A DATA BASE GENERATION SYSTEM FOR DIGITAL IMAGE GENERATION

BY
ARTHUR P. SCHNITZER
THE SINGER COMPANY, LINK DIVISION

INTRODUCTION

In the last ten years, we have witnessed a fortyfold increase in the edge processing capability of real-time digital image generation (DIG) systems. While it would appear unlikely that the next decade will produce another fortyfold increase in capacity, one thing, at least, seems clear. Like some insatiable science fiction monster, today's DIG system is devouring data bases at a prodigious rate that promises to be ever-increasing. To satisfy this gargantuan demand, as well as the desire of users for more detailed and realistic simulation, greater emphasis will have to be placed on the data base generation process. Indeed, we have already seen the beginnings of this trend in recent procurement specifications such as that for the DIG for the FB-111 simulator, (1) which not only specifies a total data base storage capacity several times the size of the deliverable data base, but also requires the delivery of a data base generation facility, thus hinting at the Government's intent to fill up this spare storage capacity after delivery. In recognition of the substantial effort required to generate visual simulation data bases, there is even a quantitative specification of data base generation speed.

A comparable situation exists in the radar simulation field, and it is interesting to observe what has transpired there. The need for greater modification capability and higher resolution has led to the adoption of the digital radar land mass simulation (DRLMS) system over the earlier analog (transparency) system far faster than has been the case with visual simulation. Data base generation for DRLMS started with automated scanning and digitization of existing analog RLMS transparencies, and is now tending to standardize on data compiled by the Defense Mapping Agency Aerospace Center (DMAAC) in St. Louis, Mo. At DMAAC, there has been a substantial investment in equipment and personnel to facilitate the encoding of photographic and topographic data, primarily for radar simulation purposes.

The applicability of this data for forward-looking infrared (FLIR) and low-light-level television (LLLTV) sensor simulation has apparently been recognized, (2) and from there it is a natural step to extend its use to visual simulation. Not only would this reduce the cost of visual data base generation, but it would also result in correlated imagery spanning a wide array of sensing equipment. This paper describes a system which has been developed to utilize DMAAC source data for DIG visual simulation; it also describes the laboratory real-time DIG system used for validation of the results.

LABORATORY DIG SYSTEM

Concurrent with the development of the data base generation system was the development of a real-time laboratory DIG system, which served as the test-bed and demonstration vehicle for the data base generation development. The laboratory DIG hardware will be briefly described to illuminate the environment in which the data base must operate, although continuing development has since resulted in more powerful capabilities.

Figure 1 is a block diagram of the laboratory DIG system. Operation of the system is controlled by a Datacraft (now Harris) 6024/5 general-purpose computer. This same computer was the central controller for the data base generation system. The computer communicates directly with the frame calculator, whose function is to translate, rotate, clip, and project data base face boundaries onto the computed image plane, and store the results in the double-buffered edge list. The frame calculator operates on object descriptions extracted from the active data base (ADB) store, a 16K x 48-bit core memory loaded through an alternate data path by the general-purpose computer.

Edge list contents for an entire frame's computation are processed on a line-by-line basis by the scanline computer to produce a sequence of visible scanline segments, which are converted to video information

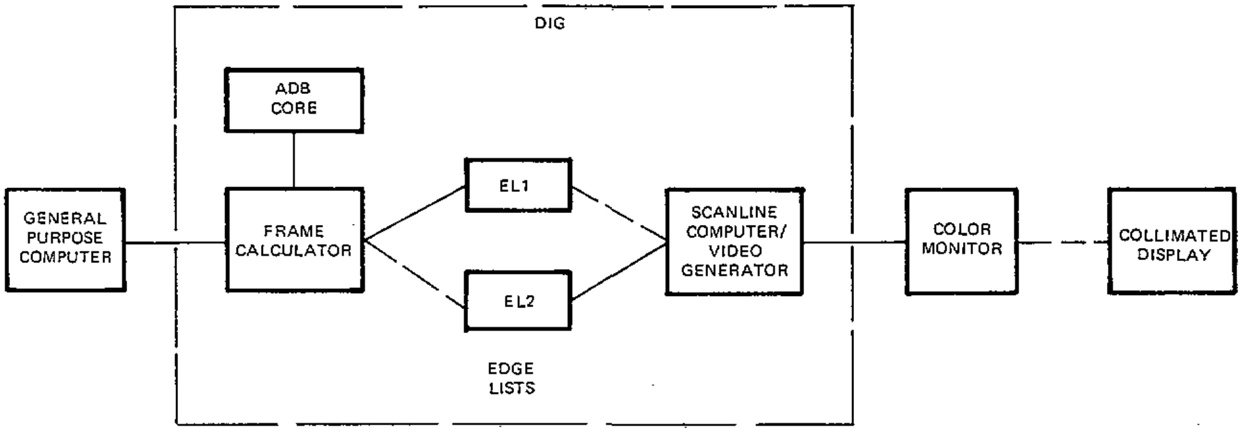

Figure 1. Laboratory Dig System

by the video generator. This is displayed on a high-resolution color monitor viewed through a collimating mirror-beamsplitter display system.

As far as features and capabilities are concerned, the system handles about 5,600 face boundaries in real-time (30 frames per second) and generates 1,000 scan lines, each with 1,000 resolution elements. Up to 256 edge intersections can be accommodated per scan line, and this level can be sustained on every scan line. Smooth-shading is incorporated to provide a rounded appearance to curved surfaces. At any one time, up to 64 distinct colors may be displayed; each color is specified by 18 bits, 6 each in the red, green, and blue primaries, to form a potential color space of more than 262,000 possible hues. Horizontal and vertical edge smoothing techniques have been included to minimize quantization effects.

