CHALLENGES AND SOLUTIONS IN DEVELOPING A DYNAMIC TERRAIN ENABLED PC-BASED SOFTWARE IMAGE GENERATOR

Graham F. Upton
Diamond Visionics LLC
Vestal, New York

Brian Holmes
STRICOM
Orlando, Florida

ABSTRACT

Over the last several years there has been a growing requirement for Ground-based simulation training systems. As part of this requirement there is also a need for added realism within the simulation to provide, in real-time, the manipulation of a simulated terrain database in a physically realistic manner during an interactive simulation. Dynamic Terrain is not new to the Ground-based simulation community. However, current technologies require high-end computational platforms, are not real-time (30Hz), and are often cartoonish in appearance.

This paper will examine techniques to provide real-time dynamic terrain in a commercial-off-the-shelf (COTS) PC with commercially available graphic accelerator cards. The task of developing Dynamic Tessellation is challenging, especially on a PC-based system. Dynamic Tessellation provides the ability to deform terrain anywhere in the database in real time without the need for predefined deformable areas. Both Pre-Tessellation and Instantaneous-Tessellation approaches will be reviewed as well as the effects of soil dynamics and dynamic texture.

Dynamic Terrain is a requirement for realism for the maneuver forces in the Synthetic Environment. Specifically the application of dynamic terrain encompasses mine breaching, bomb damage, building damage, soil plowing and snow plowing. Specific applications of dynamic terrain are for the Grizzly Trainer, the Armored Vehicle Launched Mine-Clearing Line Charge (MICLIC), Track Width Mine Rollers and Explosive Standoff Minefield Breacher (ESMB).

As part of the results of this paper a PC-based Dynamic Terrain demonstration will be available as well as conclusions and recommendations to the methodologies employed.

AUTHORS BIOGRAPHIES

Mr. Graham Upton is the Director of Engineering at Diamond Visionics Co. in Binghamton, NY. He has 25 years of experience in systems engineering and management for flight and ground vehicle simulators. He holds a degree in Aeronautical Engineering from Southall College of Engineering in London, England. As a design engineer for Link Flight Simulation for over 21 years, he has been involved in the flight and ground vehicle products as well as in Air Traffic Control simulations. He has led R & D teams to develop new products in this fast-paced industry, and has managed both Army and Commercial programs. He has written many papers in areas of simulation and training.

Brian Holmes
Visual Systems Engineer for US Army Simulation, Training and Instrumentation Command
BS Electrical Engineering, University of Central Florida 1988.
MS Industrial Engineering, University of Central Florida 1998.
Brian has worked for STRICOM for 6 years, and prior to that worked at the Naval Training Systems Center (now NAWC-TSD) for 5 years. He was a project engineer for the A-6E Flight Simulators, and the Lead Systems Engineer for the Aviation Test Bed, Ft. Rucker, AL. He currently is working a Small Business Innovative Research (SBIR) Phase I contract on PC-Based Dynamic Terrain, as well as being the Visual Engineer for the Eighth US Army Visual System Upgrade (UH-60 and CH-47) and the Wolverine/Grizzly training systems.

CHALLENGES AND SOLUTIONS IN DEVELOPING A DYNAMIC TERRAIN ENABLED PC-BASED SOFTWARE IMAGE GENERATOR

Graham F. Upton
Diamond Visionics LLC
Vestal, New York

Brian Holmes
STRICOM
Orlando, Florida

INTRODUCTION

This paper examines three main components of Dynamic Terrain, namely Dynamic Tessellation, Soil Dynamics and Dynamic Texturing. These three represent some of the most challenging aspects of Dynamic Terrain especially when a solution is constrained to a PC-Based system. There are other components that make up a Dynamic Terrain enabled Image Generation System that are not addressed in the paper. These include, but are not limited to, Database Structures/Legacy Interfaces, ModSAF/OneSAF DT Integration, DIS/HLA Protocol Development and Intra-vehicle Communications. Although each one of these components play an important role, they are not discussed in this paper.

