DODGING THE TREES AND BUSHES: CURRENT NOE SIMULATION

B. James Voorhies
Michael A. Cosman
Evans and Sutherland Computer Corporation
Salt Lake City, Utah

ABSTRACT

Many previous rotorcraft nap-of-earth visual simulators were designed by the "seat of the pants" method: infant technologies mated with "best guess" estimates of what a visual simulation scene should contain. Experience with the successes and weaknesses of these earlier systems (many still in valuable active use) provided the right questions, and recent empirical quantitative studies have offered first and second generation answers. Unfortunately, the technology's solutions have too often been, "This is as good as we can do, and so we'll prove it's good enough." It is now possible to approach the problem from the researchers point of view. Maturing hardware and emerging software technologies have made possible the design of a system that provides the content demanded by the research, not just to marginal but to desirable levels of performance.

The nap-of-earth visual simulation project had as a practical objective tailoring a visual data base designed for rotorcraft to the requirements of a comprehensive research study. The projected performance envelope was designed to be nap-of-earth and contour flying from five feet up and from hover to 100 knots. Empirical issues to be addressed in the implementation included: mix of 2D and 3D cueing, and total cue densities to provide optimum visual flow; elimination or minimization of negative cueing; value of terrain fidelity versus dense generic scene content; maximization of detail in the aircraft performance envelope; and scene reality versus training value. Other perceived deficiencies of some earlier systems that are addressed by this system are multiple, properly occulting, dynamic ground and air threats, and special weapons effects (cannon, rockets, FLIR, etc.) In addition, consideration was given in the structuring of the data base to provide maximum flexibility as new research and experience dictate modifications. What has evolved is a good example of visual flight simulation specifically designed for the nap-of-earth regime.

INTRODUCTION

The NOE simulator system is intended for research of advanced rotorcraft flight and control issues. The system includes a computer generated visual scene which is projected onto a dome, with both cab and dome to be mounted on a motion base.

The intent of the simulation capability is to provide visual cues to a crew for landings at unprepared sites, precision hover, nap-of-earth flight (weaving between the trees as well as flight just above tree height), air-to-ground weapon delivery, air-to-air combat, and autorotation landings. These tasks are to be simulated under a variety of lighting and weather conditions. Consequently, a number of special effects were requested to provide a more realistic simulation environment.

The NOE data base is designed to be displayed on an Evans and Sutherland CT5A image generator with an aggregate field of view of 140 degrees horizontal by 60 degrees vertical, with image resolution of approximately 3.5 arc-minutes per pixel or line. The imagery is designed to be provided by three light valves, and will be projected onto the inside of a 20 foot diameter dome. Each of the three display channels covers a 46.8 degree horizontal by 60 degree vertical field of view, allowing some overlap at the channel boundaries for blending. The system computes motional updates at the 50 Hz field rate to avoid image breakup under high motion rates. The system configuration and fields of view were chosen specifically to address the needs of the rotorcraft simulation mission and the data base was tailored to these specifications as explained later.

The CT5A system can display 3750 polygons, with up to 1500 polygons within each channel. Both th 3750 and 1500 polygon limits have been shown to be conservative, and the system exhibits significant "headroom" in its ability to keep average loads at these levels, rather than just momentary highly tuned peaks. CT5A has no sensitivity to edge crossings per scan line.

THE DATA BASE

Research as a Basis for NOE Design

Part of the source material used was a copy of John B. Sinacori's report, "Research and Analysis of Head Directed Area Of Interest Visual Systems Concepts".[1] The conclusions and recommendations of the study become the design goals of the NOE project. Some important points made in the study are:

1. Nap-of-earth "flight requires a tradeoff between masked time and speed". "To the pilot . . . the primary visual task is to search for and find suitable masked areas along the route, while maintaining small but safe clearance from the immediate surroundings." His skill and his familiarity with the terrain determine how fast and how low he can travel without being seen or hitting something.

2. The pilot spends 40% of his time watching out for things he might hit, 40% looking ahead to where he is going next, 10% looking for navigation cues, threats, and targets, and 10% for everything else (including 5% of his "seeing time" blinking).

3. Motion perspective, the interpretation of optical flow within the vision field to determine time and point of impact, is the most important cue

mechanism for nap-of-earth flight. With enough imagery of the proper kind, motion perspective "can provide a depth map directly by observation of several objects in the visual field". This awareness, augmented by other perceptual mechanisms like familiar size and apparent size, allows nap-of-earth flight over unrecognizable terrain on the first encounter.

4. The *minimum cue density needed for nap-of-earth flight* can be mathematically determined. The research recommendation for this cuing is one texture item for every eye height above ground for a random texture pattern on a flat surface. After this level is exceeded, a selection of vertical (3D) objects will improve pilot height holding ability and reduce his work load. Lateral spacing for 3D objects should be at least one for every five eye heights.

5. Sufficient ground truth is required to provide realistic visual and topographic complexity and a usable "nap" of the earth within which the craft can remain masked. Terrain skin complexity should be as high as possible, with about 1000 polygons per square nautical mile the goal. Modeled terrain should match real world terrain to within plus or minus 20 feet vertically.

Predicting IG/Data Base Interaction

In implementing these design concepts, the challenge to the data base designer is to push the image generater to its limits, so he gets the densest images, the most mission relevant cueing, and the *most impressive visual display, without exceeding* the capacity of the machine to execute quality imagery. He must have tools to predict image generator performance with a given input data base, or some aspect of the system may be driven past its limits and the picture quality may deteriorate.