The system uses a priority list technique for resolving visibility contentions between objects which overlap when projected onto the image plane. This approach constrains objects to be convex, so as to preclude the overlapping of faces of the same priority. Up to 256 priorities may be specified. Object priorities are typically assigned in real-time by the general-purpose computer, using a separating plane approach.[3,4] Types of objects which may be specified include two-dimensional objects (parallel to the ground plane), three-dimensional objects (solid or oblique to the ground plane), point lights, strings of point lights, and lines. Lights can be specified to be either 1 scanline by 1 resolution element or 2 scanlines by 2 resolution elements in extent. Lines can be either 1 or 2 elements wide. Objects which are not lines or lights may either have their face intensities computed dynamically, as a function of sun position and face orientation, or be specified to be constant. This latter capability has been used to minimize real-time computation requirements and to provide for unusual sensor characteristics, such as are found in FLIR or LLLTV simulation. Owing to a particular choice of clipping algorithm, all non-light, non-line faces are constrained to be convex (this constraint has been removed in later designs).

DATA BASE GENERATION SYSTEM HARDWARE

The physical emodiment of the data base generation system (DBGS) is depicted in Figure 2. The central element is the central processing unit (CPU), which is the same one used for the real-time DIG system. It includes 48K 24-bit words of 1-microsecond core memory. The DIG and display are also included as part of the DBGS hardware, since debugging and final proofing are an integral and necessary part of the data base generation process. In addition, in conjunction with the joystick and controls, it provides a medium for interactive data base modification, as described in a later section.

Peripheral devices connected to the CPU, aside from the normal teletypewriter, card reader, and line printer, are a 28-megabyte (9 million words) moving-head disk, used for data base and program storage, and a 9-track magnetic tape system, used for backing up the disk and providing additional flexibility to interchange data bases. It is also the input device for the DMAAC data base.

The display terminal is a Tektronix Model 4010-1, consisting of a storage screen display and alphanumeric keyboard, which includes hardware vector and character generation. Graphic input is also permitted

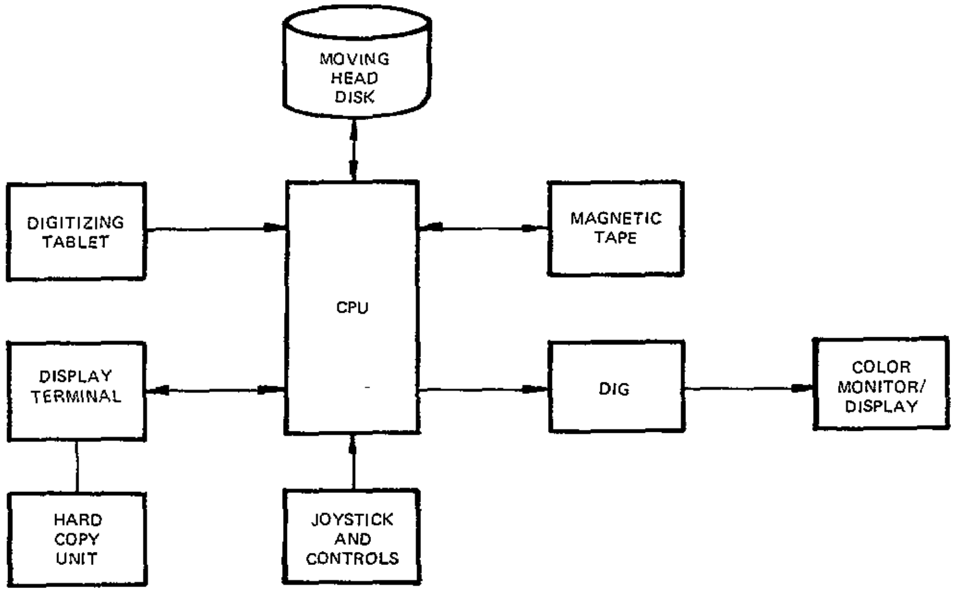

Figure 2. Data Base Generation System Hardware

through thumbwheel control of a crosshair cursor. This device serves as the primary unit for graphic interaction and for geometry verification prior to real-time display with the DIG. Connected to the display terminal is a Tektronix Model 4610 hard copy unit, which produces facsimile copies of any information on the display screen. This is used to provide automatic documentation of the data base.

The digitizing tablet is a Science Accessories Corporation Model GP-2 Graf/Pen, with 12-bit resolution over a 36- by 36-inch active area. This is used as the primary input device for the modification of the data base, and for the entry of new features or geometry not contained in the DMAAC source data.

THE DMAAC DATA BASE

The DMAAC digital data base (DDB) was conceived to satisfy the requirements for digital radar land mass simulation (DRLMS), initially on Project 1183.[5,6] At the time of this work, the DDB consisted of two distinct portions: an elevation file, which defined predominant ground contours by means of elevation values at evenly spaced (in latitude and longitude) grid points, and a cultural file, which consisted of orthographic outlines of cultural features, along with some natural features such as lakes and rivers. It is the latter file on which the present work is based, there having already been some work done on processing grid point information to produce a DIG-compatible data base.[7,8]

The cultural file is organized into manuscripts, rectangular (in latitude/longitude) areas at a given data resolution level. Of these there are three, the finest being 50-foot resolution. Within each manuscript, each feature is assigned a unique feature analysis code (FAC) number, and is encoded geometrically by a directed sequence of vertices, defined in geocentric coordinates, and a predominant height, which is equal to zero for ground level (2-D) features. Additional descriptive information carried along with the feature includes the feature identification (FID), a number which determines the feature's general category (tennis court, commercial building, water tower, etc.), surface material type, percent of roof cover, and percent of tree cover.

It was considered important that the trial DIG data base consist of as much detail as possible. At the time of this work, only the 57,000-square-mile Project 1183 data base was available; the most detailed manuscript therein (Level 3B) consisted of 1 square mile of downtown Las Vegas, and contained about 4,300 vertices. A plot of the manuscript is shown in Figure 3. The decision to use this manuscript was met with grudging acceptance by the data base modeler, who reluctantly volunteered to conduct on-site data gathering activities to supplement the DMAAC DDB.

