matrix being multiplied by the original 3D space coordinate point yielding the rotated point.

The fourth parameter used in the rotation of the original base case is called the viewer's field of view. This parameter is similar to the lens on a camera. This parameter limits the angle that the viewer will see when making the approach to the landing strip.

Field of view is a floating point value representing the tangent of the half field of view. A value of one represents a forty-five degree half field or ninety-degree full view of field. A value of three would represent a narrow telephoto view.

Anytime a change of viewer's direction or rotation takes place, a subroutine must take the change in pitch, bank, and/or heading, as well as field of view and create a new predetermined transformation matrix. This new transformation matrix must then be multiplied by the original data to create a new rotated scene.

TRANSLATION

The viewer's location in flight simulation is always considered to be at (0, 0, 0) in a coordinate space. When the position of the aircraft is at a location other than (0, 0, 0), the points in the data base must be translated while the viewer remains fixed. In other words, the world moves and the viewer remains fixed.

Translation of a point is performed by adding a positive or negative constant to each point of the three-dimensional data base.

CLIPPING

The simple application of translation and rotation to produce a perspective image has two undesirable effects.

1. Objects behind the viewpoint may appear on the screen.

2. Objects may exceed the prescribed limits of the viewpoint.

These effects are eliminated through the use of a clipping algorithm. The operation of line clipping and coding in the program is an extensive process and thus slows down the output from the microprocessor to the video display terminal. That part of the program which performs the algorithm of coding and line clipping is written in BASIC language and thus adds additional computational time to the process. Even though the process is rather slow, it does not affect the accuracy of the transformed view of the field on the video output monitor.

PROJECTION

Once the clipping process yields a line that is visible on the screen, then a perspective projection must be performed. Generating a true perspective image requires dividing by the depth of each point (2). Dividing each starting and ending point of a line that falls within the viewing pyramid by the depth and then multiplying by the half width of the particular video output monitor screen that is being used will give the true perspective image.

A key factor in the projection of a point on the screen is the order in which the transformations are calculated. The position of a point on the display screen which corresponds to a point on some object will be different if different orders of translation and rotation are calculated. For example, if the viewer's location in space (translation) is considered before his viewing direction (rotation), a different projection would result if location had been considered after direction.

Throughout the program for flight display, the following is the order in which the transformations are calculated:

1. X, Y, Z, translation.

2. Heading (rotation about Y axis).

3. Pitch (angle of view to the X, Z plane).

4. Bank

SOFTWARE

Once the input array and control parameters have been integrated into the body of the program, the subroutines shown in Figure 2 are called. These subroutines are called upon to transform each starting and ending point of the data base and to send the resulting screen start and end points to the display device to be displayed.

A commercially available graphics package (1) was used to provide much of the processing logic in the program sections performing the mathematical transformations.

The matrix generator subroutine generates the 3 x 3 transformation matrix requried for rotation. The input parameters P, B, H, and V are used to generate the rotational transformation that will be used in the 3D to 2D converter. Contained within the matrix generator subroutine is the sine/ cosine subroutine. This subroutine generates the cosine and sine of a value in degrees. This subroutine is needed because BASIC language does not have trigonometric functions. This matrix need only be calculated once for the given control parameters. All lines in the scene will use the same transformation matrix.

The program must then feed the array of 3D input points to the matrix multiplier subroutine. The matrix multiplier subroutine takes the viewer's location values and adds them to the start and end points of each line in the data base one line at a time. The points are then multiplied one at a time by the transformation matrix calculated in the matrix generator subroutine. Each rotated start and end point is then transferred to the clipping subroutine.

The clipping subroutine determines if the line just translated and rotated is on the screen or off the screen and accordingly displays, clips, and displays, or eliminates it. As each starting and ending point for a line is translated, rotated, and has the necessary clipping performed, it must be sent to the display device driver subroutine. The display device driver subroutine was written in M6800 machine language and contained several subroutine options within the program; these were the Erase and Draw routines.

The Erase subroutine is run once each time a new transformation matrix has been calculated. The subroutine clears the video monitor display of any previous objects. The Draw subroutine accepts the data from the 3D to 2D converter and outputs to the video display monitor electrical impulses that correspond to scan lines on the display image. Each time the 3D to 2D converter calculates a new translated, rotated, and clipped line, the Draw subroutine is called to output to the video monitor.