BACKGROUND

Diamond Visionics (DVC) was awarded a Small Business Innovative Research (SBIR) contract to study an approach for providing a low cost, PC-Based, realistic dynamic terrain enabled Image Generator. The Phase I SBIR contract is the study phase. SBIR

The three main components discussed in this paper, Dynamic Tessellation, Soil Dynamics, and Dynamic Texturing will be detailed in separate paragraphs. Dynamic Tessellation is by far the biggest challenge and will be discussed in detail first.

The platform for the Dynamic Terrain study and development is the PC. Whereas just a few years ago the execution of this task on a PC would have unthinkable and ridiculous it is now a very plausible option.
According to "Moore's Law" CPU complexity, and hence performance, doubles every 18 months to 2 years. In 2 years the potential upgrade of hardware could have a two-fold performance increase on an existing application with little or no changes in the software. Because of this fact no benchmarks have been given for the PC used in this study as it is such a moving target.

DYNAMIC TESSELLATION – THE ISSUES

Two major issues need to be addressed namely:

1. The ability to break up large polygons in the database in real-time to provide finer deformation to achieve realism.
2. The ability to manipulate vertices in the database in real-time to achieve terrain deformation.

DYNAMIC TESSELLATION – THE CHALLENGE

The task of developing Dynamic Tessellation is challenging, especially within the constraints of a PC-based system. Dynamic Tessellation is the ability to deform terrain anywhere in the database in real time without the need for predefined deformable areas. This was the single largest task of our SBIR Phase I effort. Two general approaches were identified that may potentially solve the problem. A preliminary trade study was performed prior to the Phase I award and a leading candidate was identified. The trade study was then revisited during Phase I SBIR.

DYNAMIC TESSELLATION - APPROACHES

Four approaches have been examined as shown in the Table 1 below:

Approach	*Pro*	*Con*
1)Pre-Tessellation	Avoid the technical complexity of real-time tessellation	Need to deal with huge amount of data
2)Pre-Tessellation of selected areas	Less data than above	Deformation limited to selected areas
3) Instantaneous-Tessellation of all the Database	Small amounts of data. More realistic tessellation	High complexity
4) Instantaneous-Tessellation of selected areas	Small amount of data More realistic tessellation	Must limit location in database to tessellate Deformation limited to selected areas

TABLE 1

Our effort, for the study focused on approach 3 – Instantaneous Tessellation of all the database. This approach was taken as a "selected area" approach would impose too many restrictions for the training exercise and a Pre-Tessellation approach would involve too much data being passed in real-time. Tessellation in order to provide realism is of very high importance to future projects requiring dynamic terrain capabilities.
The following section is a discussion of our findings and results for Dynamic Tessellation:

DYNAMIC TESSELLATION - METHODS AND RESULTS

Introduction

The purpose of dynamic tessellation is to allow a realistic deformation of terrain caused by the interaction between the vehicle and the terrain. In the most simplistic explanation, additional polygons are created in the localized area where such interaction takes place. This yields a smooth appearance of dynamic terrain movement as the vehicle deforms the terrain.

Normal Terrain:

The terrain in the database is made up of many polygons of various shapes, sides, and sizes. A section of this mesh may look like the following: (Figure 1)

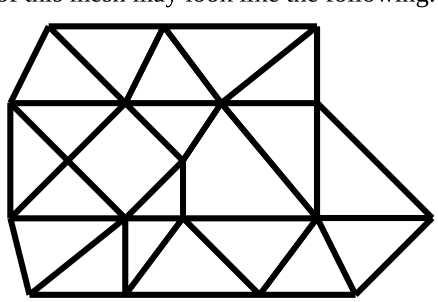


Figure 1: Typical polygon mesh of a database section.

