INTERACTIVE SAND-COVERED TERRAIN SURFACE MODEL WITH HAPTIC FEEDBACK

Václav, PURCHART[1], Ivana, KOLINGEROVÁ[2], Bedrich, BENES[3]

[1,2]Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, 306 14, Plzeň, Czech Republic

vpurch@kiv.zcu.cz, kolinger@kiv.zcu.cz

[3]Department of Computer Graphics Technology, Knoy Hall of Technology, 401. N.Grant Street, 47907-2021, West Lafayette, IN, U.S.A

bbenes@purdue.edu

Abstract

We describe a method for an intuitive haptic rendering and modeling of GIS terrain models represented as triangular irregular networks (TIN). The TIN terrain model does not allow representation of concavities and overhangs but it is sufficient for most GIS applications. Model editing is achieved by a set of virtual tools with various shapes which are controlled by a haptic device with force feedback. The haptic feedback makes manipulation with the terrain model much more intuitive than by the mouse, because it operates in 3D and it also provides better control with the force. The haptic device is point-based, so the forces provided by different tools must be mapped to a point. This, in effect, helps eliminating certain vibrations that could be caused by numerical errors and insufficient scene sampling. Our approach allows an adaptive adjustment of the level of detail of the model according to the terrain properties and user requirements. Haptic visualization in GIS models has been already used but such solutions are not common. Unlike existing approaches, our solution uses a general shape of the tool, it allows interactive editing of the TIN model, and it utilizes parallelism and coherence between consecutive states of the model that make haptic visualization faster and smoother.

Keywords: geovisualization, haptic visualization, triangular irregular network terrain model, interactive manipulation

INTRODUCTION

Various approaches to exploration, visualization, and intuitive editing of GIS data have been developed last years, but there are still many open problems. One of them is the haptic visualization and editing of terrains. The haptic visualization has the potential to provide additional information to the user (sense of touch) in addition to visual output only. And, unlike the mouse or keyboard, the haptic devices provide native 3D manipulation, which is also more intuitive. Haptic devices were very expensive but, nowadays, they are becoming affordable for general public. New falcon devices, such as Falcon Novint are priced at $250 in December 2011.

The most commonly used terrain representation in existing haptic applications is digital elevation model (DEM) based on a regular grid or volume data. Volumetric-based or grid-based methods are memory consuming, because the resolution of the grid must correspond to the smallest detail of the the model. To the best of our knowledge, there is no suitable haptic method which works natively with TIN and allows both editing and simulation of physical processes on the terrain.

We have developed a haptic visualization method which works natively with TIN and allows interactive editing of the terrain and erosion simulation using a haptic device. User can imprint a tool shape into the terrain or he/she can drag a tool across the terrain. The physical model of the terrain represents sand but our approach is general enough to simulate different materials. All changes in the terrain are smoothened by thermal erosion which can be also controlled, e.g., allows varying the sand humidity. This limitation of our technique is that it does not allow shapes with overhangs. The goal of our haptic visualization is to be realistic, easy to use, and reasonably fast to be usable.

Our method uses the Delaunay triangulation (DT) [Berg 97] for the TIN representation and the Constrained Delaunay triangulation (CDT) for editing [Sloan 93]. We have created various editing tool shapes, such as rectangle, circle, sphere, ellipsoid, and man foot print. As the haptic device provides point-based feedback, all forces exerted on our shapes must be mapped to a single point. Because of different frame rate requirements, our implementation uses multiple CPU threads; one for the graphics feedback and one for the haptic visualization.

The paper is structured as follows. The following section will briefly introduce the existing work. Next, our method is described. Next section summarizes our experiments and the last section concludes the paper.

RELATED WORK

Haptic visualization methods of geospatial data are quite rare. The previous work is focused on haptic exploration and haptic visualization of volumetric data sets or point sets.

Faeth at al. present a framework for haptic visualization of geospatial data using X3D in [Faeth 08], their framework also allows making deformation of the model. They use the ArcGIS program to convert various GIS formats into X3D indexed mesh and grid image. Vertices changed their heights after deformation and, additionally, they are rearranged according to Generalized Chainmail deformation algorithm [Li 03]. It results in a cloth-like behavior of the terrain. "Soft" and "hard" materials are distinguished. Force response can map terrain properties.

Guo at al. [Guo 04] present a method for interactive haptic manipulation of point sets. User edits are described with implicit functions. The model consists of two representations. The first is a global scalar field which stores a distance from the zero plane. The other representation is a voxel regular grid. A global mass-spring system is created between voxels. This method allows simulation of voxel-based physics processes. Haptic visualization uses the exponential weighted average extrapolation scheme to smooth the resulting force.