Once all the lines from the base data have been sent to ask for a new position and direction, the program will check to see if the input variables have changed. If none of the variables have changed, the program will not erase the screen, but instead ask again for new viewer parameters. Once any one of the six parameters are changed, the program will then calculate a new transformation matrix, erase the screen, and calculate a new display image on the video monitor.

This program was written with the versatility to include many different data bases, larger or smaller screens, wider or narrower fields of view, and different positions and directions. The major portion of the program is written in the BASIC language, which makes the calculations of perspectives slow.

HARDWARE

The computer hardware used in this study was built from kits produced by the Southwest Technical Products Corporation of San Antonio, Texas. The major components of the system and the associated costs were as follows:

1. M6800 microcomputer (32K RAM, 1K ROM) at an approximate cost of $1000.
2. Dual mini-floppy disk drive at an approximate cost of $1000.
3. Graphic Interface Terminal at an approximate cost of $200.
4. Video output monitor at an approximate cost of $200.
5. Console Terminal at an approximate cost of $500.

The dual floppy drive has the ability to store programs on small magnetic disks. The magnetic disk affords the ability to store many different data bases which are readily available to be called up by the operator. This gives the operator the ability to change the training enviroment from one area to another within seconds.

The graphic interface terminal serves as a buffer between the video monitor and the M6800 microcomputer. The graphic interface takes digital output data from the microcomputer and converts it to a composite video signal for display.

SYSTEM OPERATION

The most crucial part of constructing a three-dimensional graphic program is testing the system to see if results are what is expected. The entire process of rotation, translation, clipping, perspectives, and visual display must be integrated into a software module or modules that have coordinate values of a point in the three-dimensional space.

This module can be depicted as a transfer system as shown in Figure 3 (3). The input consists of points X, Y, Z, in the three-dimensional space. Along with the inputs are eight transformation parameters.

The three angles of rotation P, B, and H.

The three translation parameters X, Y, and Z.

The two scaling factors W and V.

The output variables that one is expecting to display are X_p and Y_p, which are in 2D screen coordinates.

The intended purpose of any flight simulator is to accurately generate the perspective views of the data base for a given position. The data base selected for this study was that of Herndon Airport located in Orlando, Florida. As viewed in Figure 4, Herndon has two major runways in a basic X configuration.

In order to simulate flight landings on the runways at Herndon Airport, the three angles of rotation and the three translation parameters must be supplied to the program. Figure 5 depicts various points on and off the runway that were to simulate flight landings. Table 1 is a listing of the various points on the runway layout that have been inputted to the computer and the translated and rotated picture displayed on the video monitor.

In order to simulate an approach landing on the runways at Herndon, a general view of the field was generated to provide an overall validation of the model performance. Figure 6 is a view of Herndon Airport when the values of the angles of rotation and translation parameters are:

X(3) = 0 (origin of X-axis)

Y(3) = 5000 feet (altitude)

Z(3) = 0 (origin at Z-axis)

P = -90^{o}

B = 0⁻

H = 0^{o}

Approach landing to Runway 7 will be depicted in a series of figures. Figure 7 is a view of Herndon Airport when the values of the angles of rotation and translation parameters are:

X(3) = -10,000 feet

Y(3) = 3,000 feet (altitude)

Z(3) = 10,000 feet

P = -15 degrees

B = 0 degrees

H = 50 degrees

Figure 8 is a view of the airport from:

X(3) = -5,000 feet

Y(3) = 1,500 feet

Z(3) = -5,000 feet

P = -20 degrees

B = 0 degrees

H = 50 degrees

Figure 9 depicts an even closer view of Herndon's Runway 7. The parameters are:

X(3) = -2,500 feet

Y(3) = 800 feet

Z(3) = -3,500 feet

P = -25 degrees

B = 0 degrees

H = 50 degrees

Examination of Figure 10 shows that the plane is about to touch down on Runway 7. The parameters are:

X(3) = -1,700 feet

Y(3) = 200 feet

Z(3) = 2,700 feet

P = -25 degrees

B = 0 degrees

H = 50 degrees

The final approach and touchdown scenario on Runway 7 is depicted in Figure 11. The parameters are:

X(3) = -1,500 feet

Y(3) = 100 feet

Z(3) = -2,400 feet

P = -15 degrees

B = 0 degrees

H = 50 degrees

CONCLUSIONS AND DIRECTIONS FOR FUTURE STUDY

The evolution of computer graphics with its associated graphical displays has without a doubt surfaced as one of the most fascinating areas of computer technology. The essence of this research was to explore the feasibility of a microprocessor to demonstrate the use of low-cost computer graphics for flight simulation visual display. Many large-scale flight simulators are available that utilize computers to create flying conditions. A major obstacle to widespread use of these relatively high costs is associated with the complexity of a total systems simulation. There is a large potential demand for limited-function flight simulators and other specialized trainers if certain parameters can be achieved:

1. Low Cost: The system must be designed to

be implemented on a small computer, equipped with a relatively small highspeed memory, a mass storage device, a linedrawing display and input devices appropriate for the intended application, possibly including a keyboard and joystick.

2. Versatility: The use of a local dedicated computer rather than a remote time-shared system enables the system to provide a good response.

3. User- Oriented: The system is designed to be operated and programmed by interaction with a single, high-level language.

Because display devices have limitations, display systems are a compromise between the interests of men and capabilities of the equipment (4). Two of the major limitations of the hardware used in this study were resolution and round-off.

Resolution is a measure of discrimination within a fine detail. Resolution depends partly upon the visual activity of the eye and partly upon the resolution of the display itself. The resolution of a display depends upon the size of its screen, the strength of the phosphor and the graininess of the web upon which the display is projected. Thus, some of the blurriness which shows up on the pictures has to do with the size of the screen which causes resolution.

The second limitation on the screen image is associated with numerical round-off. The graphics display hardware can only accept integer values as inputs to the video display unit. Thus, some of the lines exhibit a smooth appearance, such as the center lines of runways because of the round-off that is inherent in a grid definition of only 64 x 96 points. This is an inherent limitation of the equipment that is currently being used.

One major area for future research is to increase the speed at which the new projections are calculated and projected on the video monitor. In order for the system to be used for actual flight simulation, the program must be able to generate 20-30 new projections and associated display updates within one second. This performance can conceivably be approached with microprocessor based-systems by converting the program from BASIC to machine language and/or by investigating the possibility of distributed computers using multiple microprocessors.

The second area where additional research can be done from this study is to make the position and direction parameter inputs come from cockpit controls and displays instead from the console terminal. These inputs can be generated by the use of transducers and can be fed directly into an analog to digital converter. The outputs from the converter would then be the inputs to the microprocessor to be used for generating new projections.

It is clear that great advances have been made in building highly versatile, and user-oriented, graphics transformation processors. It is anticipated that the next few years will bring about the development of high-quality, high-performance, and inexpensive training adjuncts to large scale display simulators that are currently in use.

REFERENCES

1. Sub Logic, Inc., 1976. "BASIC microcomputer graphics user's manual." Tampa, Florida: Sub Logic. Inc.

2. Newman, W. M., and Sproull, R. F., 1973. Principles of interactive computer graphics. New York: McGraw-Hill.

3. Giloi, W. K., 1978. Interactive computer graphics. Englewood Cliffs, New Jersey: Prentice-Hall.

4. Miller, W. E., 1968. "Visual information display systems." Survey presented to National Aeronautics and Space Administration, Washington. D. C.

	(Feet)			(Degrees)		
No.	X	Y	Z	P	B	H
1	-6000	600	-6500	-45	0	40
2	-3500	400	-3700	-45	0	40
3	-2000	200	-2800	-35	0	40
4	-1500	100	-2400	-30	0	40
5	-1000	100	-1900	-30	0	225
6	1400	200	-3000	-15	0	140
7	1600	200	-3600	-35	0	145
8	2500	500	-3900	-35	0	145
9	-5000	800	2600	-40	0	-45
10	-3500	300	1500	-40	0	-45
11	-2600	200	900	-35	0	-45
12	-2300	200	700	-10	0	-45
13	-1000	300	-500	-35	0	135
14	4200	800	3000	-35	0	225
15	3500	600	2300	-35	0	225
16	3000	400	2000	-40	0	225
17	2700	200	1800	-30	0	225