A dynamically moving square box is located around the intersection of the vehicle and terrain. The diagram shows the point of intersection and what the square box may look like at any instance of time. (Figure 2)

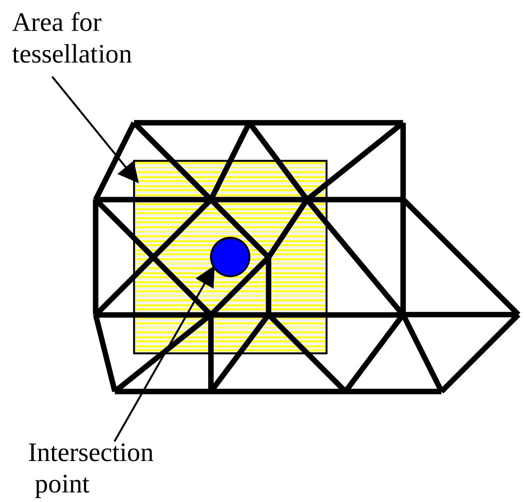


Figure 2: Area to be tessellated

Triangulation:

Next, all polygons located in the square box are checked for triangulation. All polygons are triangulated. Figure 3 below shows the same section of database where all polygons are now triangles.

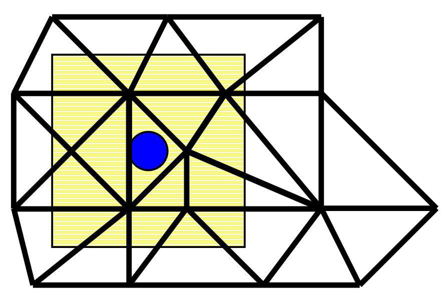
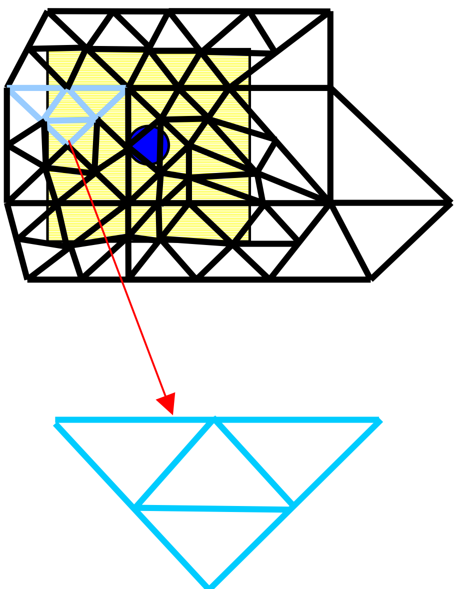


Figure 3: Triangulation of tessellated area.

Tessellation:

All triangles intersecting with the box are tessellated. This means that each original triangle becomes 4 new triangles. This is done by connecting each segment midpoint to the midpoint of the other segments in the original triangle. The result is shown below. Additional triangles allow for a large increase in resolution in the localized area since now there are four triangles where there used to be one. (Figure 4)



Detail of Tessellation

Figure 4: Tessellation

Tessellation continues recursively until all triangles are sufficiently small. Each triangle can be tessellated again if necessary. Actual terrain movement is accomplished by altering vertex positions, so a larger number of triangles will yield a smoother deformation. Figure 5 below shows an additional tessellation of one section.

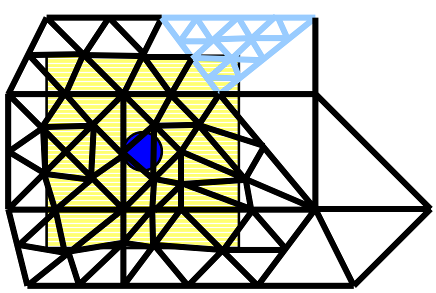
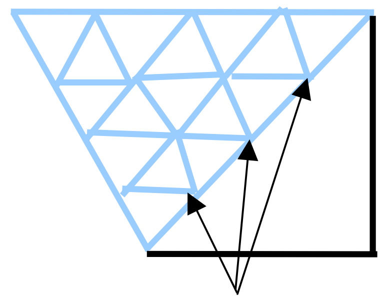


Figure 5: Further resolution by additional tessellation.

Tessellation poses a problem for the rendering database. It does not know how to resolve the new segments created by tessellation on the adjacent un-tessellated polygon. These new intersections are known as "T-sections". Figure 6 below shows the T-sections at the end of the tessellated region. These need to be resolved or anomalies caused by small gaps along the edges will occur.



T-sections.

Figure 6: Detail of figure 5 with the resulting T-sections.