Beneš at al. have described an algorithm for haptic visualization of the sand surface in [Beneš 06]. The user can interactively edit the terrain using a haptic device and the terrain changes are smoothened by erosion. The terrain model is described as a regular height-field. This is significantly better than voxel-based approaches in terms of memory requirements.

Grid-based methods have high memory requirements and have implicit limits on the details they can represent; TIN-based methods do not allow editing but only a deformation of existing terrain. No new vertices can be added to the TIN. Our method provides such features.

PROPOSED METHOD

Our algorithm consists of three main parts: (i) the geometric model, (ii) physics simulation, and (iii) haptic interaction. The input of our algorithm is a set of 3D points [x, y, z], where x and y defines the position in terrain and z is an elevation of the terrain. The points are given by a height map, by an image, or a bivariate function. The output of this process is the TIN modeled as a Delaunay triangulation of the given set of points, which we use for physics simulation. DT is computed using the incremental insertion algorithm [Berg 97]. This algorithm is very useful for our purpose, because it can add vertices into the triangulation on the fly. So the terrain modification could be easily handled. Terrain could be modified by erosion or by haptic device. As a testing algorithm for physics simulation, we have developed an algorithm for thermal erosion in [Purchart 11].

We use a set of virtual tools for the haptic manipulation. The tool consists of a set of control points and edges between these points. The edges are used as constraints for the CDT. The terrain is deformed inside the virtual tool shape according to this shape and according to the user requirements. The height of these points adapts to the terrain height in the current tool position when the tool is imprinted into the terrain, (see

Figure 1). The amount of material pushed away by the tool is defined as a difference between heights of the tool depth level, and height of the control points in the direction of tool movement.

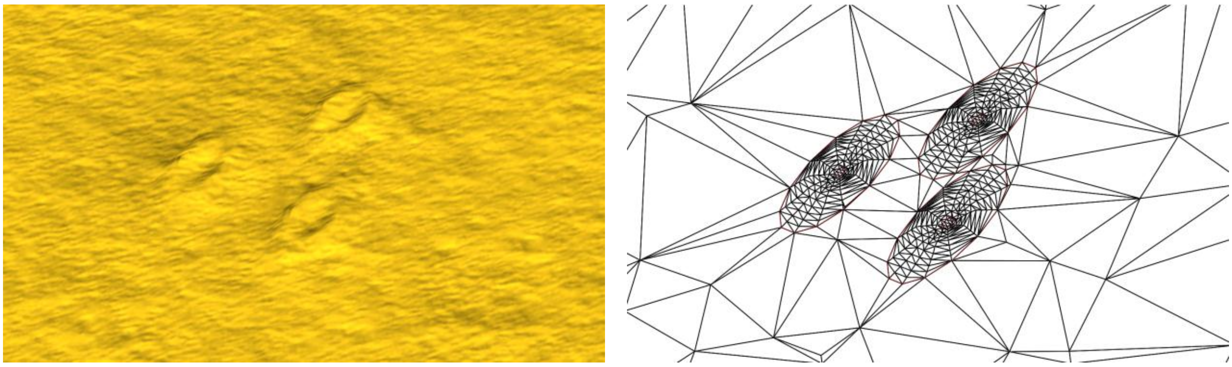

Fig. 1. Demonstration of terrain editing. Left: Three imprints of elliptic shape tool on a nearly flat terrain covered by yellow sand. Right: The same scene displayed as a wireframe model. Border edges of ellipses are constrained.

Haptic visualization

The haptic manipulation is initiated by a haptic tool handled by the user and it causes geometric changes of the TIN. The erosion algorithm is applied to the changed vertices. The challenge is to maintain a correct haptic feedback when the TIN is changing its geometry even though the TIN is not in a stable state for a while.

The graphics visualization refresh rate is about 60Hz that is way below the required refresh rate for haptic rendering that is about 1,000Hz. The TIN is constantly changing because while the user edits it and the erosion smoothes it, the haptic thread must update its data only if the slow graphics and geometry thread reaches a stable state. The only stable state is the end of graphics thread loop (all terrain edits have been performed and the TIN has been repaired).

The data, the update of which is fast enough to be directly handled in the haptic thread, are the position of haptic cursor and the force feedback. The haptic device sends a 3D position of the cursor as a point [*x*, *y*, *z*] 1,000 times per second. The virtual tool is correctly updated based on this position. The tool movement (direction as well as its velocity) is acquired as a difference of two consecutive tool's positions. We must set the force feedback F_{out} to the device (3D vector in newtons). This value must be also set 1,000 times per second. The device acts in the direction of force vector with the given force amount.