TABLE 1

Three-Dimensional Data Points for Various Landing Approaches to Herndon Airport

$$
[X'Y'Z'] = [XYZ] \begin{bmatrix} [\mathrm{COS}\ H \ \mathrm{COS}\ B + \mathrm{SIN}\ H \ \mathrm{SIN}\ P \ \mathrm{SIN}\ B] & [-\mathrm{COS}\ H \ \mathrm{SIN}\ B + \mathrm{SIN}\ H \ \mathrm{SIN}\ P \ \mathrm{COS}\ B] & [\mathrm{SIN}\ H \ \mathrm{COS}\ P] \\ [\mathrm{COS}\ P \ \mathrm{SIN}\ B] & [\mathrm{COS}\ P \ \mathrm{COS}\ B] & [-\mathrm{SIN}\ P] \\ [-\mathrm{SIN}\ H \ \mathrm{COS}\ B + \mathrm{COS}\ H \ \mathrm{SIN}\ P \ \mathrm{SIN}\ B] & [\mathrm{SIN}\ H \ \mathrm{SIN}\ B + \mathrm{COS}\ H \ \mathrm{SIN}\ P \ \mathrm{COS}\ B] & [\mathrm{COS}\ H \ \mathrm{COS}\ P] \end{bmatrix}
$$

Where $[X'Y'Z']$ = Rotated Point

$[X\ Y\ Z]$ = Original Point

P = Pitch

B = Bank

H = Heading

Figure 1. Rotational Matrix multiplied by original 3D space coordinate point yielding the rotated point