We have developed a mathematical methodology for designing data bases which allows the prediction and control of the way the data base loads the various parts of the image generator.(2) This methodology properly accounts for the complex integrated interaction of all the data base elements, and allows us to design and tune particular types of features and cues to particular mission requirements. It is based on the treatment of "*density domains*". *In a visual environment which is* constructed so that simple versions are used for distant things and more complex versions are used for those same things when we are near them, we can think of the resultant environment as it is "seen" by the data base management portion of the hardware. Our world now consists of an expanding set of concentric rings which surround our viewpoint and move with us throughout the data base. Things which lie in the outermost rings are drawn with very simple geometric representations. Nearer things, those lying in the middle rings, are drawn with more complex geometry. The rings closest to us are populated with the most complex things. Throughout this region of rings, the density of scene details also varies to keep most of the active items near us, and use few of them in the *distant rings. The overall structure of these* density domains may include hundreds of levels of detail, and behaves in a very smooth, continuous fashion to ensure that most of the system capacity is used to draw scene details which are near us, and consequently visually significant, while avoiding the expenditure of IG capacity on tiny elements out near the horizon.

The NOE data base presents a typical and interesting example of how such an environment is designed and dealt with. First, a rigorous analysis of important domains of interest was made and a *polygon budget established. The polygon budget* uses as its total the constraining limit in the image generator. In the NOE CT5A system the "per channel" specification is the limit: 1500 polygons on screen per channel. Targets and special effects are needed everywhere in the data base. Experience has shown that they can require up to 200 polygons, leaving 1300 polygons for everything else. How much of this remaining amount should go to 2D details, and how much to 3D? A first cut would be about 40% for 3D and 60% for 2D. The budget for 2D things is now about 800 polygons. We used half for the terrain skin and half for textural decoration, giving us 400 polygons per channel to represent the ground out to the visibility limit of 6000 feet. A quick analysis of the area visible in a 46 degree slice of a 6000 foot pie shows that this is *consistent with our ground truth goal.*

The polygon budget for 2D textural embellishment must now be considered. How far away is texture needed to support contour flight? Sinacori suggests about 800 feet, based on operating envelope analyses. Again we compute the area of this "pie slice." What is the area of densest texture needed for nap-of-earth flight, where the pilot's attention is focussed on his "area of impact?" Given typical *nap-of-earth flight* speeds and the projected performance envelope of a helicopter, 400 feet seemed appropriate. A first analysis gave 50% of the remaining 400 polygons to low level texture and 50% to high density texture. Given the size of the ground polygon established from the analysis above, how many texture elements could be used on *one terrain skin poly to produce a total of 200* within a 400 foot pie slice? The analysis says that about 160 can be put on each, giving us a ground textural patch about every 18 feet, and an edge every nine feet laterally. A test data base confirms that this is achievable.

3D feature analysis is likewise done with a rigorous adherence to polygon budgets and other load factors imposed by the hardware. The end product *is a test data base that is virtually guaranteed* to work the first time, and from which various artistic concerns can be addressed and polygon budget tradeoffs made. At this point the CT5A Simulator software allows the designers to obtain accurate readouts on how the data base is doing, without tieing up large amounts of IG time.

DESIGNING FOR FLEXIBILITY AND CHANGE

The NOE data base is designed to be used for engineering work, and is likely to excite interest in data base modification more than a system whose purpose is to train a succession of pilots to do the same small set of tasks. Engineers delight in thinking up new situations to test new ideas that had not been conceived at the time the original *system was designed. Thus consideration was given* to making the NOE data base flexible and easily modifiable. This flexibility was achieved by careful top level design strategies, and by the nature of the particular modeling tool environment in which it was built.

The data base structure is extremely modularized. Each square kilometer of the eight Km x eight Km NOE section, for instance, is a basic unit that edge matches with its proper geographic neighbors. These squares are assembled into groups of four, four of those into a group of 16 squares, and four of those into the 64 square Km basic NOE terrain segment. The outer perimeter edge boundaries of this segment were modified to tessellate the 64 Km piece with itself. This allowed us to build a larger chunk of world that would extend quite far in all directions.

The one Km squares are subdivided into 400 meter pieces and again into 100 meter triangles, yielding a final terrain complexity of about 686 polygons per square nautical mile. Each triangle can actually be thought of as a vertical volume of space which has a triangular footprint in the XY plane, and all of them have the same size and shape footprints which all fit together contiguously. Within each volume an actual terrain skin polygon will "float" at the *proper elevation and slope, joining properly with* its neighbors to form a continuous three dimensional terrain skin. The actual filling of each triangular region with its proper contents is a separate process, and is what transforms the top level structure into a real data base. This allows changes in the terrain skin shape and decoration to be done with no impact on the top level structures.

The modeling environment itself promotes a great degree of flexibility. A variety of text based and interactive graphical tools are used to create the bits and pieces of the data base.[3] These scene features are developed independent of their eventual visual context, and become part of a growing library of components, much as subroutines of a large program may be developed and compiled. They are "connected" to specific places and uses within the data base by symbolic references contained in the organizational data. The final step in making the loadable data base is analagous to the linking of a large program from independently developed and compiled subroutines. Such scene features are positioned in real time by the "instancing" capability of the IG.

