

TERRAIN INDEPENDENT FEATURE MODELING

L. Charles Clark and Michael A. Cosman
Evans & Sutherland Computer Corporation
Salt Lake City, Utah

ABSTRACT

Historically, much of the time and expense of developing visual environment databases has occurred in the process of customizing three-dimensional features to fit properly on the terrain skin, and this interdependence has often imposed limitations in terrain fidelity and feature placement and density. A new Evans & Sutherland system performs this terrain/feature marriage in real-time with special feature-conforming processes which are implemented in the CIG hardware and which rely on a depth-buffer visual priority solution. This allows modelers to optimize the terrain model for maximum fidelity, and create and organize the feature overlay without regard for the topography of the underlying terrain, greatly simplifying feature design and placement. Modelers can work at much higher levels of abstraction, while generating visual environments which are more accurate and realistic.

This paper describes a new set of modeling strategies which convert high-level feature representations into displayable databases. Broad-brush feature descriptions such as DMA DFAD or 2851 can be rapidly transformed into compact data structures which create dense high-fidelity visual environments. Geo-Typical and Geo-Specific features can both be readily accommodated where mandated by mission requirements, and advanced hardware instancing modes allow features to be highly customized with each placement, achieving high compression of the feature database. The development process may be largely automated, and feature and terrain production can be performed in parallel, greatly reducing database development time and cost.

INTRODUCTION

Current image generation systems employ visual models in which the marriage of features to terrain is performed during the model-building step. As the database is constructed, features coerce the terrain and the terrain coerces the features, with resulting compromises in database fidelity and resolution as features are interleaved into the terrain model. Areal and lineal features are broken up to accommodate the terrain subdivision and level-of-detail strategy, point features are moved to avoid falling on terrain boundaries, and each occurrence of every scene detail is customized to reflect the orientation of the underlying terrain facet. These processes are complex, time consuming, and expand the database storage and paging requirements significantly. These in turn impose crippling limitations on database generation speed and efficiency, total database size, content and density, and set limits on the maximum speed the viewer can travel through the database (Cosman, 1989).

All of these problems, which are inherent in the process of merging terrain and features during database generation, must be solved by modelers and modeling tools. If terrain and feature databases could be modeled separately and independently, and then processed with a CIG system designed to merge terrain and features in real time, much of the inefficiency and limitations of integrating features and terrain during the modeling process could be eliminated. We will review the limitations imposed on visual environment database generation by current CIG architectures, and we will discuss how these limitations are overcome through a new hardware system architecture which supports complete terrain/feature independence. We will then describe a new philosophy for feature database generation, and illustrate powerful new strategies for feature modeling and placement which provide large increases in database development efficiency and visual effectiveness.

CIG-IMPOSED MODELING LIMITATIONS

In the 1970s, the first real-time CIG systems with a scan-line range solution for visual priority to handle interpenetrating objects were built. However, the market demands for higher scene capacities sparked a CIG technology revolution, yielding a new architecture for the 1980s. The core of this new architecture was an area-based approach to picture computation. Images were computed in feature-sequential order instead of scan-line order. Area processors replaced scan-line processors and area representations (spans) replaced scan-line samples (segments). This area-based approach to image generation, combined with a faster and more efficient front-end viewpoint processor, resulted in a significant increase in polygon processing capacity (Schumacker, 1980). Among other things, this increased scene capacity made possible the real-time rendering of complex topographic models for the first time. And the introduction of 3 degree-of-freedom (3-DOF) hardware instancing (i.e., the ability store an object once in the image generator's memory and then position it in many places on the 3-D terrain model) also added to increased scene complexity (Mayer and Cosman, 1981). These innovations in CIG architecture, along with the broader application of texture, made possible significant advances in scene complexity and scene realism during the last decade.

An integral part of the area-based architecture was the use of list or cellular priority as the computational mechanism for sorting the visual environment into a front-to-back order prior to pixel generation. Although area-based architectures provided the most efficient and cost-effective approach to achieving increased scene complexity, they imposed certain limitations on the nature and construction of the visual model. These limitations, while inconvenient, did not significantly mitigate the usefulness of CIG systems in any simulation

application until recently. However, with the introduction of mission rehearsal requirements in the late 1980s, these limitations became the critical factor: extremely rapid generation of large area, highly specific databases was simply not possible. These limitations are discussed in the following sections.

Cellular Priority Constrains Feature Placement

Cellular priority requires that the visual occultation relationships for features versus features, features versus terrain, and terrain versus terrain be pre-defined by the modeler and modeling tools and imbedded in the database. These relationships are constructed with separating planes (i.e. any two objects must be separable by a plane in space) and all separating planes in the database must be organized in an hierarchy of meshes with a variety of topological constraints. While this mechanism can provide a very efficient object-level solution to visual priority for many objects in a database, it does not handle the separation of objects which are not convex, such as a small square building nestled into the corner of an L-shaped building. List priority also disallows the inter-penetration of polygons, requiring that features positioned on the ground be modified to custom-fit the slope and orientation of the local terrain polygon. In addition, the motion of dynamic models among terrain and feature objects in the database is often constrained. Finally, the rules and topological requirements of the priority structures are hard to implement in any sort of automated development process.