Transient Layer:

A transient layer is built to alleviate the T-section problem and provide a smooth merging with non tessellated polygons. First, the triangle with T-sections is checked to see if there is more than one T-section. If there is more than one, it is tessellated as shown in Figure 7 below:

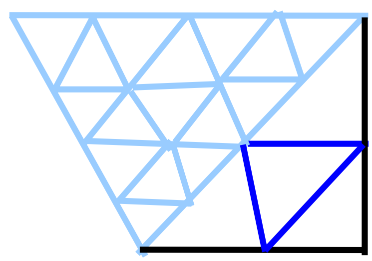


Figure 7: First step of transient layer building; tessellation if necessary.

More triangles have been created. The new triangles now only contain one T-section. The one T-section in each triangle is then connected to the vertice of the next outer polygon. The final T-sections are then removed by connecting to the vertice of the next outer polygon. This eliminates all the T-sections and completes the transient layer. The high density polygons and the low density polygons have now been connected seemlessly. (See Figure 8 below)

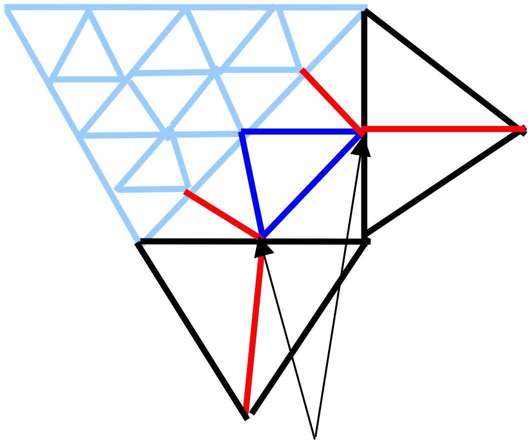


Figure 8: Resolving the final T-sections; transient layer now complete.

It's important to keep track of the T-sections in software. A map is utilized to associate T-sections with the triangles they belong to. When a triangle is tessellated, on each of its sides, a center point is generated. The three center points are used to construct sub-triangles. The program then maps the center points to their respective polygons. If the same point is stored to the map twice, the point will be removed from the map. After tessellation, only T-section points are left in the map. These points are utilized by their transient layer generation algorithms described above.

De-Tessellation:
As the vehicle moves away from the tessellated area, the terrain has been deformed and is no longer moving. The original area of interest is no longer the current area of interest. If the vehicle stays away for a predetermined amount of time or moves a certain distance the terrain begins to de-tessellate. During de-tessellation, the extra triangles that were created are now deleted.

When high density polygons are deleted, T-sections will also be generated. Those T-section vertices will be put into the T-section map following the same rule we discussed in the tessellation section. Since the transient layer generation algorithm will handle all the T-section vertices in the map without knowing where the T-section vertices comes from, de-tessellation becomes trivial as long as we have a robust transient layer algorithm.

Dynamic Tessellation – Conclusions
Dynamic Tessellation is an crucial component of Dynamic Terrain. To realistically deform the terrain caused by the interaction of a vehicle with the terrain is vital to effective training in the ground simulation environment. To be able to perform these operations in a PC-Based computer system in real-time is a significant step in providing a smooth appearance of dynamic terrain movement as the vehicle deforms the terrain.

Soil Dynamics – The Issue
Different soil dynamic models have been investigated and were addressed from the realism versus cost feasibility viewpoint. This is the main issue, particularly considering the PC-based requirement.

Soil Dynamics – The Challenge
The challenge in developing a soil dynamics model for Dynamic Terrain for a PC-based system is one of fidelity and realism. As soon as the requirements were defined, four approaches were evaluated as discussed below, prior to a method being selected.

Soil Dynamics - Approaches
Approach 1 - Physically-based Modeling
The Institute for Simulation and Training (IST) at the University of Central Florida (UCF) has done extensive studies on the physically-based modeling of an object using a mathematical representation of its behavior by incorporating principles of Newtonian physics. Dynamic soil models are required in animations and real-time interactive simulations in which changes in the natural terrain are involved. This method uses numerical algorithms with linear time complexity to meet the requirements of real-time computer simulation.