DATA BASE GENERATION SYSTEM

The data base generation process, as well as the software modules required to implement

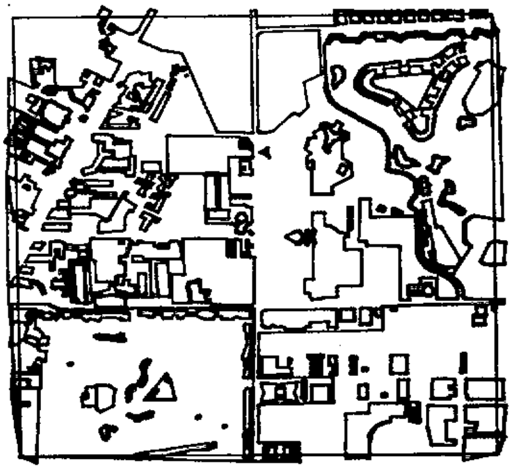

Figure 3. DMAAC Level 3B Plot of Downtown Las Vegas

the process, is illustrated in Figure 4. The elements of this system are discussed below, and consist of the Data Tape Reformatting Program, the Data Base Enhancement Program, the Data Base Compiler, the Perspective Line Drawing Routine, the Demo Program, and ancillary programs.

DATA TAPE REFORMMATING PROGRAM

The DMAAC DDB was furnished on a magnetic tape prepared on a UNIVAC 1108, a 36-bit machine. The function of the Data Tape Reformatting Program (DTRP) was to search the tape for the appropriate manuscript, convert all items for compatibility with the 24-bit word length of the Datacraft CPU, and transform the geocentric vertex coordinates to a rectilinear system in a plane tangent to the earth at the southwest corner of the manuscript. The output of DTRP is written onto a disk file (DTRP file) for convenient access by the Data Tape Interpretation Program (DTIP). DTRP is executed only once for a given manuscript, and involves no human interaction.

DATA TAPE INTERPRETATION PROGRAM

The DTRP file contains a combination of 2-D and 3-D features, of which the 3-D features must be "extruded" out of the ground plane to the extent of their predominant height. As in the case of the Level 3B manuscript, feature outlines which should ideally be smoothly curved (e.g., the Landmark Tower) are ragged and uneven, and contain unnecessary vertices. In addition, the manuscript contains many more vertices than the

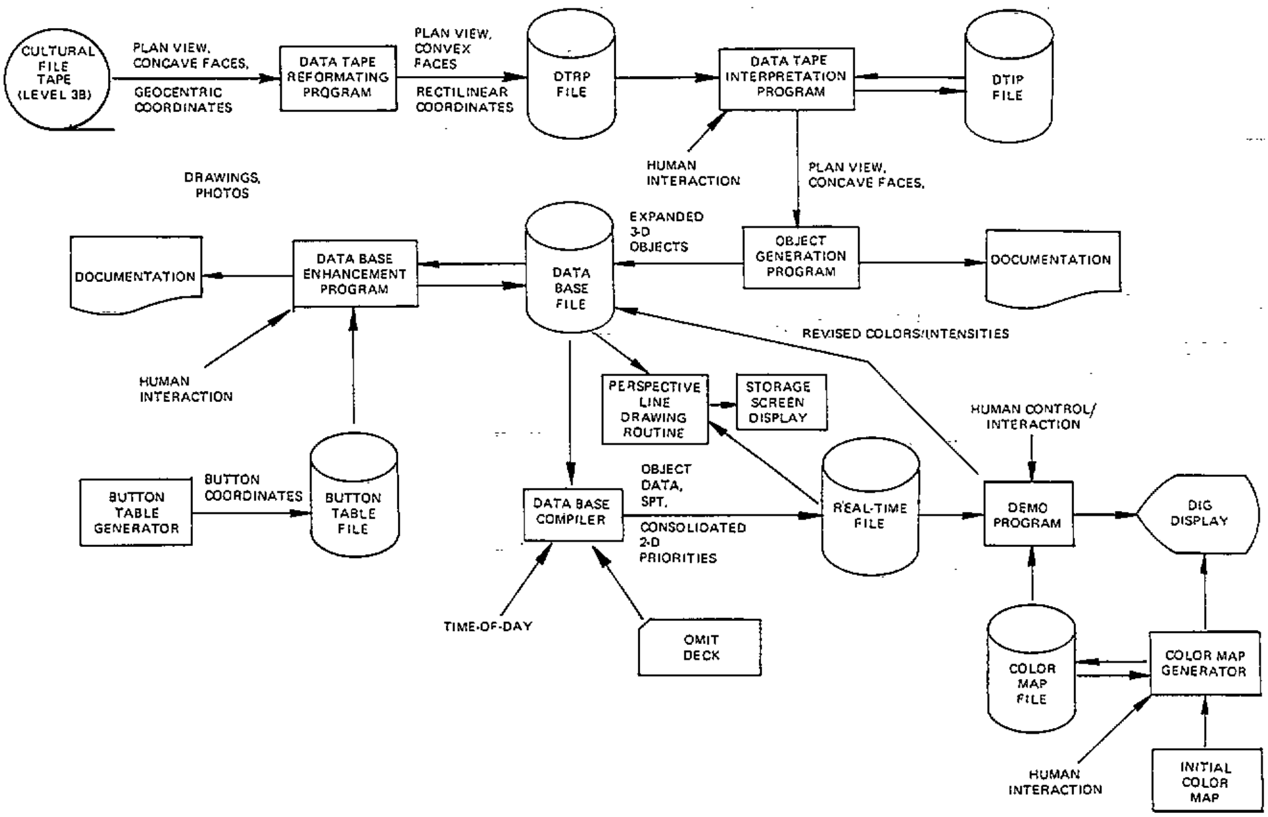

Figure 4. Data Base Generation Process

DIG system is capable of handling: of the 4,300 vertices in the Level 3B manuscript, approximately 2,400 are contained in the 145 3-D features. When these features are extruded from the ground, an additional 9,600 face boundaries will be created, for a total, including 2-D, of 13,900 face boundaries, without even considering those that are added to force convexity. Consequently, a considerable number of vertices must be eliminated in order to remain within the DIG processing capacity, and the remaining vertices must be chosen for efficient and accurate portrayal of feature outlines, as well as for subjectively important qualities such as symmetry and regularity. This process requires a high degree of human involvement, which is facilitated by the Data Tape Interpretation Program (DTIP) in conjunction with the storage screen display terminal.