This approach has several interesting advantages. First, the development of a large data base may *proceed on many fronts in parallel, minimizing* overall schedule time. Second, features may be modeled independently of their final context, and need be modeled only once per data base, instead of once per use. Also, modifications or improvements to a scene element, once made, are quickly reflected throughout the data base everywhere that element is used. This makes substitution of a tree for a rock (or a haystack, or whatever) a simple matter of changing the global reference and a deciduous tree becomes a rock, everywhere that feature is used. Once the organizational data has been created, the process of swapping one feature for another takes very little time, and results in a major visual modification throughout the simulated world.

This approach also reflects a secondary objective of the data base development effort, which was to provide to the user a powerful modeling technology which would encourage and support experimentation with the data base. Achievement of this goal may well be one of the best features of the data base, since it allows the capabilities of the system to easily grow beyond the present perceived needs of the users. As new insights into nap-of-earth simulation problems are developed, new ideas will be tried and the NOE data base of 1986 may bear little resemblance to the one initially developed. It is hoped that this evolutionary process will have been eased by some of this basic design planning at the start of the project.

Instancing and Basis Sets

Much of the "scene complexity" in a nap-of-earth data base is of a very similar, repetitive nature: trees, bushes, rocks, etc. that are not important as individual unique cues but rather as elements in a matrix that provides visual flow, height cues, scale cues and parallax information. The pilot is seldom told to "turn right at the lonesome pine", but rather fly low and fast over or between the trees on a hillside. The idea of instancing is to reuse these features, modeling some small number of unique items and using the organizational strategies of the system to distribute them, thus *saving memory, modeling time, and disk paging* effort. For non-specific detail, this method of achieving dense visual geometry without overloading the storage ability of the image generator works well.

If a tree can be reused, why not a section of hillside made up of reused trees? Imagine a mountain constructed of stacked building blocks, each designed to visually match its neighbor. If a consistent size block is used a mountain of any size and shape can be built and the ground truth is only limited by the size building block chosen. Imagine looking down on a grid made up of triangles, not squares. The corners of the triangles are positioned laterally at regular intervals, 100 meters. The surface of the mountain made with stacked blocks before can be skinned quite accurately with these triangles. If the Z difference between grid points is constrained to be a small multiple of some basic elevation change (50, 100, 150, 200, 250 feet for example) and if the terrain to be modeled has few excursions where there is an elevation change of more than 250 ft in Z for 100 meters in X or Y, a very finite set of puzzle pieces could be fitted together to create the terrain. The set of puzzle pieces becomes a terrain basis set.

Superimposed on the terrain basis set triangle (which becomes the ground polygon) is a 2D collection of texture decoration polygons, and a collection of 3D items: instanced trees, bushes, rocks, fence posts, etc. This whole structural piece can then be instanced, positioned by organizational information to its position in X, Y, and Z on the skin of the mountain. We can ensure that its edges will match up with its neighbors and skin the mountain. Only a limited number of individual pieces need be modeled or stored in memory. Each piece will be used many times and in many different places. The possible apparent geometric complexity is multiplied tremendously. With properly chosen grid spacing and elevation gradients, the world can be accurately modeled by using and reusing the small set of generic pieces.

Basis Sets for Specific Features

Once the basic contours of the terrain have been modeled, enhancement to any desired level can be begun. In the Hunter-Liggett area, peculiar geographic features were found to be significant visually and navigationally. Streams had sand bars at curves, and trees grew close to the water but not in the open fields around. Some gully areas had this same effect: trees in the washes but not on the hillsides on either side. A reservoir was included in the area as were random small hills too tiny to show accurately on the topographic maps. Important cultural details included a two lane paved highway, an extensive network of dirt roads and trails, telephone lines, farm ruins, corrals, fence lines, small dikes and bridges, and some stock watering tanks which are used as location fixes. Consequently, several additional basis sets were developed to substitute for or add to the 2D and 3D general terrain sets originally developed:

Two lane paved highway with lane stripes and shoulders, for use in the flat terrain areas.

Dirt road and trail set to be used in flat terrain.

Tree clump set to replace random vegetation in selected areas.

Gully basis set to replace random distribution of trees in selected areas.

Telephone poles with lines to augment roads or cross flat plains as indicated on the map.

Water texture set to decorate the reservoir and lake surfaces.

Less dense 2D texture pattern where extra 3D detail was needed.

Small hill basis set for hills of less than 100 meter overall size, which included 2D and 3D components.

High tension power transmission tower set with wires.

In addition, a collection of reused 3D features was developed: a 100 meter diameter group of trees, an orchard, a row house, farm house and barn. Once these individual parts and basis set collections of components had been modeled, selecting combinations that closely matched the details specified by the topographic maps and source photographs was a very straightforward process.