Strengths**:**
- This method will provide a very accurate soil model
- Allows for model extensions to include conservation of soil computations

Weaknesses:
- Not a true "real-time" simulation, would not run at 30Hz in a high end PC
- Very extensive and complicated model

Approach 2 - A Simplified Version of the Above Approach to the Required Dynamic Operation
This method would take the existing models from IST's studies and simplify it to perform only to the level required for realistic training for a dynamic operation. This is exactly the approach used in FLIR simulation for training. The fundamental requirement is realism for

the purpose of training, not the requirement of high fidelity modeling to recreate the precise physics of the problem.

Strengths:
- Less demand on computational horsepower
- Ability to run at true real-time (30Hz) on a high end PC

Weaknesses:
- Extensive work to "simplify" the model may be required

Approach 3 - A Table-Derived Computation Method
This approach would be to develop tables from a soil model in order to provide soil modeling without using extensive run-time algorithms.

Strengths:
- Low computational requirements
- Model accuracy maintained as tables are derived from off-line computations

Weaknesses:
- The Table sizes can be very large

Hybrid Approach – Using Tables and Real-Time Modeling
This approach would use a mix of both tables and extrapolation to provide a run-time model.
This is the preferred approach because it provides the best balance of run-time modeling cost and realism for training.

Soil Dynamics – Methods and Results
The following section describes an overview of the soil dynamics module. As described below, the soil dynamics module requires grided spatial elevation data in order to efficiently and accurately simulate terrain interaction. Rather than have vehicle dynamics interact directly with the visual database, it will instead interact with the grided elevation data via the soil dynamics module. The elevation data is generated in real-time by interpolating from polygon data provided by the visual database. The soil dynamics module computes the deltas in elevation at each post location based on the vehicles current action (e.g., digging, sinking, slipping, etc.). The final data set will then be supplied to the tessellation module to determine the best visual representation for the deformed terrain. The tessellation module will then generate visual LOD's for the newly deformed terrain and apply any texture modifications required to accurately represent the new terrain state.

In order for elevation grids to be efficiently generated, the visual database may need to be spatially grouped. This would allow the solid dynamics module to query the visual database for polygons in a specific area without having to traverse a complex hierarchy of database objects. Visual database information could be accessed simply by supplying the bounding area for the terrain being modified. Figure 9 shows a high-level view of a proposed architecture.

The diagram in Figure 9 illustrates the architecture for the dynamic terrain subsystem with focus on details necessary for incorporating a higher fidelity track soil interaction model.
The Track-Soil Interaction Module (TSIM) interfaces with the remaining systems through three interfaces. Configuration parameters control the fidelity, model, and other similar parameters of how the TSIM operates. Queries to the "terrain database" are structured around the concept of a grid of posts that contain elevation and soil properties data. The TSIM provides a rectangular area that surrounds the track along with a desired density and expects the data associated with the terrain at the specified elevations posts within the grid. The result of the "mini-simulation" in TSIM is a new grid of the same size but different elevation and spatial placement of the elevation posts. This information is then passed back to the original database and also piped to the network for transfer to other nodes.

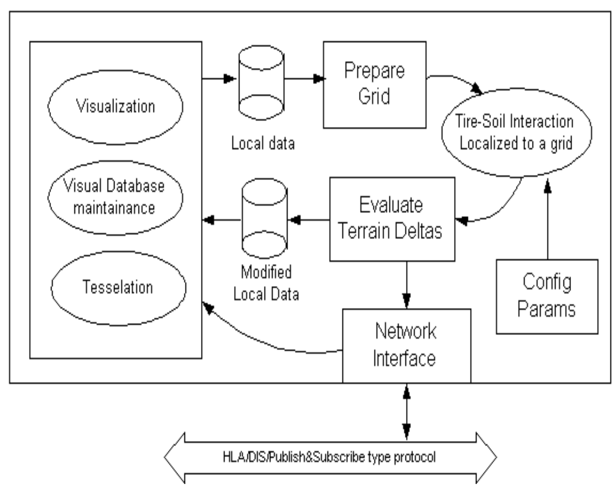


Figure 9

A more detailed functional specification of the block diagrams is provided below.

Prepare Grid

Given the grid geometry, this module calculates the elevation posts by interrogating triangles provided by the database system. In addition, based on the texture or other tag information associated with the triangles, this module determines the necessary surface properties to complete the information needed for each terrain post.