Using DTIP, the data base modeler works on one feature at a time, first selecting it, either by referencing its FAC number or by positioning the graphic cursor anywhere within the feature boundary, and then modifying its geometry as required, using an orthographic plot of the feature or the entire manuscript. This modification may be done by deleting or moving existing vertices, or by adding new ones. An alternate mode allows complete respecification of a feature through the sequential placement of its defining vertices. In all cases, vertices are referenced or placed by positioning the graphic cursor and depressing the "space" bar on the keyboard.

On the 5- by 7-inch display terminal screen, a 4½-inch square area on the left is reserved for the graphic plots. The right-hand scale of the screen is reserved for the display of the menu, or listing of allowable actions at each step of the modeling process. Each action is identified with a digit (0-9) and is selected by activating the appropriate digit key on the keyboard. Additional display fields identify the name of the file being plotted from, and the scale of the plot. The use of the former field arises from the identical formats of the DTRP and DTIP (output) files; consequently, either file may be used as the starting point for modifying a feature description. The need for the latter field is due to the capability at various steps to redefine the extent of the plot, either by zooming in or out by a factor of two, or by arbitrarily defining new extents with the graphic cursor.

When the modeler is satisfied with the appearance of a feature (at least in an orthographic view), he may declare it to be "complete." At this point, the feature boundary is not necessarily convex. Since this is harmful from two standpoints (nonconvex faces and nonconvex objects are not allowed), steps must be taken to insure orthographic convexity. This is done automatically in DTIP by adding internal edges connecting selected vertices, so that the feature is divided into a set of convex faces. The result is plotted on the display screen to allow the modeler to determine whether the "convexitization" is optimum (i.e., minimum number of internal edges); he may wish to delete or move vertices so that fewer faces are generated, or he may wish to define the internal edges himself, a capability which has been provided because the present convexitization algorithm is not optimum.

When all is complete and satisfactory, the modeler then "saves" the feature. This passes the feature description to the Object Generation Program, after first verifying that the feature, if 3-D, does not overlap any existing 3-D features in the DTIP file. Were this to be so, overlapping objects would be generated; these are not allowable with the priority list technique.

OBJECT GENERATION PROGRAM

The Object Generation Program (OGP) is actually a subroutine called by the DTIP mainline, and after execution it returns to DTIP to allow work on additional features. The function of OGP is to convert the feature descriptions from the DTRP/DTIP format (which takes advantage of the orthographic nature of the vertex information to reduce storage) to a true 3-D data base file (DBF) format, after first expanding 3-D features by adding "walls" to the "roofs."

The DBF format (see Figure 5) is a hierarchical list structure, consisting of feature, object, face, vertex reference, vertex coordinate, and vertex normal beads (the latter only for smooth-shaded objects); each bead, or cell, consists of a contiguous (in memory) sequence of words which define the requisite information for that level of the structure, as well as pointers to the next bead at the same level and the first bead at the next lower level. This type of format lends itself to convenient addition, deletion, or revision of information.

In addition to the mechanical task of converting file formats, OGP also calls a series of routines which provide automatic hardcopy documentation of the feature, including objects, faces, and vertices. This documentation, is extremely valuable during later debugging.