On Art and Training Functionality

One school of thought expressed by the user community is that more realism in scene content gives more transferable simulator experience. Another school believes that a checkerboard nap-of-earth data base can teach low level flight skills as accurately as any other design with much less modeling effort and expense. The jury is still out. There are some observations to be made, though. Some kinds of visual information are more relevant to the rotorcraft training experience than others. For example, when practicing height holding skills, the pilot looks for horizontal cues that are size correlatable to an altitude range: a ladder, a multi-story building, a pole with one meter horizontal alternating black and white stripes, etc. Given the limited capacity of image generation systems, should only relevant cues be included or is this "cheating", artificially enhancing the environment? Should the pilot need to seek out more sparse relevant cuing from a non-relevant matrix, i.e. work for his cues, or are the cues so all pervasive in nature and so easy to spot that emphasizing the relevant cues in simulation will not produce false training results? These questions are not answerable yet: the NOE simulator system is targeted to study some of them to better design the next generation data bases. At present we design using intuition and studies like the Sinacori paper.

Decisions have been made on some of the issues. Research will need to verify whether these decisions have truly resulted in a more functional nap-of-earth simulation. We can summarize some important things which have been learned, though:

1. Models should be as simple as possible and still give target believability and recognition at the relevant distances. More "simple" things provide better cueing than fewer "complex" things. Also, level of detail changes should be as subtle as possible while eliminating as much non-relevant information as quickly as possible.

2. 2D polygon budgets give more cueing per polygon expended than 3D, so more system resource should be devoted to 2D cueing. Such texture should be subtly colored so that the visual streaming of 2D cues is perceived but the pilot is not distracted or annoyed by its behavior. The optimum solution is to provide much of the needed cueing with hardware generated texture, leaving the bulk of the polygon capacity to address the need for 3D cues. Since the inception of the NOE program, hardware texture of both the intensity modulation and contour modulation types has become a CT5A system option.

3. 3D cueing is most effective when its density is roughly uniform. Groves of trees and bare fields are more realistic but a tree, bush, or rock every seven to ten meters is easier to fly over. The automatically generated areas of the data base have a random appearing 3D matrix of decoration: big trees, medium trees, bushes and/or rocks. More exactly modeled areas may have more tree groupings and bare areas so that both patterns are available for experimentation and pilot testing.

4. Hills and geographic features should be realistic enough to not detract from the illusion of reality. Pyramidal hills, for example, though less expensive in system performance are costly in unbelievable simulation. Likewise more complex trees, detailed enough to not detract from the illusion significantly are thought to be worth the extra polygon cost.

5. By properly color tuning items, they can be gracefully removed from the scene earlier. The difference in contrast between tree and hillside or hillside and sky, if reduced, allows the tree to fade into hill color and the hill to dissolve far quicker than would be possible if the actual colors of nature or cultural artifacts were shown. In this sense, the data base designer is making a conscious effort not to mimic reality in

order to make polygon saving transitions more subtle and less distracting.

6. The system resources should be used in producing enough basic patterns so that the "orchard effect" of rows of repeating geometry is minimized. Patterns should be selected randomly as basis set pieces are assembled. Since map or DMA tape data does not have individual tree placement information, this approach is as true to life as the resolution of the most common source data. A thorough understanding of cue needs is essential to design models that are optimally functional as well as artistic. A paper given in a simulation conference last year reported densities of hundreds of polygons per square mile as adequate for low altitude flight training, presumably because that level of density taxed the limit of the simulator system discussed there.[4] The NOE system, using instancing, displays effective nap-of-earth polygon densities in excess of three hundred thousand per square mile. The Sinacori study indicates that these densities should be adequate for a functional nap-of-earth data base. The terrain, texture patterns and models have been designed with the best available research data. Regular feedback from civilian and military pilots as they use the data base will provide the simulator experience to validate the result.

DATA BASE IMPLEMENTATION

Early in the project it became clear that we could produce a much larger and more ambitious data base within the NOE system constraints than was initially planned. As a result, an area in Monterey County, California, just south west of the Hunter-Liggett Army Airfield was selected as the best region to model. The area of greatest interest was an eight Km x eight Km. square covering the Stony Valley and a portion of the Nacimiento River Valley. As source material we obtained DMA topographic contour maps and photographs of the area taken during a drive around the improved roads in the valleys. After the program had progressed to a suitable point, a day of helicopter flyover was arranged for a few of the design team during which we took extensive video tape and still photographs.

The experience proved to be most valuable, helping us to visualize the importance and impact of relevant cues such as dirt roads, stream beds, cultural features, etc. and better design the models used in the data base.

DMA tape data was not available when the implementation phase of the project commenced, so elevation data for the area was extracted manually from the DMA contour map. (The tools and data are now available to assemble the information directly from DMA tape.) A regular 100 meter grid was selected as appropriate based on the Sinacori study. An elevation gradient change of 200 feet maximum in 50 foot increments across any edge of this 100 meter grid was used. This set of slopes was analyzed to be able to duplicate the ground truth to at least the resolution of our source maps or to DMA tapes when they became available. The 2D texture pattern was the most difficult to develop, requiring several levels of scale and graceful transition between each. Depending on the needed cultural detail and special effect loads, a pattern density of from 350 to 750 polygons per 100 meter right triangle was allotted. The smallest shape, designed to resemble a patch of vegetation or a "sage brush" clump is one meter in diameter. The shapes range in size from there to approximately ten times that size, the largest shapes being part of the pattern that becomes the second level of detail as the smaller texture fades. Transition range for the fade of the small texture is approximately 400 feet from the eyepoint, for the larger pattern, 800 feet. Beyond 800 feet the 100 meter triangle ground polygon remains to the 6000 feet transition to null. Since the visibility constraint is 5280 feet, the 6000 foot "dissolving of the world" is lost in fog.