- The terrain database system should be able to provide a small set of triangles that represent the existing terrain near the tire/track.
- The technique by which texture or tag information is converted to soil properties is to be determined during the early part of this project. Some of the most feasible options include using a lookup table that maps texture name to surface properties, explicit storage in a different database, or determining the surface properties by determining the actual pixel color and mapping that color to surface properties through a table.

Evaluate Terrain Deltas

This module is responsible for retrofitting any changes to the terrain grid back into the required representation by the visualization/tessellation module. In the process of doing this, changes that apply directly to the visualization module are queued for transfer to any other nodes in the network that are using the same database.

Network Interface

This module is responsible for receiving terrain updates from the current node and piping it into the network using a desired protocol (most likely using an HLA compliant architecture). It is also responsible for receiving terrain updates from other nodes and propagating these changes to the local database. The current idea is to propagate the external changes directly to the visualization engine. Changes to the actual algorithm used for determining terrain modifications may necessitate the modification of this approach.

TSIM

Real-time track-soil interaction capabilities are based on the project requirements. Information about the vehicle, soil, and environment will be used within the TSIM to calculate forces generated at the track-soil interface. These forces will be used by the vehicle module to determine vehicle kinematic states for the next simulation point. In a similar fashion, interface forces will be used by the soil module to determine changes in soil parameters. These changes are of different nature, including mechanical changes due to deformation, database terrain heights, visual appearance, etc.

Track Soil Dynamics

TSIM will include internal capabilities to different levels of complexity limited by the real-time requirement. Further enhancement of the models will be possible in future implementations.

1. Move Dirt-Scraping: A simplified theory of passive earth failure will be utilized to represent the effect of moving soil and to represent the impact on the dynamic characteristics of the vehicle. The force on the blade while moving dirt can be obtained by integrating the passive earth pressure function over the area of the blade.
2. Digging and Plowing: Most plowing models developed in the past consider the steady state forces required to move the soil. Non-steady state forces, however, occur during operations involving variable soil depths and this can affect the payload and ride performance of the vehicle. This resistance force is expressed as a function of blade depth. Information will be supplied to the vehicle system so the interaction is captured in real time. The force generated at the equipment-soil interface during digging operations will be calculated as a function of the geometry of the contact and the dynamic information. Plowing models developed at WES will be considered for implementation.
3. Bombing and bomb damage: Based on the intensity of the blast, pressure levels acting on the soil can be estimated. Based on these levels cratering effects and soil motion could be calculated based on a combination of empirical and analytical considerations. The cratering effects will be used to modify terrain information visually within the interactive real-time environment.
4. Slippage-Stuck in Mud: Bekker's soil model will be used in combination with empirical relationships to predict vehicle motion over deformable terrain. This approach will help predict vehicle motion, slippage, trafficability and mobility.
5. Trafficability: Due to physical obstacles and extreme rut depth on the deformed soil due to multiple passes, vehicle systems might not be able to traverse certain terrain areas. A combination of discrete indicators, due to obstacles, combined with Bekker's soil deformation calculations, will be used to determine the level of trafficability for a given vehicle configuration. WES models for trafficability will be included in this capability.
6. Mobility: NRMM (NATO Reference Mobility Model) defines the Mobility Index (MI) using an empirical relationship for a tracked based on a number of vehicle design parameters including, gross vehicle weight, track load, track width, contact pressure, engine and transmission

characteristics. The Vehicle Cone Index (VCI) parameter that represents the minimum soil strength of the critical layer that is required for a vehicle to traverse number of passes successfully, is related to the MI through an empirical relationship. This measure can be used as a preliminary indicator of Mobility of vehicles over deformable terrain.