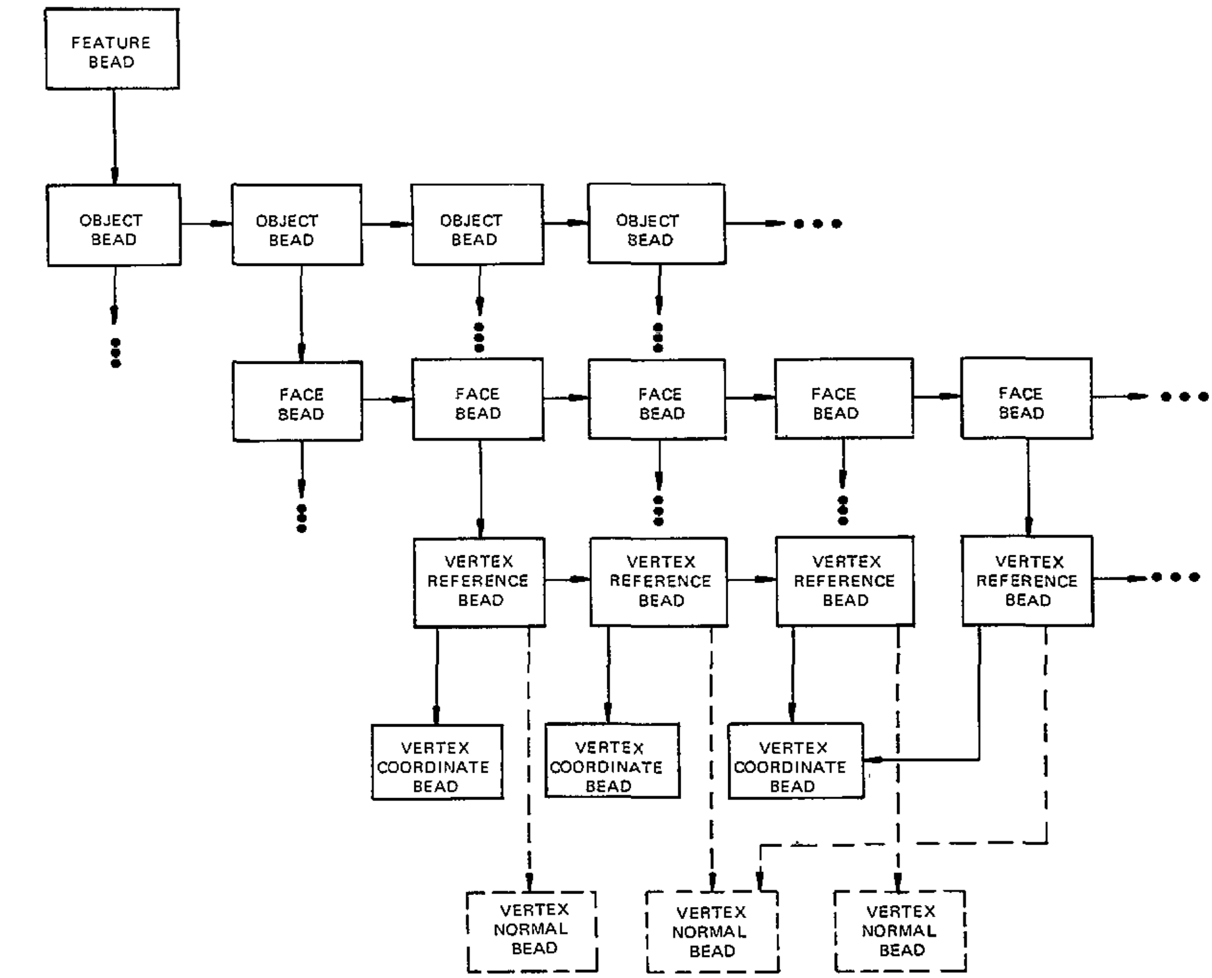

Figure 5. DBF Data Structure

DATA BASE ENHANCEMENT PROGRAM

Not all significant features can be extracted from the DMAAC DDB, and some of those that can are not well represented by simply expanding them in three dimensions. For example, the Las Vegas Convention Center dome is extremely recognizable in the real world, but not in the DMAAC data (Figure 6); the Landmark Tower, while it is represented as a separate feature in the DDB (Figure 7), would not, when mechanically expanded, allow the accurate portrayal of the actual structure. In cases like this, source material other than the DMAAC DDB must be used. The Data Base Enhancement Program (DBEP) allows the entry of this information into the data base, and is also the principal tool for the modification of the data base.

DBEP is controlled exclusively from the digitizing tablet, using predefined areas of the digitizing surface as "buttons" which

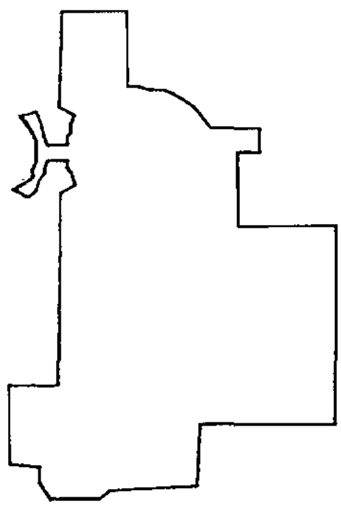

Figure 6. DMAAC Plot of Las Vegas Convention Center

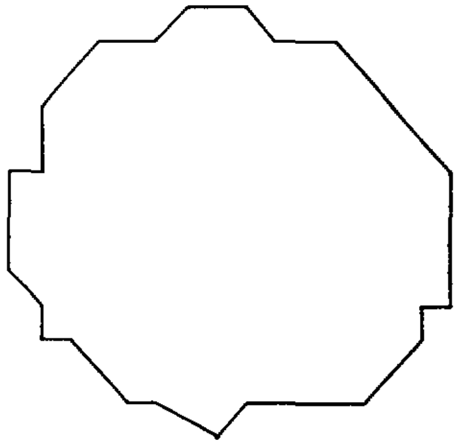

Figure 7. DMAAC Plot of Landmark Tower

the data base modeler may activate at the appropriate times. In addition, the tablet is also used as a pure graphic input device, allowing entry of specially formatted drawings. For 2-D features, these drawings depict a plan view of the faces to be entered. For 3-D features, both plan and side views are represented on the drawing, one above the other, as shown in Figure 8. To digitize a face from this drawing, corresponding pairs of vertices from the two views are entered in counterclockwise order around the face. A perspective view of the solid *figure obtained from the drawing in Figure 6* is shown in Figure 9. While less elegant than some other approaches for 3-D data entry,[9] this scheme was nonetheless easy to program and proved quite workable in practice.

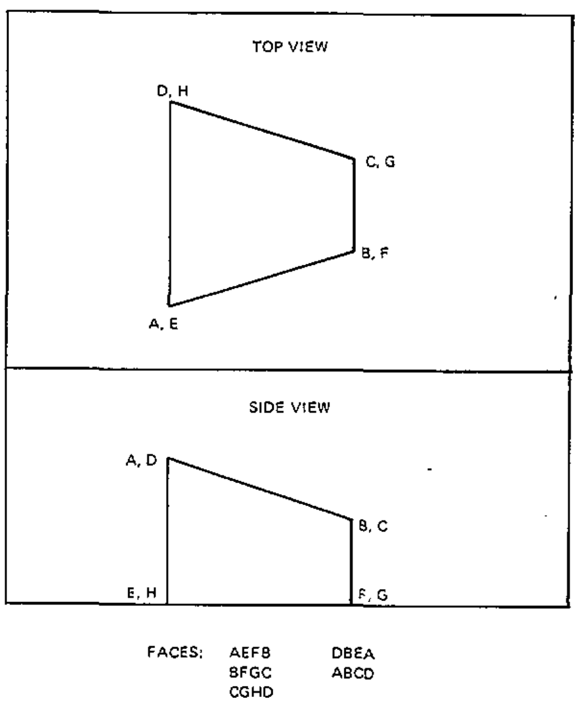

Figure 8. Three Dimensional Data Entry

Aside from the ability to enter new features, DBEP also provides feature modification and *manipulation* capabilities. For example, existing faces, objects, or features may be translated, rotated, or both; face colors may be selected; vertex normal vectors may be automatically computed as an average of face normals, or may be specified numerically; duplicate copies or mirror images of existing features or objects may be created; and existing features, objects, faces, or vertices may be deleted from the data base. Extensive error checking is conducted at all steps to insure object and face convexity and face planarity, among other conditions. At *any time partial* documentation may be requested for an intermediate check on data validity, and automatic documentation is generated upon saving a feature in the DBF. The documentation routines used are the same as those called by OGP, and also provide a check for linear separability of objects within the feature. In all, DBEP provides a powerful tool for data base generation and modification.