Features Must Avoid Terrain Boundaries.

One of the most troublesome limitations of list priority is that features are not allowed to overlap terrain boundaries in a database. Two factors may cause features to fall on terrain boundaries. First, real-world feature positions are largely unconstrained, so frequently a feature footprint overlays a terrain polygon boundary. In this case, the feature must be moved far enough from its actual position to be clear of the terrain boundary, or must be clipped into terrain-specific parts straddling the boundary, at considerable expense. Second, the footprint of a feature may exceed the size of the terrain cell where it is being placed. In this case, moving the feature will not solve the problem. Instead, the feature must be subdivided or clipped to eliminate the overlap conditions, or the terrain cell must be enlarged to accommodate the feature. The point is that conflicts between features and terrain boundaries must be resolved during the database generation process, either by the modeler or the modeling tools. Since the probability of occurrence for either of these problems is inversely proportional to terrain facet size, database design tends to be driven toward larger, coarser terrain polygons, with correspondingly degraded terrain fidelity.

Feature LOD Dependent on Terrain LOD.

Another limitation of list priority is that feature LOD is inexorably entangled with terrain LOD. For optimum processing efficiency, terrain LOD structures typically group a number of terrain polygons into fixed listed objects at all but the highest terrain LOD. This strategy makes it almost impossible to integrate features into the terrain at any but the highest LOD. In addition, it is very difficult to plant features on lower terrain LODs and then ensure that they remain properly attached as the terrain transitions to higher LODs. As a result, the threshold of emergence (transition range from null) for most features in a database can be no greater than the transition range into the HLOD terrain. This results in feature emergence at ranges shorter than required, and terrain development at ranges (and to polygon sizes) larger than desired. The visual effectiveness of both domains is compromised by the necessity of delaying feature emergence until the highest terrain LOD has fully evolved.

Feature/Terrain Database Dependence

In DFAD or 2851, feature positions are defined in terms of lat-long coordinates. The elevations of feature positions are not provided since the digital topographic data contains all terrain elevation information and features are presumed to sit on the terrain surface. However, traditional CIG architectures require that features be explicitly positioned in the database in X, Y, and Z. So for each feature placement, modelers or modeling tools must compute a specific Z component. In addition, instancing generally only occurs at the object level rather than at higher, broad-brush levels of the database hierarchy, since collections of co-instanced features would have to lie on flat (if not level) sections of terrain. The operations of elevation computation and instancing on a per-object basis severely limit feature database generation speed and compression, especially as feature density and terrain model complexity requirements increase.

A related but separate problem is that of feature/terrain fit. Because many CIG systems rely on list-based, object-level priority schemes, the interpenetration of feature objects with terrain is not allowed. To achieve a perfect fit with the local terrain, a unique, custom-clipped version of each feature must be created by modelers or modeling tools for each slope and orientation of the underlying terrain where it is eventually positioned. This process is time consuming and significantly expands the model library. The alternative is serious visual compromises in feature/terrain fit as unanchored features "float" when placed on terrain slopes and orientations they were not designed for, or if terrain must be "bulldozed" at loss of fidelity to accommodate a limited feature model library.

Limitations of Standard 3-DOF Instancing

The primary restriction of the instancing capabilities of most CIG systems is that they are limited to 3-DOF (X, Y, and Z) feature placement.

Given this constraint, each instance of a feature model appears with the same size and orientation. As a result, model libraries are forced to include various rotations and sizes of the same object in order to enhance variety and improve fidelity in the visual scene. Another limitation of 3-DOF instancing is apparent in the use of basis sets (Costerbader, 1984). In the case of linear feature basis sets for modeling roads or powerlines, many lengths and orientations of the same feature segment are included simply to approximate the path of the feature in the real world. Such basis sets typically grow to ungainly sizes as they are generalized to handle the almost infinite variety of feature topologies which can occur.

SOLUTION: A NEW CIG ARCHITECTURE

The solutions to feature placement constraints, feature/terrain dependence, and feature instancing limitations are best implemented in CIG hardware, not modeling tools. A new Evans & Sutherland CIG system, the ESIG-4000, possesses the architectural characteristics and the computational power that are key to solving these problems.

R-Buffered Visual Priority

Recent advances in computer memory and processors have made sub-pixel, range-based visual priority solutions tractable. Range solutions to visual priority can eliminate unnecessary obstacles for modelers and modeling tools. If carefully implemented, a range buffer (R-buffer) yields all the benefits of a Z-buffer without the side effects of image quality degradation and excessive image depth-complexity. The R-buffer also avoids the mathematical difficulty inherent in Z-buffers when dealing with FOVs greater than 180 degrees.