7. Feedback to vehicle for vehicle dynamic changes: Vehicle motion prediction over a deformable terrain could predict the state of the vehicle that can be used as a feedback to the system. The interface between vehicle and soil during all aspects of the dynamic interaction will be defined and exercised as needed. Mostly this interaction is given in terms of forces generated at the interface and passed to the vehicle subsystems to predict the state of the vehicle in the next computational point.
8. Soil types: Different types of soils will be covered as a function of the soil parameter data available. Bekker's parameter data available in the literature for at least 4 different types of soils will be included within the environment, with the ability of extending the library if data becomes available, e.g., Clay and Sand.
9. Soil surface: Homogeneous soil surface models will be modified to accommodate discrete soil elements and different layers. Surface roughness will be considered as an important factor to differentiate soil surfaces. Frequency spectrum of the soil surface will be included to provide corrections for frequency dependent factors.
10. Flood effects: Excess of water content in soils produces radical changes in the ability for vehicles to develop traction. An assessment of available modeling techniques for this type of condition will be made to determine basic capability.
11. Volumetric coefficient: Several soil attributes are better described using volumetric information. A number of coefficients based on volumetric ratios are available to describe soil properties (void ratio, porosity, moisture content, degree of saturation, etc.)
12. Mine Breaching: The modeling of the interaction between the vehicle and the mine during mine breaching maneuvers involve a combination of factors.
13. Breaching of wire, posts, rubble and ditches: This involves behavior on uneven terrain with non-homogeneous terrain properties.

Soil Dynamics - Conclusions

The effective use of soil dynamics computations provides both the realism and training fidelity in the dynamic terrain environment. By modeling several different soil types and soil surfaces in the simulation the feedback to the driver is provided and the realism generated provides effective training necessary for many ground-based training systems.

Dynamic Texture – The Issues

To determine the feasibility of different Dynamic Texture approaches in order to achieve realism. Using Dynamic Texturing methods, many visually complex effects can be produced without excessive computational overhead".

Dynamic Texture – The Challenge

Implement different Dynamic Texture methods to achieve realism and low cost.

Dynamic Texture - Approaches

Three approaches were studied.

1) Texture Swapping
2) Texture Manipulation
3) Texture Overlay

To determine the feasibility of different Dynamic Texture to achieve realism. As terrain surfaces are deformed, the visual appearance should change to match the soil properties of the visible area. By utilizing texture mapping, many soil effects can be produced that use significantly less processing than a polygon based approach. Swapping or manipulating the texture on a polygon can give the appearance of complex visual changes without trading speed in other areas such as networking.

Dynamic Texture Approach #1: Texture Swapping

In this method, the real-time database would maintain a table of textures for different soil and surface types including grass, sand, clay, mud, and as many other variations as needed depending on the fidelity required. Polygons in the database would keep an index to the texture in the table that best represents the current surface or soil type for that polygon. When deformation occurs and a polygon's surface or soil type changes, that polygon's index into the table would also change to match the new state. The texture would be picked based on a three dimensional table indicating what soil type is located at each location in the database.

Strengths:

- Very little run-time overhead. This method only involves quick lookup into tables, so no computationally expensive actions are performed.
- With proper selection of textures, the effect can appear very natural.
- This method has been partially implemented and tested at DVC.

Weaknesses:

- Performance for memory tradeoff. Because this method is table-driven, it could potentially require a large amount of texture memory (a texture would be required for every type of surface and many combinations in between).
- The texture change must be mapped for an entire polygon. Because of the nature of graphics systems, a single polygon can usually only have a single texture. This means that for a realistic and detailed effect, the polygons must be sufficiently small. This incurs a computational burden on both the tessellator (if done in real-time) and the graphics software/hardware because a large amount of polygons would be required.

Dynamic Texture Approach #2: Texture Manipulation
With Texture Manipulation, rather than swap the texture that a polygon is using, the texture itself would be modified at the pixel level. A pallet of small textures would be used for modifying the main textures. When deformation occurs, the texture of an affected polygon would be updated by first determining what appearance the polygon should now have. That information would be contained in the 3-D table described in the previous method. From that, a texture would be chosen from the palette that best resembles the deformed surface type. This texture would be used to "paint" onto the polygon's texture.

Strengths:

- More accurate. Because the texture is manipulated directly, more accurate affects could be achieved than with Texture Swapping.
- Fewer polygons would be needed than with Texture Swapping.

Weaknesses:

- Texture Manipulation requires special hardware support not often found in PC based graphics accelerators.
- Large amounts of texture memory (more than with Texture Swapping) may be required.
  This method has not yet been proven at DVC.