As it is mentioned above, visualization and haptic threads have different refresh rates. Usually it is 60Hz versus 1kHz. So the haptic thread makes $1{,}000/60 \approx 17$ iterations for each graphics render. We call this number of iterations it_p and we assume it to be constant. Slow communication with the haptic device would be suboptimal – the haptic perception would change very slowly, only with 60Hz refresh rate instead of 1kHz. The force feedback feels runny for low framerate. The speedup in our approach is achieved by interpolation of the forces between two consecutive frames. Our algorithm stores the old force feedback state before an update and the new one. The force is interpolated in time in each haptic iteration, so that the resulting force feedback is smoother.

Force feedback components

The force feedback F consists of two forces (see Figure 2). The first one is a penetration force F_p and it depends on the tool depth level – a deeper penetration results in a larger force which acts strictly in the vertical direction for sand and our set of tools. Sand settles down quickly because of erosion, so we can expect a nearly horizontal surface [Benes 06]. It is impossible to feel the difference between our simplified strictly vertical F_p and the real, physically correct F_p.

The second part of the force feedback is the friction force F_r, which is physically defined by Equation 1. It depends also on the penetration depth in the direction of tool movement and on the speed of tool movement – a higher velocity means a larger force which acts against the tool movement direction.

$$F_r = \mu \cdot F_n, \quad (1)$$

where μ is the coefficient of friction, which is an empirical property which differs for each material, and F_n is the normal force which is perpendicular to the surface and acts against the tool. In our method we use an approximation of this physical law as explained below.

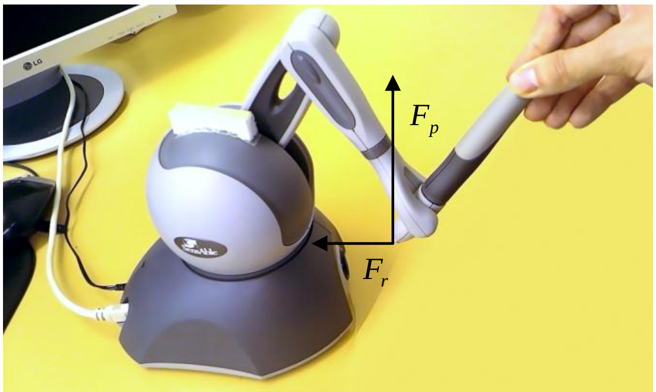

Fig. 2. Low end haptic device (Phantom Omni) used for the test.

The force feedback is a bit more complicated as the user editing tools can have an arbitrary shape rather than a single point. We have created tools with various shapes, such as a rectangle, a circle, a sphere, an ellipsoid, and a human footprint, but the haptic cursor is a single point for the haptic device we use. We need to map all the forces on the circumference of the tool into the single point. This can be roughly computed as a mean value of forces in all control points of the tool (Equation 2). Such approach is not physically accurate for non-circular tools, but it gives a good force feedback for tools we have tested.

$$F = \frac{\sum_{i=1}^{N}(F_{ri}+F_{pi})}{N} + F_G, \quad (2)$$

where F_{ri} is the friction force for the i-th control point, F_{pi} is the penetration force for the i-th control point, N is the number of control points, and F_G is the force caused by the so called gel effect.

The force F_G is an additional horizontal force. It is a haptic effect [Lin 08] known as a “gel effect”. This force depends on the velocity of the tool and acts as a tangent viscosity; higher velocity causes that a larger force acts against the moving direction. This force very well describes behavior of the real sand and simulates a sand resistance against the fast movement.

Force computation

Let z_{tool} be a current tool depth level, c_i is a position of *i-th* control point of the tool, z_{ci} is a z coordinate of *i-th* tool control point, P_{act} is a current position of the tool, P_{old} is a previous tool position, and T is a vertical force tolerance. If the tool is in vicinity of the terrain surface (tool's distance from the surface is smaller than T) the force starts to act.

The height difference for *i-th* control point is computed by Equation 3. Note that the points coordinates have to be normalized into the interval [0; 1].

$$\Delta h_i = z_{tool} - z_{ci}. \quad (3)$$

The force feedback starts to act only when the inner part of the tool (a tool depth level) is below the terrain level or slightly above it to avoid numerical errors ($\Delta h_i > -T$). The direction dir_i of the tool movement from the tool position to the *i-th* control point is described by Equation 4:

$$dir_i = \frac{c_i - P_{act}}{|c_i - P_{act}|} . \quad (\mathbf{4})$$

The friction force is computed by Equation 5:

$$F_{ri} = -dir_i \cdot \Delta h_i \cdot \frac{F_M}{2} , \quad (\mathbf{5})$$

where F_M is the maximum force allowed by the haptic device (10N in our case).