A CIG architecture which determines visual priority at the sub-pixel level based on range from the eyepoint facilitates the integration of features with terrain and eliminates the requirement to embed priority information in the database. This in turn simplifies the modeling process and minimizes the database storage size, both of which contribute to the rapid database turn-around necessary for supporting mission rehearsal. Another advantage is that the real-time integration of dynamic models into the database is unconstrained, allowing greater freedom of object motion with correct visual priority all of the time. Finally, the inter-penetration of objects can be easily and realistically portrayed. For example, the foundations of buildings on slopes need not be clipped at the ground plane in the database since this is done automatically in hardware. This results in quicker database development, and a more realistic image.

Real-Time Feature Conforming

A design objective of the system was to separate the terrain and feature models, not just at the modeling stage, but completely enough to support an entirely different model-development paradigm. Mission rehearsal systems must achieve many-fold increases in database generation

efficiency, while at the same time reducing the stored model data by large factors. These conflicting objectives rule out compile-time solutions to the terrain-feature marriage, and require the inclusion of powerful real-time processes internal to the CIG. The resultant system not only fully supports this new paradigm, but allows the creation of visual environments, imagery and effects which were hitherto impossible, no matter how much time was allowed for database generation. For example, continuous terrain level-of-detail evolution cannot be achieved with a compile-time solution, since the instantaneous height of the terrain is not known until a relative eye position is specified at run-time.

In considering the conforming processes necessary to implement terrain-independent feature modeling, three modes were identified. The first, determining a height Z at which to display an object so it "sits" on the terrain, has been approximated for a decade by systems which compute and store appropriate Z values for each instance of an object. This "origin-conformal" mode is conceptually the easiest to provide in real time. The new approach requires the CIG to determine the instantaneous height of the terrain below the origin of the object, and translate the object to this height. The CIG must take into account the state of processes controlling the continuous terrain LOD to do the conforming properly. This mode may be thought of as "conforming relative to a point."

The origin-conformal mode is either inadequate or inappropriate for a broad class of features which are suspended or supported above the terrain. The second mode, "conforming relative to a line," is called line-conformal. The new approach is implemented by computing the terrain heights under the endpoints of a modeled "line datum," and displacing modeled vertices relative to this sloping datum. Note that the ends of the datum need not lie on the same terrain polygon, and that the sloping datum (and hence the resultant transformed feature geometry) will not generally respond to terrain details between those endpoints.

In the third mode, "conforming relative to a surface," each vertex of the polygon to be conformed is displaced by the Z of the terrain immediately below it. If the feature polygon straddles a terrain polygon boundary, it is cut or clipped by the hardware at that boundary, and the ends of the clipped edge are also displaced by their underlying Z values. This accommodates the "surface-coplanar" case, and provides proper "settling" of above-ground geometry into position. The clipping function is essential to assure visually reasonable results, and ensures that features will behave properly even when they straddle terrain boundaries. This aspect of terrain-feature independence is a subtle one, but is the key to employing high-fidelity terrain with small terrain polygons, while allowing the exact placement of scene features where the source data demands.

These three modes can generally be designated a polygon at a time, and mixed together in the construction of features without requiring artificial modeling sequences or structures. Datum

information is transformed by the instancing process, along with the feature geometry, so it is always correct for each conforming situation.

Enhanced Instancing

Over the last decade the large increases in continuous image density have been more the result of advances in database development and management techniques than increases in CIG polygon capacities. 3-DOF instancing, which first appeared in the CT5, provided 1000-fold increases in the geographic density of scene features, while containing the overall growth of database storage. To meet the model-compression goal of the new system, instancing has been enhanced to include specification of a heading angle, independent scale factors in X, Y and Z, and independent mirroring through X, Y and Z. As in previous CT systems, instancing is hierarchical, with the effects of one instance-reference concatenated to prior references in a coordinate chaining that implements the functional intent of the modeler.

Instancing now parallels the DFAD/2851 paradigm much more closely, and allows greatly increased utilization of the stored feature models which correspond to the digital feature catalog. Later discussions will illustrate this power more intuitively.

Betweening Level-Of-Detail

Continuous image density is maximized when the visual environment implements a thorough multiple-LOD modeling strategy. Simple versions of distant scene details conserve system capacity for a larger number of more detailed close-up features. The modeling problem, part technical and part artistic, is to develop case-by-case strategies for frequently used models that provide a visually effective, non-distracting LOD progression from the complex models needed up close to the simple ones which suffice in the distance. The effectiveness of any approach at maximizing visual results depends on getting away with the shortest possible LOD transition ranges, and this in turn depends on finding transition mechanisms which can be used at short ranges without distracting the viewer.

