ALGORITHMS FOR RUN TIME TERRAIN DEFORMATION

STATUS REPORT

SBIR Phase II Contract FA8650-05-C-6537 Topic AF04-064 Feb. 8, 2006
Prepared for AFRL/HEAE, Air Force Research Laboratory
Attn: Steve Stephens, 6030 South Kent Street, Mesa, AZ 85212-6061
Tel: (480) 988-6561 ext. 146 e-mail: steve.stephens@mesa.afmc.af.mil

Prepared by Computer Graphics Systems Development Corp.
Attn: Roy Latham, 2483 Old Middlefield Way #140, Mtn. View CA 94043
Tel: 650-903-4922 Fax: 650-967-5252 e-mail: rwl@cgsd.com SkypeID: rlatham001

Project Abstract

The goals of Phase II are to develop and implement algorithms for a real time mission rehearsal simulation which will deform the terrain database to match target data. A correction function c(x,y) that is added to every vertex in view. The correction function must (1) adjust the terrain surface to meet the specified features, (2) appear smooth and continuous so that the adjustments appear natural, and (3) do not distort aspects of the database that must be preserved. The implementation is to be in C++ and compatible with Open Scene Graph. The code will be placed in the public domain in keeping with an open source philosophy.

• *Previously Completed Work*

The Phase II contract was signed on April 8, 2005 and work started immediately. ▪ A kickoff meeting was held in May ‘05, and the program plans were reviewed. ▪ Don Burns, one of the originators of Open Scene Graph was added to the project team. ▪ Completed the first draft PDL that links to OSG and deals with database layers and tiles. ▪ The SHAPE file input software was written and debugged. An algorithm to find the shape of the object contact to the terrain was devised and programmed, with the terrain modified abruptly. ▪ The comprehensive algorithm for real time terrain modification was completed and documented, and is being implemented for testing.

• *Work Accomplished This Reporting Period*

We have substantial progress to report this month, as a result of the past two month’s work. The algorithms for the interpolated function smoothing and for iterative adaptive tessellation have been coded and are well into debug. These are the algorithms that build the smooth hills and valleys in real time to adapt the terrain to the polygon target data. In addition, in accordance with the Phase II contract requirements, we have prepared a draft of the first non-proprietary interim summary report. The draft is offered for the 30 day comment period before being made final. Roy and Don Burns have also submitted to the IMAGE Society a proposal for a paper on the Phase II dynamic terrain algorithms. The IMAGE conference will be in Scottsdale in July.
http://www.public.asu.edu/~image/EVENTS/06IMAGE/06.html

The interim summary report fully discloses the project objectives, the overall approach, and the details of the algorithms we are implementing. This broad non-proprietary disclosure is in keeping with our open source commitment for the project. It also indicates, along with the paper proposal, that we are confident that the algorithms are fundamentally sound and are ready to be discussed publicly.

In preparing the IMAGE paper proposal, we discovered that the algorithms we have intended for dynamic terrain modification happen to also apply to implementing local illumination sources. When a local illumination source, like a flare, is over terrain, the illumination associated with each polygon must be modified. If the flare happens to be over the center of a large polygon, the distant vertices of the polygon will have only a small change in illumination and the bright spot in the middle of the polygon may be missed entirely. The algorithms proposed for the terrain could be used to insert a vertex under the flare and to tessellate the polygons in the terrain to ensure a smooth variation in illumination. Similarly, the algorithm handles multiple overlapping illumination source modifications and multiple levels of detail. The algorithms allow each light source to have a different illumination pattern. In each case, the algorithm would modify the polgon vertex illuminations rather than the polygon terrain elevation that are our present concern.

IMAGE Paper Proposal

Here is the extended abstract Roy and Don submitted to IMAGE. It is a good summary of the project approach and algorithms:.

> The problem is to adjust the elevation in a terrain database for visual simulation so that the terrain surface will smoothly fit a set of newly described *targets*. Targets $t_1 \ldots t_n$ are geometric objects described by a set of polygons, with some of the polygons in contact with the terrain. Targets may be point features described by a single coordinate (x_i, y_i, z_i), lineal features described by a connected sequence of points, or areal features described by a set of polygons defining the area in contact with the terrain. Traditionally, an approach has been to change the elevation of the targets to match the terrain, or to reprocess the database off-line to match the targets. For mission rehearsal, however, it is desirable to have the terrain processed in near real time to match the target data. The approach adopted here is to define a correction function $C(x,y)$ defined over the whole database such that when the correction function is added to every corresponding point of the terrain in the original database, the result will exactly conform to the targets and will vary naturally between targets. If there is a point target above the flat surface of the original database, for example, the correction function will provide a hill of the correct height to match the target.
>
> The method works as follows. The space of $C(x,y)$ is triangularized independent of the underlying database. A function corresponding to the shape of the smooth fit is defined independently for each target vertex. Nominally, a two-dimensional Gaussian function is used for each target vertex with the peak equal to the desired correction and the width proportional to the height, so for the *i*th target vertex, we define $f_i(x,y) = a_i * \exp[- d^2/k^2 a_i^2]$ where $d^2 = (x-x_i)^2 + (y-y_i)^2$ and k is a slope constant. For each point within each triangle of C, the value of the correction to be made at the point is a weighted sum of the three functions associated with respective vertices of the triangle, i.e. $C(x,y) = w_1 f_1(x,y) + w_2 f_2(x,y) + w_3 f_3(x,y)$. The weights sum to one and are inversely proportional to the distances to their respective vertices.

C is defined over the whole database, but it is only evaluated at selected points in the underlying terrain database. The correction is first computed at the vertices of the terrain polygons, and then at the midpoints of the sides of the polygons. If the midpoints all lie within a small tolerance of the plane of the corrected triangle, the triangle is ready for display. If any of the midpoint corrections are not within the tolerance, then the midpoints are connected to divide the original triangle into four new traingles. The procedure is iterated until the surface has been approximated smoothly.

An advantage of the correction function approach is that it lends itself well to application in real time. The correction function itself must be computed globally, but it is likely to be small in comparison to the terrain database, because the number of targets is likely to be small compared to the number of terrain polygons. The actual terrain modifications can be applied to each area block when it is retrieved. An additional advantage is that virtually any functional form can be associated with each vertex, so that variations in terrain roughness can be simulated. The method can be applied to terrin modifications made by bomb cratering or earthworks, as well as to target data. A modification of the method could be used to modify the illumination in a database rather than the elevation; a local illumination source would result in dynamic subdivision of the terrain polygons to achieve a smooth illumination effect.

• *Summary of Status*

The project is on schedule. The status of tasks is summarized below:

ID	Description	9/8/05
Task 1	Research & verify the timeliness of the full-scale Algorithm/technique	100%
Task 2	Verify the accuracy of the full-scale algorithm or technique	95%
Task 3	Design, code and test the full-scale algorithm	60%
Task 4	Develop a web site for the release of open source code	55%
Task 5	Examine the compatibility of the open source code with the existing IG hardware	20%
Task 6	Demonstrate the prototype	12%
Task 7	Write Interim Report(s)	40%
Task 8	Write Final Report and Summary Report 0%	0%

• *Problems*

No significant problems or information that might impact schedule have been encountered in this reporting period.

• *Interim Results*

The draft of the first interim non-proprietary summary report is attached for review.

• ***Recommendations and Proposals***

There are no recommendations or proposals as a result of efforts in this reporting period.

• ***Summary of Future Plans***

On the software side, it is time to clean up and document the code that has recently been debugged. On the algorithm side, we will start looking at implementation of line features (like roads) and possible problems associated with level-of-detail switching.