Dynamic Texture Approach #3: Texture Overlay
This method is an extension of the Texture Swapping method. With Texture Overlay, location specific deformation, such as tire tracks, is achieved by overlaying a partially transparent texture onto the affected polygon by using multiple rendering passes. Once the polygon has a pre-determined number of overlays on it, a texture is chosen from the texture table (described in the Texture Swapping method description) that best matches the current state of the polygon. This helps to keep the number of overlay effects down to a manageable number.

Strengths:

- Almost as accurate as Texture Manipulation, but no special hardware is required.
- Less texture memory required than with the other two methods described.
- Little computational overhead required.
- This method has been used and proven in other systems developed by DVC.

Weaknesses:

- Because of the multiple passes used in this method, it would be as though more polygons were being rendered, putting more demand on the graphics accelerator.

This dynamic texture capability (#3) was originally developed for the Laparascopic Surgical Training visual system that Diamond Visionics developed in 1998. This method provides realistic effects on organs and soft tissue (such as specular reflection). The realism of organs and soft tissue effects has impressed many medical specialists, including Dr. Satava of Yale/DARPA during his evaluation. We believe this method can also be used to provide texture effects in a dynamic terrain simulation.

The texture overlay method provides very realistic effects while using less texture memory from the graphic accelerator than approaches one and two. Also little computation overhead and no special hardware is required.
It is our conclusions from this study that dynamic texturing using the texture overlay method is the approach to use, in that it provides realism at low cost.

Dynamic Texture – Methods and Results
The Textured Polygon Overlay method will be implemented in Phase II. With this technique, deformation will affect not only the topography of a database, but its visual attributes as well. By applying layers to an affected polygon, the appearance will change dynamically as deformation occurs. For example, as the blade of a Grizzly might cut into a grassy area, not only would earth be displaced (modifying the underlying geometry), but the appearance of the deformed area would transition from

grass to dirt to make the deformation more obvious to an observer.

The technique of adding overlay polygons is simple, provides very realistic results, and can be implemented using low-cost hardware. Any card capable of alpha blending and can be utilized. Most consumer-level boards today are more than capable. In fact, some consumer-level boards are capable of rendering multiple layers in a single pass. So without a hardware limitation, it is purely up to the software to handle the application and management of overlays. This requires a database hierarchy that allows for the addition and removal of overlay polygons. Early in our studies, this capability was implemented with DVC's database hierarchy with no impact on performance. While this feature may not be present in some current-day software IGs, it is not seen as a significant challenge to add support for such a mechanism.

Not only must the hierarchy be able to support overlays, but there must also be information in the database describing the attributes of underlying terrain. This information needs to be present for both the dynamics aspect (e.g., clay "behaves" differently than sand) and the visual system (e.g., clay looks different than sand). To add the appropriate overlay to a polygon, enough information must be preset to determine which texture to apply for a given overlay. A library of textures will be available representing each type of terrain that needs to be supported. When deformation occurs, the new depth of the polygon along with other information will be used to select the closest representation from the texture library and an overlay is applied utilizing that texture.

If a polygon's attributes ever transition completely from one terrain type to another, then layers can be removed to reduce unnecessary overhead. For example, if a Grizzly blade was digging and its edge fell in the center of a polygon, then half of the polygon might require a dirt texture and half might still require the original grass texture. This would require adding a dirt textured overlay to the underlying grass texture. If the Grizzly were to make a second pass and deform the remaining half of the polygon such that all of the polygon should be represented with a dirt texture, then the grass polygon could be removed or flagged so that it would no longer be processed. Only the dirt overlay would be processed and be visible.

Dynamic Texture - Conclusions

The appearance of the database as the vehicle traverses over it will be greatly enhanced using dynamic textures. Using the overlay method, by applying layers to the affected polygon, the appearance will change dynamically as deformation occurs.

ACKNOWLEDGEMENTS

Information provided by the University of Iowa 's Center for Virtual Proving Ground Simulation on Soil Dynamics was a valuable contribution to this paper. The university has done extensive research on this subject and their contribution was appreciated.