Discrete model changes were the norm until CT5A introduced transparency-fading as an additional mechanism to mitigate the distraction of changes. A third mechanism called "betweening" has been added to the new system. In effect, each vertex can be defined to have both an initial and a final position. A betweening parameter is used to interpolate a display-vertex position somewhere in between these two (hence the name...), and is generally controlled by range from the eyepoint to the feature. The process is general, powerful, and results in greatly reduced noticeability for LOD changes. The point-polygon topologies of adjacent LODs are designed consistently so that structural complexity can emerge gracefully along with geometric detail, or the two may simplify naturally, as betweening interpolates LODs over range.

A NEW MODELING PHILOSOPHY

In the absence of constraints imposed by previous CIG architectures, modelers and tool builders are now approaching the problem of database design from some entirely new perspectives. Consider, for example, feature/terrain independence. The feature database and the terrain database are created and maintained separately, and then merged in real time in the image generator to support the instantaneous scene. The features are modeled as if they were all on a flat, infinite plane, and the terrain is modeled without regard to the features which will eventually be placed on it. This strategy allows each database to be modeled in its most efficient form, allowing high data compression, fast paging, and the freedom to explore new level-of-detail strategies. The creation of these databases from DMA or 2851 source material is very fast, and final database storage requirements can be significantly reduced. This approach results in a much higher total database content, feature density, and allowable flight speeds (Cosman, 1989).

With the feature placement task now in the 2-D domain, several advantages accrue. The 2-D, planimetric view of feature database construction is the most intuitive human interface, being similar to the maps, aerial photos, and engineering drawings that people are accustomed to working with. Graphics workstations suited to the 2-D task are less expensive than 3-D workstations, and 2-D operations are less compute intensive, and hence more interactive. Manual feature positioning is simplified to a "point and click" process, where the modeler interactively places, rotates, and scales features to "fit" their footprints in the photographic terrain texture. Modeling tools employing DFAD or 2851 data can be much more straightforward and efficient.

Feature modeling is now unconstrained by the need to understand and provide list-priority data constructs. Objects are no longer required to be convex, but may now be nested. They may interpenetrate freely, and complex models such as aircraft may now be constructed as several separate parts such as wings, fuselage, and stores which are then concatenated together, relying on the R-buffer to resolve the connections of one part with another.

FEATURE MODELING STRATEGIES

A few specific examples help illustrate the combined power of the R-buffer, advanced instancing, and feature/terrain independence. In each case the overall modeling strategy benefits from a powerful synergy of these capabilities.

Origin Conformal Examples

The most obvious conformal mode is origin conformal, used ubiquitously for placing an infinite variety of point features, such as buildings and trees. However, when structured hierarchically the origin conformal approach yields enormous modeling leverage. The following example explores both the obvious and subtle capabilities of origin conformality.

The Infinite, Dense Forest

In many cases large areas of the visual environment are covered by trees, and the occlusion of surface features such as roads by tall surrounding trees is a key visual issue. Proper visual behavior can only be simulated if a dense decoration of 3-D trees can be provided out to significant visual ranges. When the required density of individual trees is multiplied by the very large geographic areas usually required, staggering quantities of scene data are suggested. The time required to instantiate these trees individually, and the resultant quantity of model data, make a model-time marriage of trees to terrain intractable.

In a CIG system that keeps terrain and features separate, the "dense forest" problem is approached by first creating a few terrain-independent tree models. The tree itself is made "origin conformal" so it appears properly "planted" on whatever local terrain slopes it is eventually applied to. A "surface conformal" cast shadow is modeled, and it will always appear coplanar with the underlying terrain. The tree is given several lower levels of detail which will be invoked at successively longer viewing ranges to minimize the number of displayed polygons expended on distant trees. The final instancable tree model can be treated as a single "object" which will interface properly with any terrain slope it is eventually placed on.

The next step is to create an "infinite", dense, visually varied forest overlay using advanced instancing and hierarchical database structures. Beginning perhaps with just two compiled and stored models--a coniferous and a deciduous tree, a grove is created by referencing each model numerous times with varying position, scale and orientation data. In the displayed grove, each tree appears unique because it has a unique size, proportion and orientation. Because the instantiation pointers are very compact data structures, a complex grove can be created using only a few hundred additional bytes of data. This grove can in turn be instanced numerous times with variations in position, orientation and scale to provide yet a larger piece of forest that exhibits no obvious regularity or repetition. This next level of forest also requires only a few hundred additional bytes of data.

The instancing process can be continued hierarchically until the final patch of forest covers many hundreds of square miles, yet visually seems never to repeat or exhibit noticeable regularity or structure. This large patch of forest can then be placed as required, and as the CIG unravels and instantiates the hierarchy, each individual tree and shadow is properly conformed to its local underlying terrain, while geographically coherent LOD management occurs properly.

Use of Mesh Signatures

The "signature" function can be used to instantiate customized portions of the forest whose boundaries are contoured to match the forest outlines depicted in the source material. It can also be used

to "thin" the highest-level grove areas to provide different forest coverage levels. Signatures provide additional database compression and allow accurate boundary representations to be compiled efficiently. They are particularly effective when applied to hierarchical database structures.