In the CT5A system, an "object" is a list of polygons and light strings stored in memory. This whole list, or subsets of the list, can be referenced in turn as if they were objects. In this way, for instance, all the levels of detail from most complicated to simplest could be addressed as subsets of the same list, saving memory space. This means that the data base designer can produce four very detailed, very dense textural patterns for 2D ground texture and extract twenty or thirty meaningful subsets to randomly place on the ground. The advantage to the user is a virtually random texture, visually, and a massively complex data base that uses less than half the available geometric storage capacity of the system. The memory thus saved is available for additional data base enhancement.

The 3D embellishment of the basis set pieces consists of ten 3D items: six or seven bushes, two or three medium trees, and one 30 foot tree. The bushes are visable within 300 feet of the eye, the medium trees within 600 feet, and the large trees fade generally at 1200 feet. The medium and large trees have several internal levels of detail. The most complex is 37 polygons, the simplest, five. At the maximum scene density, flying nap-of-earth just above or in between the trees (all of which will appear visually correct from any eyepoint) the apparent complexity of the world anywhere in the algorithmicly produced sections of the data base is about 300,000 polygons per square mile. As the pilot's elevation increases, the ground complexity decreases, but his radius of view increases, maintaing a quite uniform per-channel polygon load at any rotorcraft nap-of-earth or contour flight height.

Advantages of Generic Models

The NOE project called for a series of obstacle courses to test various common rotorcraft tasks in an environment that is highly structured but appears to be a section of the natural world. Several single task areas were designed, and a multiple task area was also assembled. These obstacle courses were originally conceived to be combined with the tactical data base described before but were finally implemented as separate entities that could be tesselated with the tactical area or used alone. These obstacle courses are:

1. Longitudinal task course: a series of parallel 300 ft. wide hills of 40, 50 and 60 foot height with trees and separated by valleys between 200 and 300 feet wide. Running roughly down the center of the valleys are a line of power

transmission towers with wires. To traverse the course, the pilot must go over a hill, under a wire, over a hill, under a wire, etc. The spacings are randomized so that a uniformly repetitive movement is not appropriate to getting through unscathed.

2. Lateral unrestricted course: a series of tree clumps in a line 400 feet apart. The object is to slalom in and out around the trees while proceeding down the line.

3. Lateral restricted course: tree clumps with 400 feet spacing but with a row of trees on either side parallel to the line of clumps. A group of trees closes off one end of the corridor. The pilot must enter the open end, negotiate the in and out while not hitting the restricted sides, and then up and over the trees at the closed end. Alternatively he can enter up and over the closed end, weave down the course between the tree clumps and exit straight out.

4. Pinnacle, lateral jinking, confined landing: one "course" with three small obstacle areas. The pinnacle is a small mesa structure 36 feet high and 75 feet in diameter surrounded with 25 to 30 foot trees at its base. The top is textured with 2D patches and 3D rocks and bushes, with an open area large enough to settle on. The pilot practices landings and takeoffs over the mesa where his craft is in and out of the ground effects of rotor wash and he must do precision touchdowns to avoid the 3D objects on the mesa top.

   The lateral jinking area is an enclosed box of trees with enough density for masking except through areas with "holes" through the trees. The pilot must move side to side, viewing through the area of no tree cover (and in the process exposing his craft to enemy observation) and then remasking himself. Target vehicles are provided outside the treeless "windows" at various distances.

   The confined landing area is a semicircle of trees of various heights with a lighted mobile landing site in the center. The open side of the semicircle is constrained by a line of telephone poles and wire. The pilot must enter over the wires or over the trees using certain minimum approach angles in order to avoid crashes.

5. The "S shaped" course: a multi-task area designed to test many of the basic skills of terrain avoidance/NOE flight. A corridor of dense NOE texture requiring the pilot to navigate over hills, under power lines, up and down grades with tree restricted sides, fly into a gulley, weave back and forth down a twisting corridor of trees, fly over telephone line and find and land in a very restricted clearing in a forest of tall trees.

These courses were designed to fit the same modularized structure as the tactical NOE data base so that they could be organized in whatever pattern was thought to be appropriate. The generic areas between the courses are decorated with similar 2D and 3D cue densities to the rest of the NOE tactical area so that nap-of-earth flight can be practiced while navigating from course to course. The overall area incorporates a variety of highly specialized and carefully designed operational features which do not exist anywhere in the real world. As such it will be able test skills that may not be adequately tested anywhere except in a simulation. Whereas the first non-real world simulator experiences were a poor substitute for "live" the newest ones may prove to be superior to "being there", allowing intensive drilling of specialized rotorcraft skills in a carefully designed environment.

HOW THE FINAL PRODUCT PERFORMS

Several different metrics must be applied to the final visual data base to fully understand how it performs. We wanted to have it load the IG to its nominal capacity, both for total polygons and per-channel polygons, without significantly exceeding either limit. We also wanted to minimize the difference between minimum, average and maximum ongoing loads, and also ensure that the scene details were distributed uniformly across the display plane, rather than being compressed at the horizon. We thus applied a number of different tests and analyses.