DATA BASE COMPILER

Prior to being loaded into the DIG ADB core, the data base must be converted from the easily updatable list format in the DBF to the highly compressed format required by the

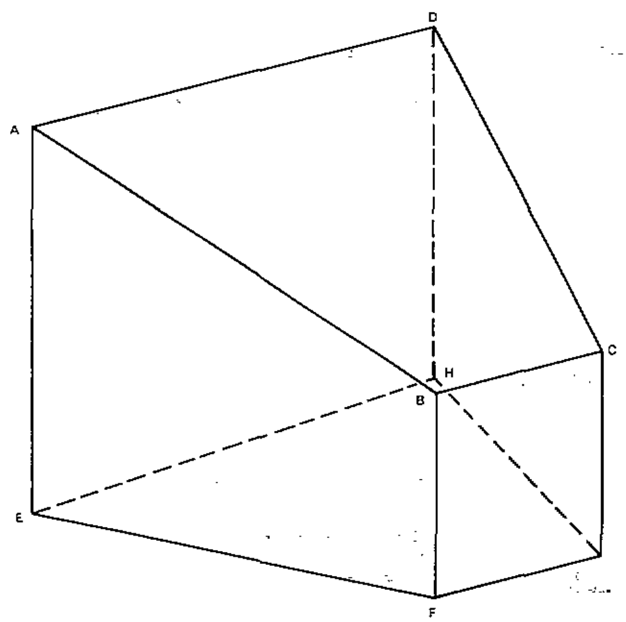

Figure 9. Perspective View of 3-D Solid

DIG processing hardware, a function performed by the Data Base Comiler (DBC). In the process, default face colors and intensities must be assigned for those faces extracted directly from the DMAAC DDB, which contains no color information. At present, this assignment is based only on FID number; where the modeler wishes this assignment to be overridden, he must manually enter a color/intensity combination with DBEP. Intensities for other sensors (FLIR, LLLTV) can also be derived during this stage of the processing.

In addition to the data conversion, the DBC performs two other important functions: 2-D priority consolidation, and separating plane generation. Since 2-D faces are all in the ground plane, they can all be assigned the same priority number, provided they do not overlap, without any deleterious effects appearing in the display. However, some 2-D faces purposely overlap others to minimize the number of face boundaries required. 2-D consolidation seeks to conserve priorities by collecting 2-D faces into the minimum number of priority classes, where all members of a class are assigned the same priority number. In the Las Vegas data base, 35 2-D features were reduced to only four priority classes, thus allowing a greater number of 3-D objects.

Whereas the priorities of 2-D objects can be assigned during the data base compilation, those of 3-D objects depend on their position and arrangement relative to the observer. Consequently, computation of these must be deferred to real-time. The concept of separating planes allows the convenient precalculation of object priorities using the modest facilities of the general-purpose computer. The computation of the separating planes, and the tree structure which defines their relationships to each other and to the 3-D objects, is performed off-line by the DBC using a refined trial-and-error technique.

To allow for experimentation with different versions of features or objects, the DBC accepts as input a card deck that defines which features and/or objects to omit from the current compilation. In addition, the time of day must be specified to allow the computation of face intensities based on face orientation and calculatea sun position.

PERSPECTIVE LINE-DRAWING ROUTINE

Final proofing of data base geometry, prior to actual display with the DIG, is done through the generation of perspective line drawings, with hidden-line removal based on the separating plane/priority list approach. In order to allow convenient user specification of the viewpoint position and orientation for generating the line drawing, the Perspective Line-Drawing Routine first creates a hard-copy orthographic map of the data base on the storage screen display, using just the objects obtained from the last previous data base compilation. The user then tapes this map to the digitizing tablet, locates the map by digitizing two corners, and then digitizes the positions of both the viewpoint and the point being viewed, first in X-Y directly on the map, and then in Z using a scale along the side of the map. The routine then computes object priorities using the separating planes previously compiled, and displays the objects on the 4010 screen in priority order. A sample of the output is contained in Figure 10.

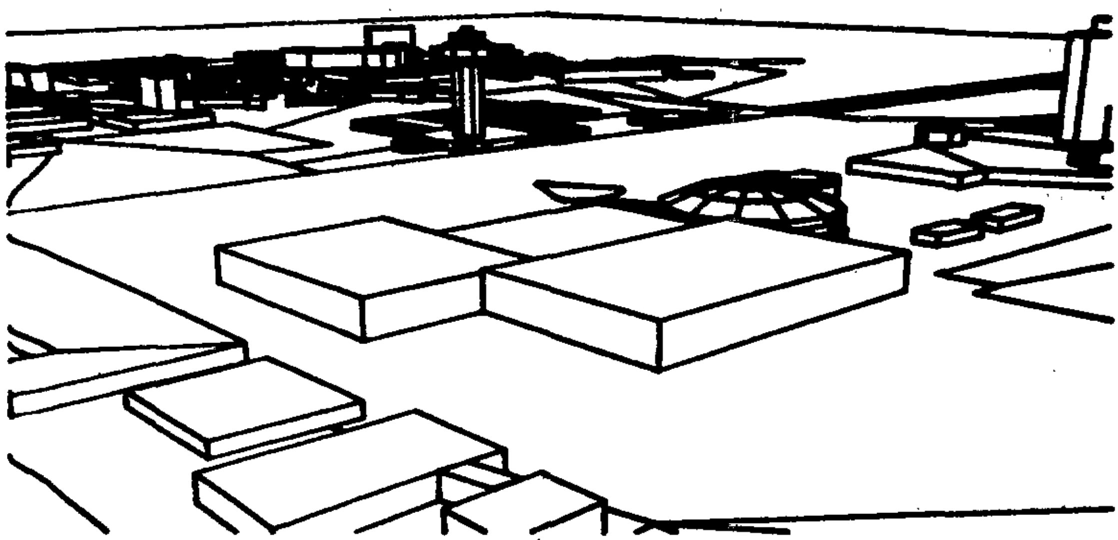

Figure 10. Perspective Line Drawing of Las Vegas Data Base

DEMO PROGRAM

The proof of the process is actually viewing the data base in real time on the color display, under the control of the Demo Program. The Demo Program performs three primary functions:

1) Aircraft emulation - using joystick and switch inputs, observer position and attitude are calculated according to simplified aerodynamics;

2) Priority calculation - priorities of 3-D objects are computed by traversing the separating plane tree;

3) DIG communication - commands and data are transferred to the DIG processing hardware using standard communication subroutines.