Within the database hierarchy, references to scene details are collected into "meshes," typically organized so a mesh contains all scene details which will eventually populate a particular geographic region of the visual environment. At the detailed levels of the hierarchy, these references are to directly displayable scene geometry. At broader levels, meshes contain references to subordinate meshes which further subdivide the geographic region. The database hierarchy is organized geographically to expedite database paging, increase the effectiveness of field-of-view and level-of-detail tests, and facilitate local-area updates (Mayer and Cosman, 1981). Signatures may be used to suppress further processing or display of some references contained within a mesh. Since the signature bits are stored in a higher reference which invokes the mesh, different calls can customize the same mesh in different ways.

Suppose the forest hierarchy consists of a succession of meshes, each of which subdivides its assigned area with a 4*4 matrix of meshes, on down to individual trees as shown in figure 1a. A 3.2 Km square of forest can be placed with a single instance reference. Suppose that in one particular instance of this square a few clearings and a boundary for a river need to be created. Perhaps six of the sixteen 800-meter regions making up the square are full of trees, three have no trees, and seven are partially populated with trees. The signature associated with this instanced square would delete the ten regions that are not full of trees. The two partially populated regions would each be created by instancing the 800-meter regions which are part of the 3.2 Km hierarchy, but with custom signature bits which further refine the outline of the clearings and river contained inside them. This process continues until individual trees are either planted or deleted at the boundaries of the clearings and the bank of the river, resulting in a very accurate portrayal of the source material with only a modest amount of data as shown in figure 1b. The example shown requires about 2300 bytes of data to encode the region to 25 meter boundary accuracy.

Conversion of DFAD Into Signatures

Areal feature data is encoded into mesh signature structures by first creating a geographic bit map from the DMA or 2851 boundary description of the feature, or from an aerial photograph. The size, orientation and resolution of this map relate to the particular hierarchical instancing approach used in building the forest square. The bit map is reduced to a succession of lower levels of detail which follow the mesh LOD structure. At each level, the new matrix contains a bit wherever the underlying 16 bits are all set, and a zero otherwise. Each group of 4*4 bits at each level of detail comprises the mesh signature for the subordinate structure. The conversion is fast,

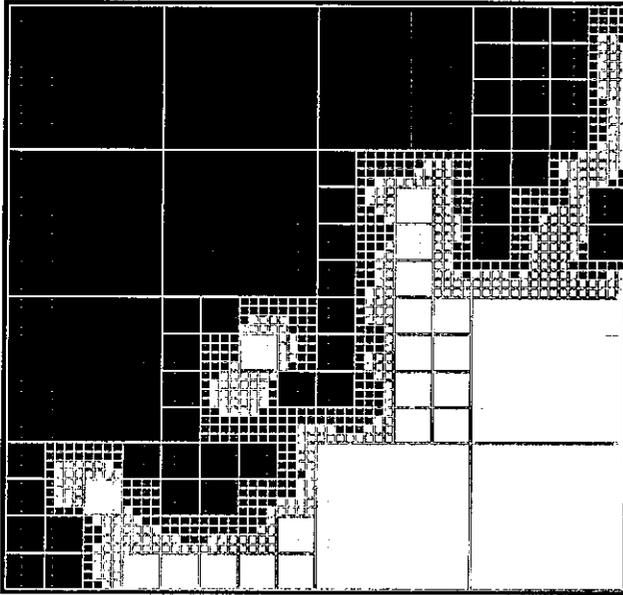


Figure 1a

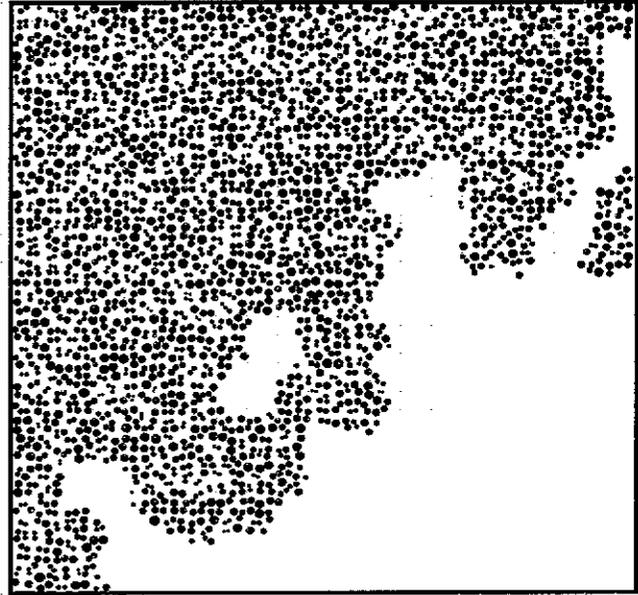


Figure 1b

Figure 1a shows the structural hierarchy of a forest model consisting of a succession of square meshes, each of which subdivides its assigned area into a 4*4 matrix of meshes. The custom signature bits at each level of the hierarchy enable or disable tree regions, successively refining the outlines of the clearings and a river. This process continues until individual trees are either planted or deleted at the boundaries of the clearings and the bank of the river as shown in Figure 1b, resulting in a very accurate portrayal of the source material with only a modest amount of data.

automatic and accurate. Note that other types of features which require extensive, dense representations can be similarly handled. For example, large areas of residential housing can be constructed from a small collection of models which are instanced to align with and follow the density and spacing of the desired area. Mesh signatures can be used to contour the resulting regions to accommodate embedded business districts, parks, etc. and to follow the outline of the city being simulated. Of course, unique features of importance would be modeled specifically and photo-textured for proper real-world correlation.