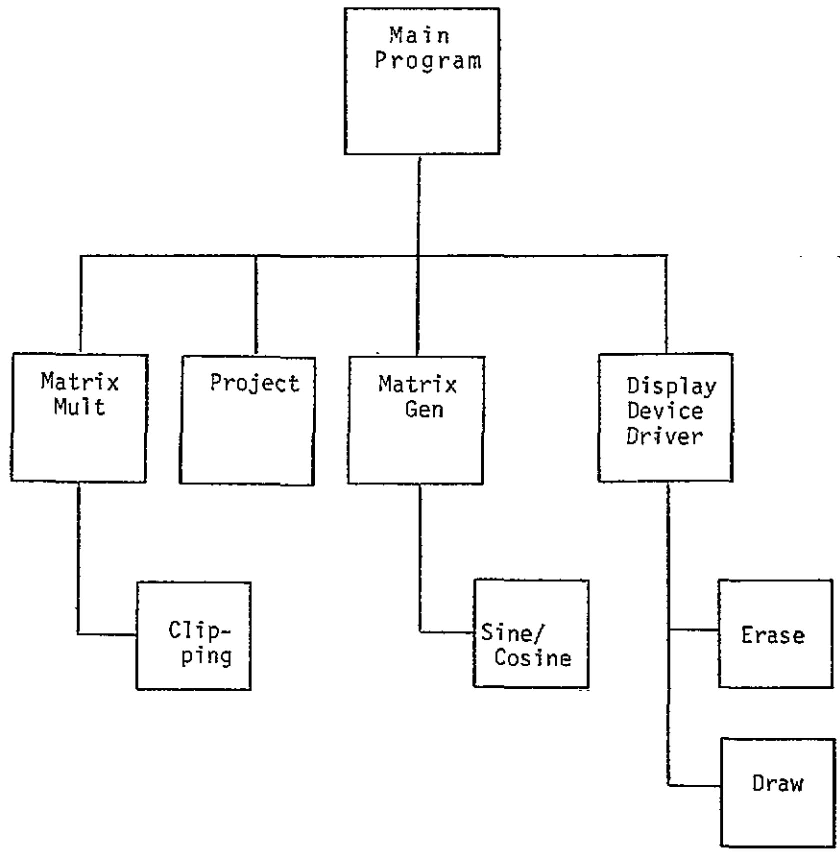

Figure 2. Subroutine Structure for Flight Simulator

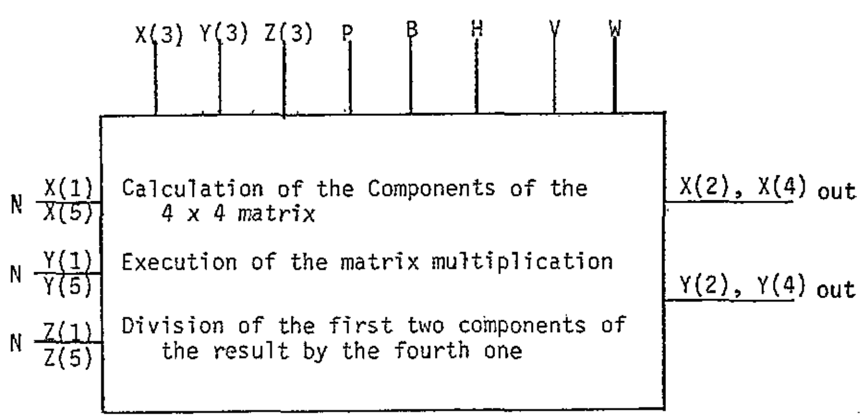

Figure 3. Transfer System view of a module for rotation, translation, clipping, and perspective on a line from (X_1, Y_1, Z_1) to (X_5, Y_5, Z_5)