Overall Load

First, we picked at random a number of typical places in the data base, and measured load levels as a function of eye height. For each position, we measured maximum and minimum polygon counts as the eye ascended from five feet to 200 feet in 5 foot increments. We noted polygon counts as we slowly panned a full 360 degrees. The maximum values ranged from 990 to 1335, and the minimum values from 486 to 870. The spread was wider than we had initially expected, but a little intuition provided the answer. The terrain is very hilly. Since most of the polygon load is due to textural embellishment of nearby terrain surfaces, the load will be significantly affected by nearby hills whose far slopes are backfaced to the observer. Indeed, this reasoning is verified by the fact that the maximum and minimum values get closer together as the eye height rises and fewer of the nearby hills have backfacing far slopes.

Next we measured maximum and minimum polygon counts as we flew over the terrain in random directions at various terrain clearance altitudes of from 15 to 60 feet. We used the terrain plane feedback feature to maintain our height, and noted polygon loads of from 504 to 1534 polygons. These single channel measurements imply that we are right on our initial target of around 4000 total on-screen polys for the full three channel system. At all of these figures the system exhibits significant remaining capacity.

Continuous Image Density

The NOE mission objectives require that a fairly complex image be maintained for the pilot regardless of his orientation or altitude, and that the scene details be evenly distributed across the plane of the image. The normal compression of details into a small band at the horizon must be avoided, since IG capacity expended in that part of the image is largely wasted in terms of pilot visual cues. One method of quantifying the achivement of this continuous image density is as follows:

> For altitudes down to an equivalent slant range of 25 feet, the minimum number of edge

crossings per degree of field of view shall be 0.04, with a design goal of 0.1. (These values correspond to an angular spacing of scene edges of 25 degrees apart in both the horizontal and vertical directions, with a design goal of 10 degree spacing.) This requirement is to be measured with a regular grid of orthogonal scanning lines, at least six in each direction, which is overlayed on the image. Intersections of scene edges with these scanning lines are counted, and the achieved performance determined.

What this implies is that patches of terrain textural detail must be spaced approximately eleven feet apart throughout the data base, for an equivalent data base geographic density of 45,000 polygons per square kilometer. The design goal requires this to be boosted to 281,000 polys for the 10 degree spacing. These values are consistent with the cue needs analysis which derives from John Sinacori's criteria. We can examine our success analytically, and then submit actual imagery to the scanning analysis.

Model Content and Density

There are two flavors of terrain basis set triangles. One has 350 polygons of 2D textural decoration, the other 750. They are used in about equal amounts, so the average is 550 polygons. Since they are 100 meters on an edge, the overall textural density is 110,000 polygons per square kilometer. This is more than double the required density, but short of the design goal. Each basis set triangle has 10 3D features on it, giving a mean lateral spacing of about 73 feet between vertical height cues. Note that this implies a total data base content, throughout the 64 specific square kilometers, of 128,000 3D features and about seven million 2D textural patches.

Analysis of the Imagery

Figure 1 shows a typical nap-of-earth scene from an eye height of about 15 feet. The overall field of view is about the same as one display channel. The required grid of orthogonal scanning lines has been superimposed on the picture so edge crossings and feature distribution can be determined. We have analysed the scene in several ways to verify both our data base and the measuring methodology.

First, we counted edge crossings for each orthogonal scanning line. The counts for the vertical ones, left to right, are 39, 32, 26, 17, 30 and 27. The counts for the horizontal scans are, top to bottom, 1, 22, 24, 8, 7 and 6. The one count line lies almost entirely in the sky, and is not required or expected to reveal much complexity. This data indicates edge per degree ratios of 0.35 to 0.62 in the vertical direction, with an average of 0.53. In the horizontal direction the range is 0.14 to 0.54, disregarding the scan line which lies entirely in the sky, with an average of 0.29. These values are an order of magnitude above the required limits of 0.04, and all exceed the design goal of 0.1. Note that the distribution of scan line crossings is very uniform.

Next, we counted the number of polygons or polygon fragments in each of the 25 larger rectangular regions defined by the scanning lines. The counts for boxes along the top row were 23, 14, 14, 2 and 1. Most of these boxes were predominantly sky. The next two rows of boxes contain features of most importance to the pilot. The second row counts were 39, 35, 36, 49 and 29; the third, 15, 20, 15, 26 and 37. The bottom two rows of boxes contains very nearby details suitable for use in judging hover or landing. These counts were 7, 6, 6, 5, and 6 (far row), 5, 4, 2, 2 and 5 (nearer row). Even for the bottom row the average angular spacing of scene edges is way above the design goal. There are no areas of the image devoid of numerous visual cues, no "bare" spots. The intent of providing "continuous image density" has certainly been met.

ADDITIONAL PERFORMANCE FEATURES

As the data base design took shape, we realized that several CT5A features which were not originally planned in the data base could have significant operational value. These included mechanisms for detecting own ship collision or crash, height of own ship above the immediately surrounding terrain, and a data base area where hide and seek tactics could be staged with several threat vehicles. These capabilities required significant additional work in the data base, but were implemented because of the increased value of the system to the end users. These features allow more precise measurement of pilot and rotorcraft performance in the NOE and Obstacle Course areas, as well as the ability to evaluate overall performance under the stress of an active tactical engagement.

Height Above Terrain

This function returns to the Host computer the plane equation describing the terrain immediately below a test point. This test point is typically associated with own ship, but may be generally used to interrogate the terrain at other places some distance from own ship. The plane equation may be used to compute the height of the terrain, and hence the clearance of own ship to the terrain. It can also be used to position a moving vehicle on the terrain at the proper height and orientation. This allows one to land on the ground and have the simulation reflect the local orientation.