Line Conformal Examples

Line conformal features are modeled relative to an imaginary line segment which establishes a parametric datum. Typically this datum runs the length of the lineal feature being modeled, and each vertex of the feature includes a value which defines the position of the vertex in parameter space. This value is typically a fraction between 0 at the initial end and 1 at the final end of the line datum.

When the object is rendered the endpoints of the line datum are first planted on the underlying terrain. As object vertices are transformed, the modeled Z values are added to the interpolated Z

values for the particular position of each vertex along the line datum. The object is thus mathematically sheared or skewed relative to the datum. Vertical edges remain vertical, and planarity is preserved. Because texture is tacked onto the polygons with texture coordinates at each vertex, the texture is also sheared, and remains properly affixed to the surfaces. A variety of important visual features require this mode for accurate, efficient representation. In general, anything that attaches to the terrain at occasional support points should be modeled this way to ensure proper visual behavior regardless of the instantaneous terrain level of detail.

Powerlines

For example, powerlines are the nemesis of rotary-wing pilots, and their proper inclusion into the database is critical for low-level mission training. They represent a major flight hazard, doubly dangerous because they can be either obvious or invisible depending on their orientation and the local illumination conditions. Powerlines represent an interesting example of several conforming modes working together.

The pole and cross-arms are modeled as an origin-conformal object, and a surface-conformal cast shadow may be applied. A single strand of wire is



Figure 2

The powerline model in Figure 2 is constructed automatically with numerous instances of a single pole with crossarms modeled as an origin-conformal object with a surface-conformal cast shadow, and a single span of wire modeled as a line-conformal object. Poles are instanced appropriately and wires are positioned and scaled to stretch properly from insulator to insulator. The feature conforming modes ensure that the wires hang in realistic arcs and connect continuously and properly to the crossarms as the powerline snakes up and down over hilly terrain. The compact data structures used to instance objects provide high compression of the resulting database.

modeled with its catenary arc, as a line-conformal object. Again, a comprehensive multiple-LOD strategy is used to ensure that distant poles and wires are represented with the fewest possible polygons, and also that the proper degree of noticeability is preserved for the given display geometry.

The path of the line of poles comes from DFAD or 2851 or is digitized as a sequence of individual pole positions by a photo interpreter. The pole positions are used to compute heading angles for the poles and cross-arms as the powerline wends its way across the terrain. The cross-arm lengths, wire attach positions and cross-arm heading angles are used to establish the endpoints for the line datum for each strand of wire. Wires are positioned and scaled to stretch properly from insulator to insulator, and the associated line datums are also scaled. All of this, of course, proceeds automatically in software once the path of the line is specified in 2-D. Figure 2 shows precise positional correlation of the poles and proper behavior of the suspended wires, which hang in realistic arcs from the cross-arms and connect continuously and properly as the line snakes up and down over hilly terrain. Again, the compact data structures used to instance objects provide high compression of the resulting database. A three-wire line with 300-foot pole spacing requires about 4000 bytes per running mile to accurately depict. Compilation rates for these data structures are in the tens-of-miles-per-second range.

Bridges

Bridges are tactically significant as navigational waypoints, targets, and critical avenues for ground-based vehicles. In the mission rehearsal arena, accurate depiction of the bridge geometry and proper interfacing of the bridge to the local terrain is essential. For more general usage, the several types of bridges named in the DFAD catalog can be modeled and stored, and called up as needed with proper position, scale and orientation to span the required spaces. The line conformal mode allows each instance of the displayed bridge to connect properly to the terrain at its endpoints.

In figure 3a, a bridge is modeled so that the roadbed is nominally at $Z=0$, with additional structure which extends significantly above and below the roadbed. An origin-conformal base for the support columns is modeled, and will be instanced to properly spaced positions along the bridge. When conformed, the bases will sit on the underlying terrain at the proper elevations below each column. Intermediate support columns are modeled to extend far enough down to ensure contact with the conformed bases. The columns will be clipped off properly by the R-buffer where they penetrate the bases. A surface-conformal cast shadow is modeled which will lie properly on the terrain below the bridge. Because the embedded instancing and the line datum are transformed by the instantiation employed to place the bridge, the relationships of the roadbed, girders, columns and bases are maintained as shown in figure 3b. The resulting bridge model is

terrain-independent, and will work properly wherever it is used.