Aside from the viewing capabilities provided by the above functions, the Demo Program also provides a powerful data base debug and modification capability. By enabling the Color Modification portion of the program with a switch at the control console, a user can cause a small square cursor to be generated in the picture. This cursor is responsive to joystick motion, and can be positioned over any face or object. When a button on the joystick handle is depressed, the numbers of the feature, object, and face overlaid by the cursor are printed on the teletypewriter. Then, by means of slew switches on the control console, the user may dynamically change the color and/or intensity of the object/face while observing the effects on the display. Future enhancements of this capability may include the ability to adjust vertex normals for smooth-shaded objects.

ANCILLARY PROGRAMS

Two ancillary, but important, programs have not yet been discussed: the Button Table Generator and the Color Map Generator. The Button Table Generator allows the definition, using the digitizing tablet, of the coordinates of buttons used by DBEP. In addition, it allows the checking and modification of button locations, and the addition of new buttons. The output of the Button Table Generator is stored on a disk file accessible to DBEP.

The Color Map Generator allows the specification of the 64 possible colors which comprise the permissible color space at any one time. This specification takes the form of a six-bit value for each of the red, green, and blue (R, G, B) components of the desired color. The specifications for all 64 colors are contained in a random-access semiconductor memory in the video generator, called the color table or color map. The color map can be thought of as a catalog of paint chips, from which the data base modeler can select the colors of faces and objects. Each chip has a number, from 0 to 63, which is the address of the color map entry for that chip; this number is what is carried along in the data base to represent color, and during the video generation process, a table lookup is used to transform from this number to the actual CRT drive voltages. The color table is loaded during the initialization phase of the Demo Program from the color map in the Color Map Disk File, which is generated in turn by the Color Map Generator.

It had initially been thought that the color map entries would be empirically determined using the actual display. However, a superior, purely analytical approach was developed which allowed the determination of color map entries directly from the hue, value, and chroma information supplied with each color chip in the Munsell Book of Color,[10] which was used as the basis for color selection. This determination took account both of the color CRT phosphor capabilities and the transfer function of the D/A converters and gamma correction circuitry in the video generator. The results achieved with this approach were outstanding, and no empirical correction was required to match the original Munsell colors within human discernibility limits.

RESULTS

As indicated, the DBGS was used on the DMAAC Level 3B manuscript of downtown Las Vegas. Using DTIP, the original 4,300 vertices were reduced to 1,783, the result of eliminating about 60 features, retaining 50 in their original form, and using DTIP to "outline" another 50. The orthographic data base, after all work with DTIP, and including edges added for convexity, is shown in Figure 11.

After this step, drawings were prepared and digitized for another 13 features, including the Convention Center dome, Circus Hotel, Landmark Tower, and Hilton Hotel. The resulting 3-D data base, consisting of about 5,300 face boundaries, is shown in Figure 12, which is reproduced from a photograph taken from the DIG monitor. Additional photos of this data base are contained in Reference 11.

The versatility of the DBGS, particularly DBEP, is illustrated in Figures 13-16, which are DIG photographs of data bases for the space shuttle orbiter, a terrain-following range, and day and night versions of the San Jose Municipal Airport. These data bases started with drawings, so DTRP and DTIP were not used.

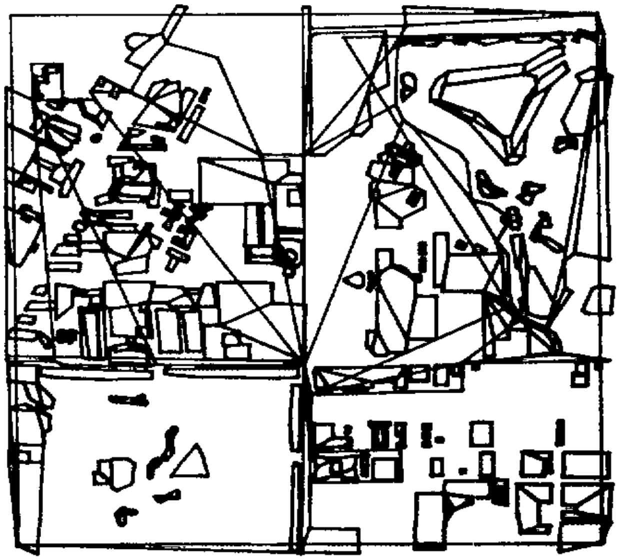

Figure 11. Las Vegas After Modification with DTIP

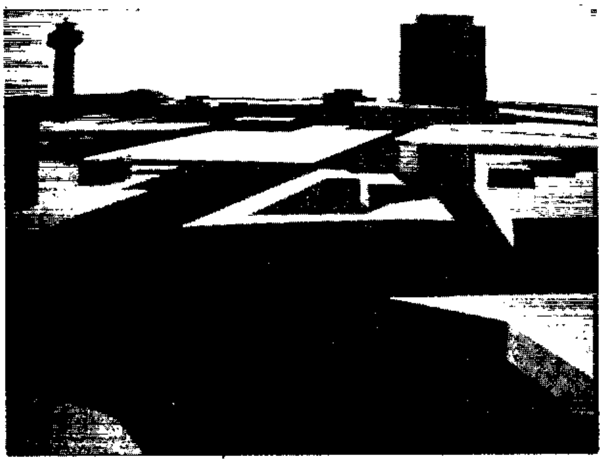

Figure 12. Las Vegas Data Base

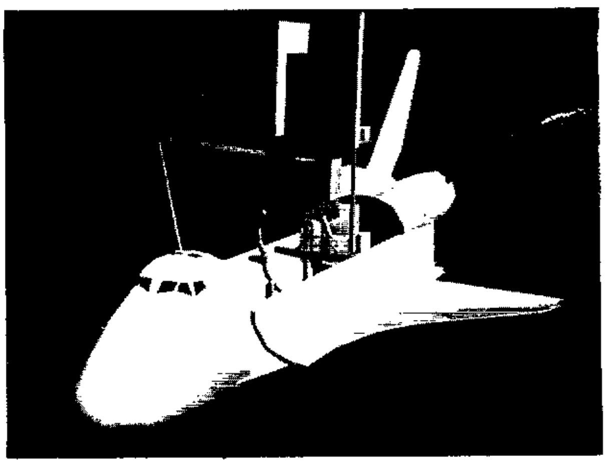

Figure 13. Space Shuttle Orbiter

FUTURE DEVELOPMENT

Many shortcuts were taken in the design of DTRP and DTIP to take advantage of known characteristics of the Level 3B manuscript, including geographic location and maximum number of vertices. The design of these modules should be generalized to allow handling of arbitary manuscripts. This will include matching of features from adjoining manuscripts, and from manuscripts at different levels of description. Some automation can be incorporated into the program to, for example, eliminate redundant collinear (or nearly so) vertices based on a user-supplied tolerance. In addition, data may have to be organized differently to satisfy the requirements of dynamic data base retrieval programs, which will probably be inevitable in any nontrivial simulation application. These changes will of course propagate through the other programs of the DBGS, principally the DBC.