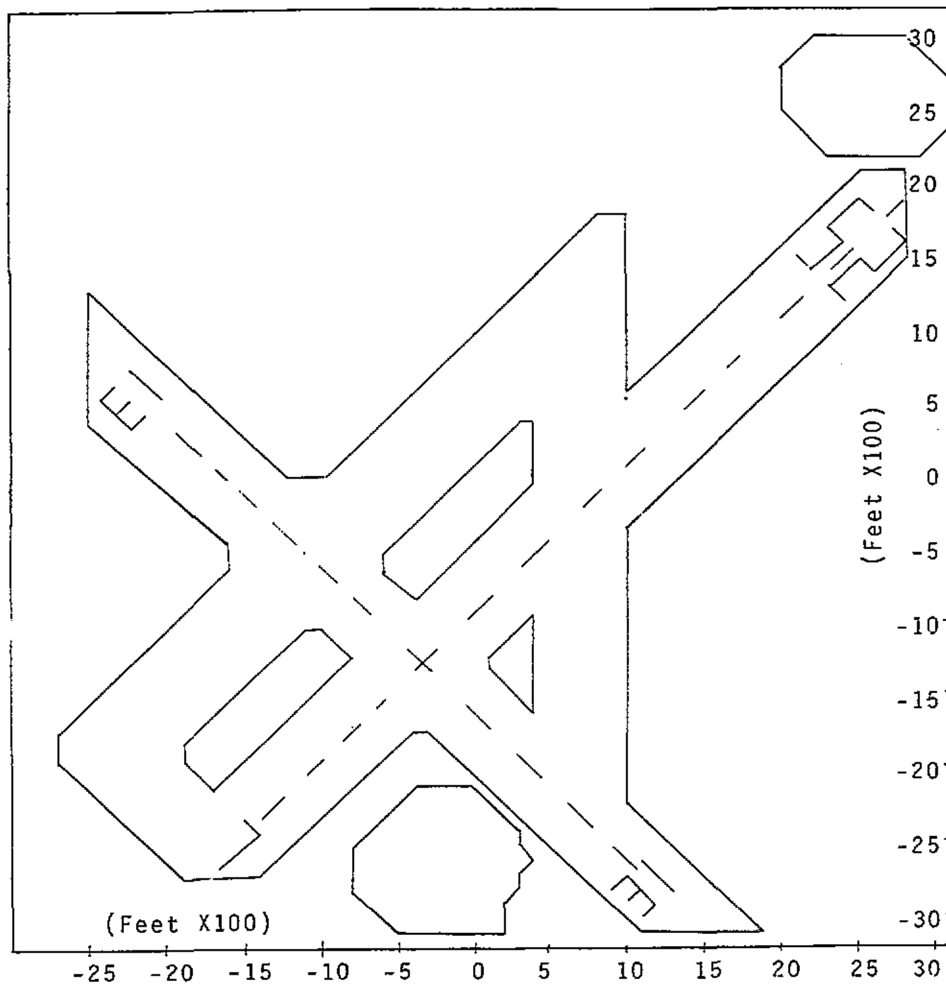

Figure 4. General Layout of Herndon Airport and Associated Runways

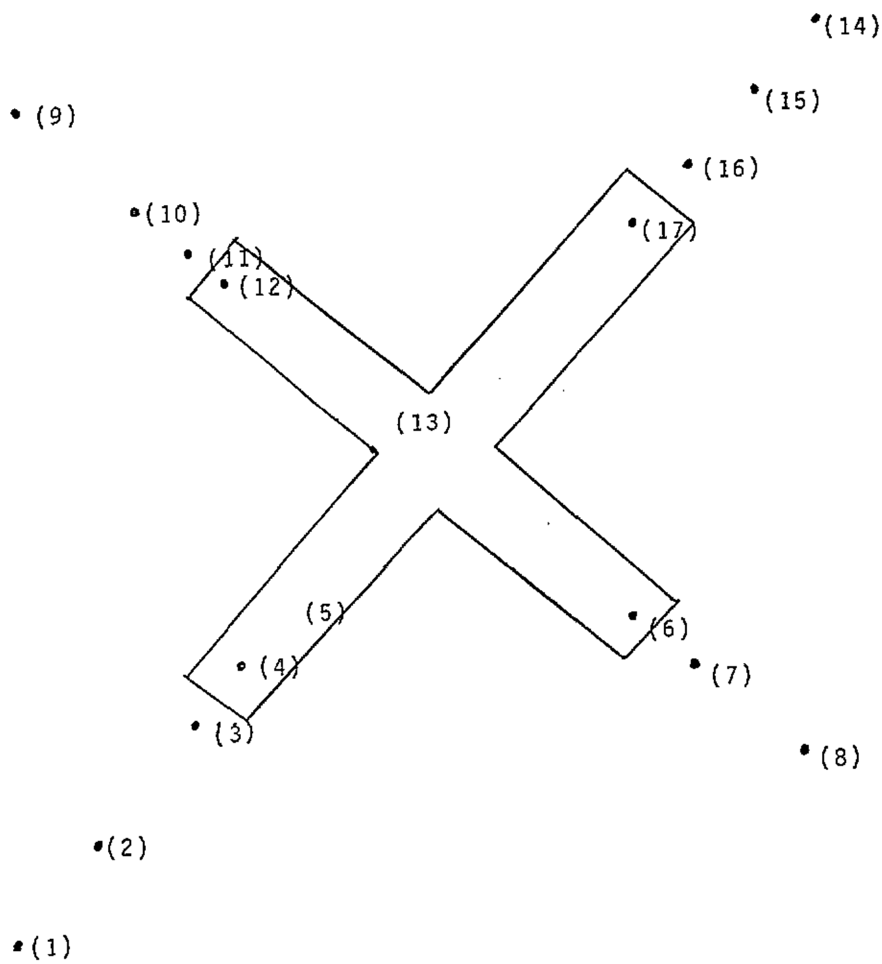

Figure 5. Point Approaches to Herndon Airport

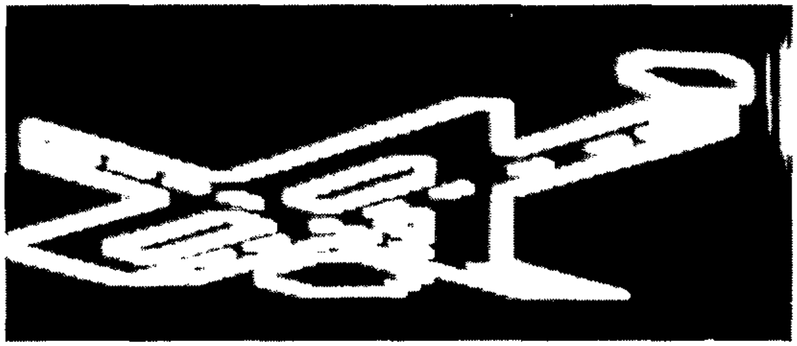

Figure 6. Aerial View of Herndon Airport at 5000 Feet

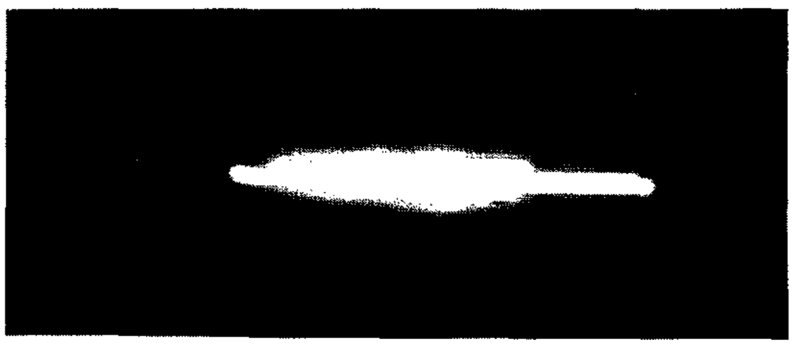

Figure 7. Aerial Approach to Runway 7 at 3000 Feet Altitude

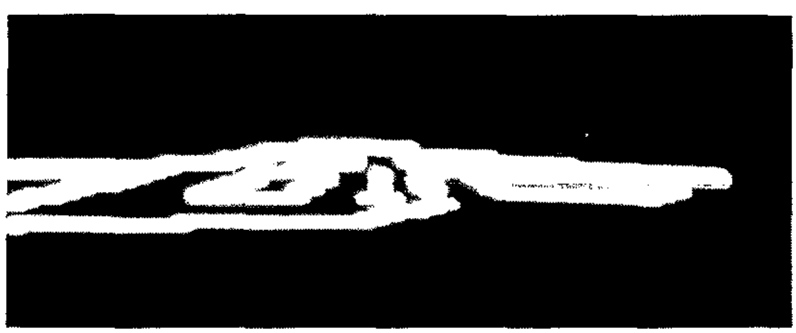

Figure 8. Aerial Approach to Runway 7 at 1500 Feet Altitude

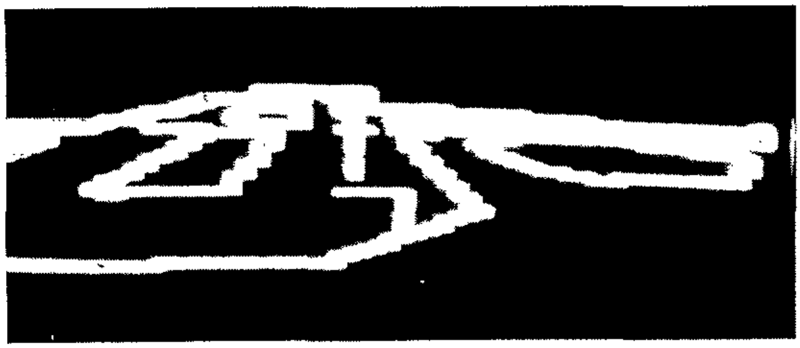

Figure 9. Aerial Approach to Runway 7 at 800 Feet Altitude

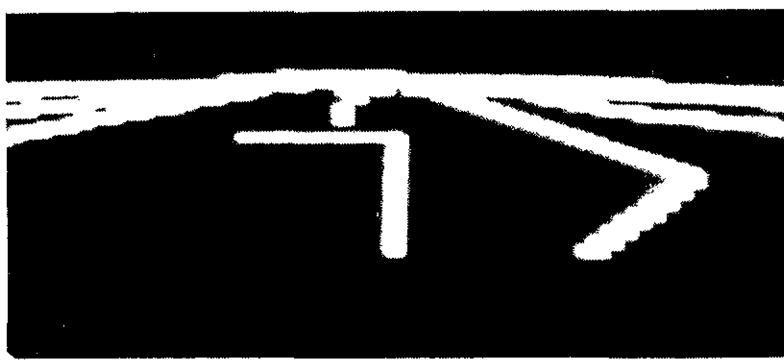

Figure 10. Aerial Approach to Runway 7 at 200 Feet Altitude

Figure 11. Aerial Approach to Runway 7 at 100 Feet Altitude

ABOUT THE AUTHORS

MR. JERRY W. CAMPBELL has 10 years experience with the Florida Power Corporation, where he does computer process monitoring and control application in power generation systems. He has a M.S.E. degree in industrial engineering from University of Central Florida.

DR. CHRISTIAN S. BAUER is an Associate Professor in the Department of Industrial Engineering and Management Systems at the University of Central Florida. He is currently performing research in visual display systems for trainers with specific-function applications. He has 13 years of experience in computer application in engineering.