Surface Conformal Examples

Vertices of a surface-conformal object are computed and displayed relative to the underlying terrain surface. The Z value of every vertex has added to it the Z of the terrain directly below it. Since each vertex of a feature polygon may lie above a different terrain polygon, surface-conformal feature polygons must be clipped, or divided, by vertical clipping planes at the terrain polygon edges. Note that the conforming process is not simply a projection of vertices onto the terrain, but a displacement of vertices relative to the terrain, with clipping and piecewise linear shearing of the attendant polygonal geometry as required. After conforming, the object still "hangs together," even though it may straddle terrain polygon boundaries.

The simplest and most intuitive use for surface-conformality is to render feature polygons which are co-planar with the terrain surface, such as roads, runways, parking lots, shadows, etc. Since conforming modes can also be used with dynamic models, a number of interesting special effects are also possible, such as surface-conforming cast shadows for moving vehicles or aircraft.

Raised Roadbeds

Railroads are tactically important, visually significant, and navigationally invaluable. In the real world they invariably involve a raised roadbed which generally follows the lay of the land. An efficient, visually effective terrain-independent section of track and bed can be modeled which allows railroad track to be "laid" by DFAD or 2851 or a photo-interpreter at supersonic rates. Curved sections of various radii can be modeled and catalogued, and then instanced into place at the proper headings to accurately follow the source data. Where digital data is available, automatic processes can generate the railroad without human intervention, or the track path can be specified by "point-and-click" operations at a graphics workstation using aerial photographs.

Surface Conformal Pipelines

The surface conformal mode allows very efficient development of detailed, accurate above-ground pipelines. Again, minimal source data sufficient to depict the 2-D path of the pipe is required, and can arise from existing digital sources or be specified at a graphical workstation. Advanced instancing allows complicated paths to be constructed from a few modeled sections, with minimal stored data to perform the real-time assembly and display of the pipe as shown in figure 4. A single pipeline with direction changes every 50 meters (on average) can be defined using less than 4000 bytes of data per lineal mile. The underlying processes which convert the source data into the CIG form can produce, for example, the entire Alaska pipeline in under a minute.

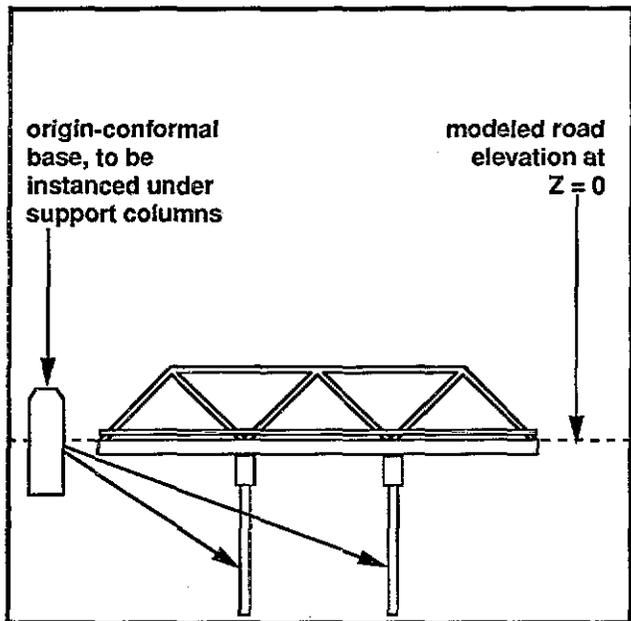


Figure 3a

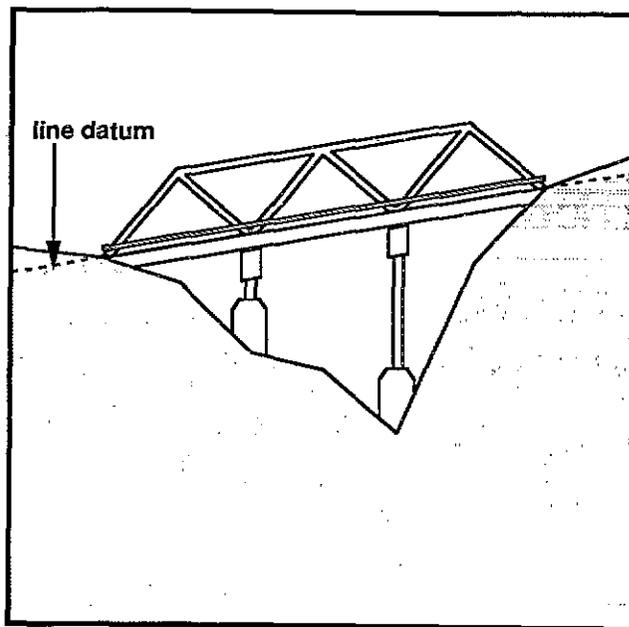


Figure 3b

Figure 3a shows a line-conformal bridge model, complete with superstructure and support columns. A single origin-conformal base, to be instanced under the columns, is provided separately. Figure 3b shows the bridge as it is conformed to the terrain. The bridge structure is sheared to align with the line datum while its support columns retain their vertical orientation. The bases sit on the terrain at the proper elevations below each column while the intersections of columns with bases, and bases with terrain, are resolved by the R-buffer. The resulting bridge model is terrain-independent, and will work properly wherever it is used.