The Height Above Terrain function uses hardware and microcode within the CT5A to operate on a terrain-plane data base which is a subportion of the visual data base. The algorithm processes the data base tree to determine which portion of the terrain the test point is immediately over, and uses the same data provided by the modeler to control the visual priority ordering. The addition of the terrain-plane constructs to the visual data base was a particularly straightforward process, since it primarily involved adding ground plane equations to the basis set elements. The planes are defined procedurally from the same data which describes the polygonal terrain skin, and follow the basis elements through the instancing process to wherever each is finally placed.

In the tactical data base, ground planes are provided for all the terrain skin, and are modeled to match the terrain at its most complex level of detail. The treatment is exhaustive, and the height functions can be used anywhere within the visual data base. A test point is submitted every computational field, and the results are used to

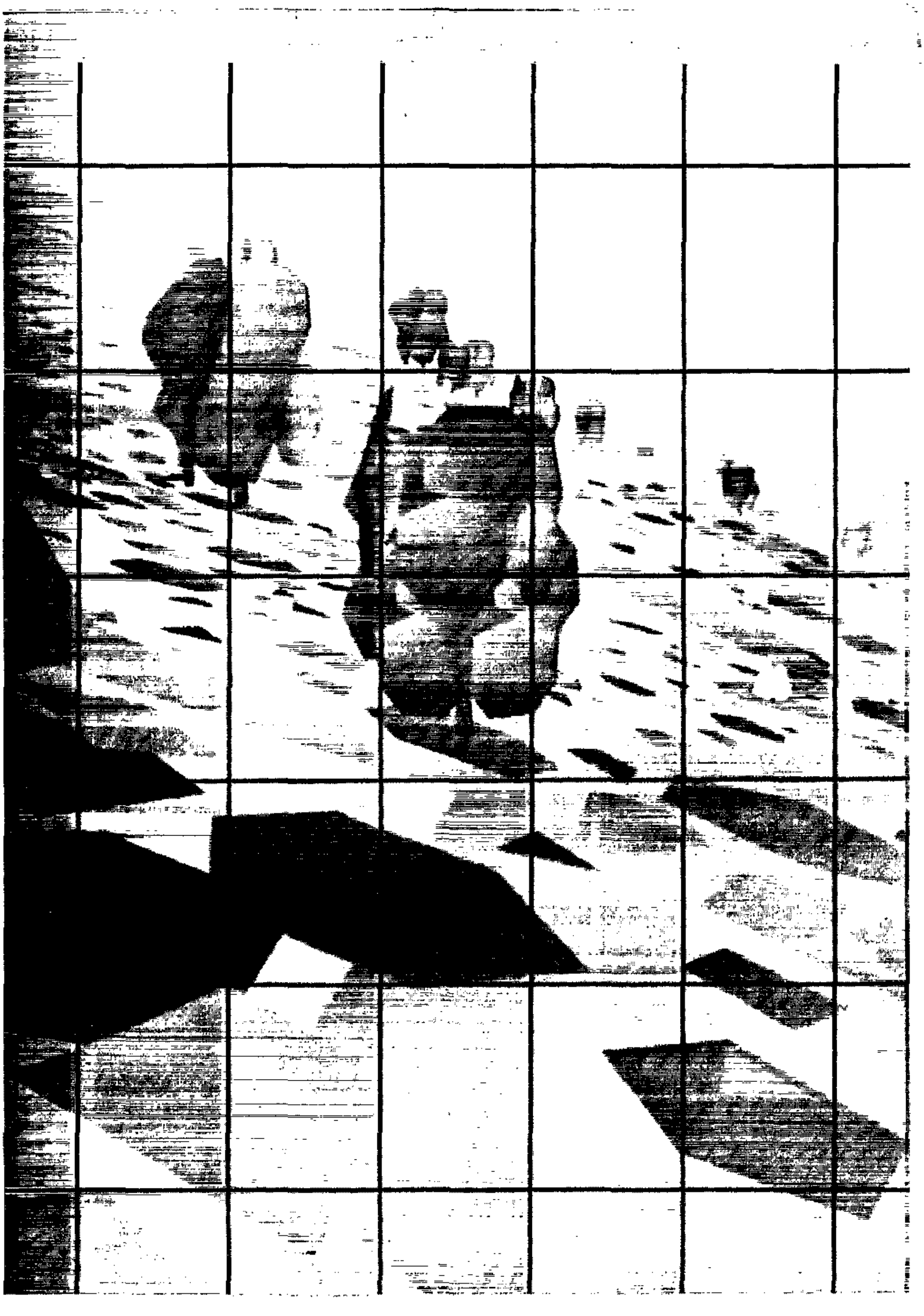

Figure 1

provide the radar altimeter readings, measure height holding performance, and effect proper responses upon landing. The data is also used to provide inputs to rotor wash and ground effects, and detection of own ship below the terrain skin constitutes a "crash".

Collision Detection

A primary mission of the system is to investigate the rotorcraft man/machine interface in the nap-of-earth regime. This implies precision flight in close *proximity to three dimensional objects*, and ingress/egress from very confined landing areas. Collision detection has typically been used to confirm strikes in small selected areas of the data base with a small set of selected objects. As these requirements were discussed during the data base working conferences, we realized that addition of a much more general collision detection capability throughout the data base would substantially improve the ability to measure the conduct of flight operations.

