The Force-Map Haptic Rendering Algorithm for Drilling into Volume

Y.Liu
School of Computing Sciences,
University of East Anglia,
NR4 7TJ, Norwich, UK
Yu.liu@uea.ac.uk

S.D.Laycock
School of Computing Sciences,
University of East Anglia,
NR4 7TJ, Norwich, UK
S.Laycock@uea.ac.uk

ABSTRACT

With the developments of volume visualization technology for complex data sets comes new challenges in terms of user interaction and information extraction. Volume haptics has proven itself to be an effective way of extracting valuable information by providing an extra sense from which to perceive three dimensional data. The work presented in this paper introduces a Force-Map method that combines the benefits of the indirect and the direct haptic rendering approaches. Users can select from a variety of virtual tools to gain continuous and smooth force feedback during the drilling of volumetric data which increases the applicability of the approach.

Keywords

Volume haptics, Direct volume haptics, Marching cubes, Force-Map haptic rendering.

1. INTRODUCTION

Volume visualization has become widely utilized for many applications. The ability to visualize volume data directly is particularly important for medical applications where a correct anatomical view of the patient can prove vital for surgical planning. Recent developments in graphics accelerator cards have enabled systems to render large and complex volumetric data sets in a variety of different rendering styles aiding the observer's perception of the data. As early as 1993, these visualizations were linked with haptic feedback devices to enable the user to touch the volumetric data. Iwata and Noma used their approach for the haptic interaction of data produced in Computational Fluid Dynamics. In this case a force could be mapped to the velocity and torque mapped to the vorticity [Iwa93]. Virtual Sculpting systems linked to haptic feedback devices have been available for many years; however, these often do not ensure the modified data remains faithful to the characteristics of the original volumetric data. In this paper, a novel approach to drill into surfaces based on the volumetric data is presented.

One of the major considerations of any application incorporating haptic feedback is the rate at which the calculations must be performed. Based on the results of analyzing human factors, an update rate of 1KHz is required in order for a user to perceive stable and smooth haptic feedback from the system. This is in contrast to the visualization which must update at approximately 30Hz. If the haptic update frequency is lower than 1KHz, an obvious vibration can be felt from the haptic device. One objective of this work is to create a system which can accurately render volume data at sufficient rates for both the visualization and the haptics. To achieve this goal, there are a number of key points that must be addressed. Firstly, the algorithm used to update the volume data must be fast, especially considering the fact that the surface representation of the volume data may be constructed from millions of triangles.

Secondly, the haptic feedback should be rendered such that when the probe point is moving across the voxel boundaries a continuous force is returned to the user. Lastly, since the haptics and visualization calculations will be performed in separate threads, mechanisms are required to ensure that each thread can be updated in a safe manner.

2. PREVIOUS WORK

Volume haptic rendering techniques can be categorized mainly into two classes. The first is indirect haptic rendering, which typically extracts an intermediate geometric representation or iso-surface from the volumetric dataset. The second class is direct volume haptic rendering, which computes the force feedback according to the information stored in the voxels. In this work, the Marching Cubes algorithm [Lor87] is used for extracting a surface from volume data. Once a surface pattern has been identified, for each voxel, a complete triangulation of the volume data will be obtained. There have been several attempts to utilize a localized Marching Cubes algorithm for haptic applications [Kob99]. However, when used in conjunction with haptics force discontinuities often result.

The first category of haptic rendering approaches utilizes a surface-based haptic rendering technique for the interaction with the volume data. In this case an intermediate geometric representation of the volumetric datasets, such as an iso-surface, is constructed. The geometry of the surface can then be easily haptically rendered by utilizing a standard constraint-based method [Zil95]. However, this suffers from stability problems which occur when the surface is updated. The direct volume haptic rendering approach is capable of providing a way to generate force feedback directly from the volume data without extracting an intermediate representation. Even though it is able to represent the force at any position in the volume data, the haptic feedback generated by this method suffers from force instabilities since it is difficult to properly decide the rendering parameters in the force function. This is especially the case when the function is changing during the process, such as when drilling or milling, in real applications. Moreover, forces may vary significantly in strength and direction which sometimes can not be represented by a simple mapping method.