The penetration force for the i-th control point is computed by Equation 6:

$$F_{pi} = \Delta h_i \cdot \frac{F_M}{2} \quad (\mathbf{6})$$

To obtain the "gel effect" force, we compute a tool velocity v between frames by Equation 7. Note that we expect a constant time interval between frames (equal to one).

$$v = P_{old} - P_{act} . \quad (\mathbf{7})$$

The force for the "gel effect" is then defined by Equation 8. Zero z coordinate restricts the F_G to the strictly horizontal direction.

$$F_G = [v_x, v_y, 0] \cdot F_M \quad (\mathbf{8})$$

The final force feedback F_{out}, which is sent to the device, is interpolated between two consecutive frames in order to smooth any abrupt changes during haptic frames between the geometry updates. In the haptic thread, there are two counters: the total frame count in the previous haptic rendering sequence it_p and the frame count rendered since the last geometry update it. The resulting force is computed from Equation 9:

$$F_{out} = F_{old} + \frac{F - F_{old}}{it_p} \cdot it , \quad (\mathbf{9})$$

where F is the force from Equation 2 and F_{old} is the force F computed before the last geometry update.

EXPERIMENTS AND RESULTS

We have implemented our algorithm in C++ and used OpenGL library, GLSL, and OpenHaptics® library [Sensable] for haptics rendering. All data were measured on a standard desktop computer running Windows 7, with 12GB RAM, Intel Core i7 clocked at 1.67GHz, and NVIDIA GeForce GTX 295, with 1.5GB of memory.

We have experimentally verified our algorithm by drawing into real box with two types of real sand and comparing visually the results as shown in Figure 3a. So far, the haptic feedback has been tested in the scope of our working group.

As a demonstration of haptic manipulation we have simulated the same effect using our system (see Figure 3b). Force feedback is smooth and it is almost the same as the real sand. The penetration force is slightly increased with increasing depth of the tool. The user must make a significant effort to push the tool to the sides if its depth is high. If the tool rests, the force components are mutually balanced.

(a) (b)

Fig. 3. The word GIS written in real sand (a); simulated using our system (b).

CONCLUSIONS

The force feedback in our model is not physically accurate because we have a haptic device with only a point cursor, but the touch sense feels very much like a real sand as we verified in the experiment. As haptic device is nowadays rather cheap and available, the described approach can be useful for the reader.

There are many possible ways how the algorithm can be extended. One apparent way is by using a different type of triangulation.

Another possible avenue for future work is to perform a user study to measure the accuracy of force feedback or to extend the haptic visualization with materials with different granularity.

ACKNOWLEDGEMENTS

We would like to thank to Petr Vaněček (University of West Bohemia, Plzen) for his valuable comments and Jan Flasar (Masaryk University, Brno) for his help with introduction to haptic.

This work has been supported by the Czech Science Foundation under the research project project 201/09/0097, and project SGS-2010-028 and by NSF IIS-0964302, NSF OCI-0753116 Integrating Behavioral, Geometrical and Graphical Modeling to Simulate and Visualize Urban Areas.

REFERENCES

Benes, B., Dorjgotov, E., Arns, L., Bertoline, G. (2006) Granular Material Interactive Manipulation: Touching Sand with Haptic Feedback. Proceedings of the 14-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, pp. 295-304.

Berg, de M., Kreveld, van M., Overmans, M., Schwarzkopf, O. (1997) Computational Geometry: Algorithms and Applications. Springer-Verlag.

Faeth, A., Oren, M. and Harding C. (2008) Combining 3-D Geovisualization with Force Feedback Driven User Interaction. In Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems (GIS '08). ACM, New York, USA, Article 25, 9 pages.

Guo, X., Hua J. and Qin, H. (2004) Touch-Based Haptics for Interactive Editing on Point Set Surfaces. IEEE Computer Graphics Applications, pp. 31-39.

Li, Y. and Brodlie, K. (2003) Soft Object Modeling With Generalized Chainmail, Computer Graphics Forum, 22(4): pp. 717–7.

Lin, M. C., Otaduy, M. (2008) Haptic Rendering: Foundations, Algorithms and Applications. A. K. Peters, Ltd., Natick, MA, USA.

Purchart, V. (2009) Interactive Manipulation of Sand on a TIN Terrain Model for Virtual Reality. In 12-th Central European Seminar on Computer Graphics (CESCG 2009).

Purchart, V., Kolingerová, I., Benes, B. (submitted) Interactive Manipulation of an Adaptive Triangulation-based Sand-covered Terrain. The Visual Computer.

Sensable. URL: http://www.sensable.com/products-openhaptics-toolkit.htm, 8th December 2011.

Sloan, S. W. (1993) Fast Algorithm for Generating Constrained Delaunay Triangulations. Computers and Structures. Vol. 47, no. 3, pp. 441-450.