Forest Canopy

Where actual forest density is too high to represent with individual trees, and perhaps as the lowest level of detail for forests which are built of individual trees, a surface-conformal forest canopy provides the proper occultation of embedded roads and clearings. The canopy is modeled relative to a flat plane, and includes the vertical development and textural decoration required for realism and height/speed cues. An hierarchical instantiation is used to build large, non-repeating regions, and signatures can be employed to contour regions of canopy to match the source data. When the canopy is applied to the terrain, it is conformed to follow the terrain contours (in addition to its inherent random vertical development), allowing realistic visual results with a compact stored model that can be rapidly developed from the source data.

Micro-topographic Detail

In many cases a feature contains geometry which depicts embankments, mounds or trenches of either natural or man-made origin, which are too small to be captured by the terrain-skinning process, but are visually and tactically significant. Examples might include the earthen bunkers associated with a SAM site, or dikes and drainage trenches around a landing area, either of which may need to be "relocatable" at model-generation time or by the

instructor in real time. The surface conformal mode allows terrain-independent models of these features to be developed and stored, and ensures that they will interface properly with the terrain when placed by terrain decoration processes, or by the real-time system. In addition, feature polygons can be flagged for decoration by the terrain photo-texture. This allows the use of feature polygons to provide small-scale enhancements to the terrain topography while preserving the continuity of the terrain texture. The entire scene feature, including mechanisms to modify the terrain, is terrain-independent and fully relocatable.

Economic Vehicle Dynamics

One interesting application of these conforming modes is to plant moving vehicles on the terrain, and keep them attached while under motion, without the real-time expense of computing and inputting Z values for each vehicle. In situations requiring high fidelity, the pitch and roll of vehicles could be properly computed by the real-time system from tire-contact terrain height feedback. In cases where less fidelity is required, the line and surface conformal modes can be used in the vehicle models to provide approximate visual behavior as the vehicles traverse sloped terrain. The real-time system then only has to control their lat/long positions. A train might be similarly modeled, and could be made to traverse long sections of straight,



Figure 4

The pipeline in Figure 4 is constructed automatically with numerous instances of a few modeled surface-conformal segments and joints, and a surface-conformal cast shadow. Joints are instanced appropriately and segments are positioned and scaled to stretch properly from joint to joint. The surface-conformal mode ensures that the pipeline hugs the ground, being clipped and folded continuously and properly as it snakes up and down over hilly terrain. The compact data structures used to construct the pipeline provide high compression of the resulting database.

(but not flat or level) track as a single dynamic vehicle.

4. Schumacker, R. A., 1980, A New Visual System Architecture. Proceedings of the 2nd IITEC Conference, November, 1980.

CONCLUSIONS

By achieving complete feature/terrain independence, a number of significant feature modeling and placement problems are avoided. Visual environment fidelity and density are greatly increased, while the amount of stored data is similarly reduced. The process of developing features and distributing them properly on the terrain is much faster, and excellent results can be achieved with completely automated approaches for mission rehearsal applications. Database development time and cost are reduced, and overall life-cycle costs for systems which must support continuing database generation are lowered. Powerful new hardware capabilities incorporated in the image generator perform the required feature/terrain marriage in real-time, allowing the generation and display of imagery with new levels of fidelity and realism.

ACKNOWLEDGMENTS

The ideas and techniques discussed in this paper have been and continue to be developed and validated by a number of people at Evans & Sutherland. We express appreciation to the people of CT Exploitation, CT Database Marketing Support, and Modeling Systems for their creativity and persistence in this effort.

ABOUT THE AUTHORS

Charles Clark has been with Evans & Sutherland for over eight years where he has provided support to the development and exploitation of high-end CIG and database technologies. Mr. Clark holds a Bachelors degree from the University of Utah.

Michael Cosman has been with Evans & Sutherland for over sixteen years, and has been associated with the development of high-end CIG systems for most of this time. He helped develop many of the database management strategies used in the ESIG product line, as well as some of the internal image-processing algorithms. Mr. Cosman holds a Bachelors degree from Brigham Young University.

REFERENCES

1. Cosman, M. A., 1989, A System Approach for Marrying Features to Terrain. Proceedings of the 11th IITSC Conference, November, 1989.
2. Costenbader, J. L., Bits and Pieces: CIG in an Instance. Proceedings of the Image III Conference, May, 1984.
3. Mayer, N. L. and M. A. Cosman, Enhancing the Computer Generated Illusion. Proceedings of the 3rd IITEC Conference, November, 1981.