CONCLUSION

The data base generation system described herein demonstrates the feasibility of semiautomated methods for the production of cultural DIG data bases for visual simulation. This approach depends on a pre-existing source of digitized data covering a large geographic area, and is made economical through increased automation of decisions and inclusion of visually significant information (e.g., colors, intensities, textures) in the source data. While human intervention in the data base generation process will never be totally eliminated, it can be reduced to those areas in which human facilities are superior to those of machines: pattern recognition, perceptual judgement, and determination of importance to training.

Figure 14. Rolling/Mountainous Terrain

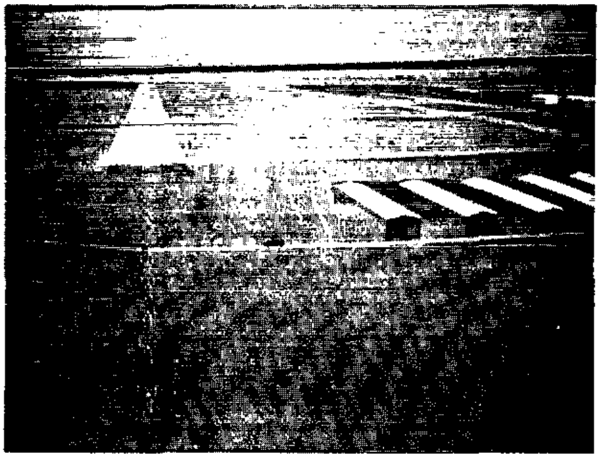

Figure 15. San Jose Municipal Airport (Day Version)

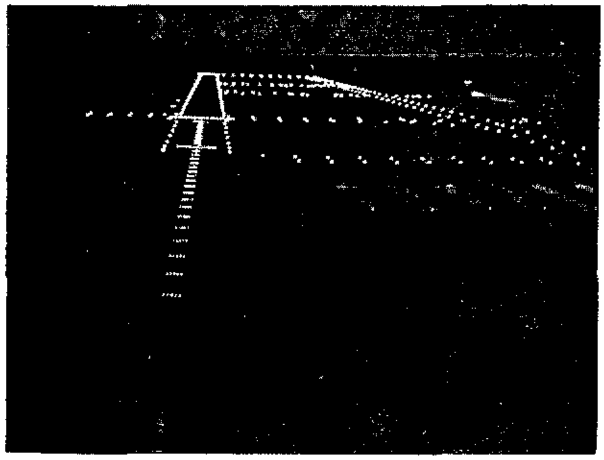

Figure 16. San Jose Municipal Airport (Night Version)

REFERENCES

REFERENCES

1. Purchase Description and Prime Item Specification for A/F37A-T36 (FB111A) Mission Simulator Visual System, MME-PD-75-8, Department of the Air Force, Hill AFB, Utah, December 1, 1975.

2. W. Marvin Bunker, "Computer Generated Images Simulate Infrared and Low Light Level Television Displays," AIAA Visual and Motion Simulation Conference, April 26-28, 1976.

3. I. E. Sutherland, R. F. Sproull, and R. A. Schumacker, "A Characterization of Ten Hidden Surface Algorithms," ACM Computing Surveys, Vol. 6, No. 1 March 1974.

4. R. Schumacker, B. Brand, M. Gilliland, and W. Sharp, Study for Applying Computer-Generated Images to Visual Simulation, AFHRL-TR-69-14, Air Force Human Resources Laboratory, September, 1969.

5. T. Hoog, R. Dahlberg, and R. Robinson, "Project 1183: An Evaluation of Digital Radar Landmass Simulation," Proceedings of the Seventh NTEC/Industry Conference, November 19-21, 1974.

6. Production Specifications (Guidelines) for Off-Line Digital Data Base, Air Force Project 1183, Digital Radar Landmass Simulator (DRLMS), Defense Mapping Agency Aerospace Center, St. Louis, Mo., September, 1974.

7. A. J. Grant, "TRIG - An Algorithm for Generating a Planar Terrain Elevation Model for DRLMS," Proceedings of the Eighth NTEC/Industry Conference, November 18-20, 1975.

8. Dr. Robert T. P. Wang, "A Grid-Based Variable Resolution Data Base for Real-Time Visual Training Systems," Proceedings of the Eighth NTEC/Industry Conference, November 18-20, 1975.

9. Ivan E. Sutherland, "Three-Dimensional Data Input by Tablet," Proceedings of the IEEE, Vol. 62, No. 4, April, 1974.

10. Munsell Book of Color, MacBeth Color and Photometry Division, Kollmorgen Corporation, Newburgh, N.Y., 1973.

11. Murry Shohat, "From Submarine to Satellite - Diverse Applications for Digital Image Generation Techniques," Proceedings of the Eighth NTEC/Industry Conference, November 18-20, 1975.

ABOUT THE AUTHOR,

MR. ARTHUR P. SCHNITZER is presently a Manager of Visual Software Engineering at The Singer Company, Link Division, Binghamton, New York. Since joining Singer in 1965, Mr. Schnitzer has been responsible for the development of software for a variety of visual simulation systems. He has most recently been responsible for the development of data base generation and retrieval software for digital image generation systems. He is a member of the Association for Computing Machinery. He has the B.S. degree in mathematics from Union College, and has completed course work toward the M.S. degree in systems and information sciences at Syracuse University.