The collision detection function is implemented in the CT5A hardware, and is a generalized extension of the notions of Cellular Priority. This method of sorting the visual environment differs significantly from the more common "binary tree" approaches, *and offers a much more efficient and general* mechanism for addressing complex three dimensional problems. Collision detection is accomplished by determining if a test point lies inside a volume of space identified as a collidable solid. If so, the collision is reported, along with some information on the item which was hit. Collision volumes are typically provided for things like trees and buildings, but the methodology is very general and can be used to provide complex, high fidelity volumes which include concavities.

The collision data base is a functional subportion of the visual data base, and collision volume constructs may be instanced just as visual elements are. The addition of collision volumes to the NOE data base was expedited by the extensive use of basis sets. A collision volume counterpart was modeled for every basis set element and library component. These were organized and positioned using the same structural information provided for the visual data base. An exhaustive treatment was provided, in that collisions are reported for every tree and large shrub in the entire data base, as well as all structures.

A collision test point is processed by the system every computational field, providing fifty tests each second. The test points are chosen in rotation from a collection which describes the shape and location of the extremities of own ship, including tail boom, rotor and skids. The results are returned to the Host computer, which monitors and logs the performance of the pilot, and can also invoke a collision response in the visual scene. This capability has significantly improved the usefulness of the data base.

Tactical Interaction Area

The data structures which allow the IG to sort the fixed environment in real time are highly optimized, and encode a great deal of information in a very compact form. These data structures rely on the fixed relationship of all the scene elements to achieve the high efficiency required. When a moving model is introduced into the scene, it must be sorted into the proper visual order with the fixed objects, but of necessity cannot use these same mechanisms. The CT5A provides this capability by allowing collections of objects to be sorted by the real time system based on their respective ranges from the viewer. Such a collection is called a Range Sorted Mesh, and it may include both fixed and moving scene elements.

A *range sorted mesh can include up to 16 items*. When such a mesh is processed its items are output in the order specified in an associated ordering list. This list is managed in real time by the software, and 64 such lists are available in the system. The real time system has data on the location of each item of each range sorted mesh. It performs computations to generate an ordering for each mesh, and updates these lists in the Object Manager. As long as items remain separated by their combined "size", the ordering will work visually. The number of lists and objects which can be managed is a function of work load in the general purpose computer: Note, though, that a generally correct solution can be maintained for such a list without updating it every computational field, thus a larger number of items can be managed if the sorting is spread out over multiple field times.

For the NOE tactical data base, a two kilometer by two kilometer staging area was provided. Within this area are four 16-item range sorted meshes, one to a quadrant. Each mesh consists of four moving objects and 12 stationary ones, with occulting properly computed for any orientation of these items. Each computational field the real time system computes a new range ordering for one of the quadrants. The management of all this is transparent to the pilot, who perceives instead an area of four square kilometers within which 48 stationary and four moving models interact properly.

The user can select between a library of trucks, jeeps, Russian and American armor, rotorcraft, and fixed wing aircraft for any of the four moving models. The "stationary objects" include houses, small hills (complete with nap-of-earth texture and trees), small patches of forest and orchards, power towers, etc. Cat and mouse attack scenarios and masking/unmasking exercises are possible with as many as four threats against ownship in a very realistic environment.

CONCLUSIONS

The NOE design effort has produced a functioning data base that meets virtually every design criteria of the Sinacori report. It fulfills the original design intent while providing a much more usable product in the end. Because of the time saved by utilizing newly developed data base design techniques, it's scope has been enlarged from what was originally envisioned and its functionality increased with an acceleration of the original completion schedule. The successful implementation of these NOE design concepts in a practical system holds significant promise that other related visual simulation needs which before have not adequately been addressed by the technology can now be met.

ACKNOWLEDGEMENTS

The development of these concepts has been an exciting and rewarding experience. Much appreciation is extended to Steve Lund and Marlin McDaniel of Evans and Sutherland, and Cary Wales for his keen technical insights, and his steady reasurring hand at the helm.

REFERENCES

[1] Sinacori, John B., Draft Final Report, "Research and Analysis of Head Directed Area of Interest Visual System Concepts." NASA Ames, November 1982.

[2] Wales, Cary E. and M. Cosman, "DMA and CIG: A Shotgun Wedding." Proceedings of the 5th IITEC Conference, November 1983.

[3] Costenbader, Jan L., "CIG Data Bases in an Instance --Bits and Pieces." Proceedings of the IMAGE III Conference, May 1984.

[4] McCormick, Dennis and Tamara Smith, Frank Lewandowski, William Prescar and Elizabeth Martin, "Low Altitude Database Development Evaluation and Research (LADDER)." Proceedings of the 5th IITEC Conference, November 1983.

ABOUT THE AUTHORS

Mr. B. James Voorhies is a senior data base designer with Evans and Sutherland Computer Corporation: He has been the principal developer of the tactical NOE data base. Mr. Voorhies holds Bachelors degrees from Brigham Young University and Southern Utah State College.

Mr. Michael A. Cosman has provided technical support in the development of CIG visual systems at Evans & Sutherland for over 10 years, working primarily on image quality and system architecture issues. Mr. Cosman holds a Bachelors degree from Brigman Young University.