Many previous efforts have been done on these issues which limit the usability of direct volume haptics. The Voxmap Point-Shell algorithm was introduced by McNeely et al. [Mcn99] in which the three dimensional volume data is represented using a Voxmap. The virtual three-dimensional object is encoded into the Voxmap using voxels that identify which locations in space are occupied. The haptic device is modelled by a number of points that sit around the surface. A force response is then calculated for the haptic device based on the voxel-point intersections. However, the force calculation method suffers force discontinuities at voxel boundaries. Pflesser and Petersik et al. [Pet02] also proposed a haptic system for virtual temporal bone surgery which uses a modified version of the Voxmap-Pointshell algorithm [Mcn99]. Their approaches sample the surface of the drilling instrument and then generate appropriate forces at each sampled point. A number of samples are distributed around the drill and a ray-tracing approach is then employed to calculate the force vectors towards the tool centre, which can subsequently be combined to generate the overall force returned to the haptic feedback device. The ray-tracing algorithm has the potential to miss voxel data located between two rays due to an insufficient sampling.

Morris et al. [Mor04] simplifies the computations for drilling through the use of another point-shell method to compute haptic interactions and bone erosion for spherical drill bits. In contrast to the work of Pflesser et al. [Pet02], Morris et al. use the data within the spherical tool to perform bone removal as opposed to sampling points on the tool’s surface. Both of these approaches limit the user to drilling with a spherical drill. Eriksson et al. [Eri05] proposed a haptic milling surgery simulator using a localized Marching Cubes algorithm for the visualization. To improve the stability they employed a direct haptic rendering method with mechanisms to remove fall-through issues. The data inside the virtual drill is set to a vector pointing to the centre of the voxel. The output force is the sum of all those vectors. This approach works well when the drilling tool moves in a small area, but a “kicking” would result when the haptic test points move across the cubes’ boundaries.

To overcome the stability issues discussed in the previous papers, the work presented in this paper utilizes a fast local Marching Cubes algorithm and a Force-Map haptic rendering procedure to gain smooth and stable force feedback during drilling with more varied drilling tools.

3. SURFACE EXTRACTION AND MODIFICATION

The volume-based representation is a natural choice for rendering a collection of digital images produced by medical scanning technologies such as Magnetic Resonance Imaging (MRI) or Computed

Tomography (CT). There are a variety of graphical rendering techniques for visualizing the three dimensional data, often with options to display the material properties such as density and viscosity within the voxels. This has the potential to greatly enhance a user's performance in medical and scientific three dimensional data exploration.

When using the Marching Cubes algorithm [Lor87], a volume can be interpreted by generating polygons representing the surface, typically constrained to a specified value of the data. To handle large data sets, an Octree-based structure is employed which enables the data to be changed dynamically in an efficient manner.

The efficiency of the volume data modification step is dependent on the volume of the tool. As the user manipulates the haptic device the tool moves accordingly through the three dimensional data. A bounding box around the tool is constructed, which is aligned to the coordinate system of the volume data. The data points within the bounding box are tested to determine if they are inside the tool's volume. If the points are interior to the volume then their density value will be reduced based on the distance to the centre axis of the cylinder or a point in the case of the sphere. This gradual data value adjustment permits the smooth removal of volume data. Once adjusted, the Bounding box volume will be updated using the local Marching Cubes algorithm to generate a new surface.

4. HAPTIC RENDERING

The approach described by Eriksson et al. [Eri05] suffers force discontinuities when the tool moves between the encoded cubes. Sample points in this work are tested for contact with the volume data.

Given a sample point position, a vector calculated from the occupancy force-map can be output. By using this method, the force feedback is stable and smooth even though it has a similar force cube encoding system. The force vectors stored in the data are calculated based on the local surface, which also benefits from the advantages of the surface-based haptic rendering approach. The synchronisation of updating the graphic and haptic loops enhances the fidelity of the virtual visual-haptic system when applied to real applications. The three steps below outline the Force-Map haptic rendering method adopted for a surface representation of dynamically changing voxel data.

Initialization and construction

Initially all the normals of the triangles contained in each octree leaf node (voxel) are averaged to result in a single force vector representing the data in the voxel. The larger the voxel is, the more volume data points lie within it. Additionally, only the data inside the voxel is assigned to a force vector while others are set to none. After this initialisation step, all the data near to the surface is set to a force vector which approximately equals the closest surface normal.

Update and reconstruction

When the surface is updated in the haptics thread the data points that are found to lie inside the new voxel are set to a force value based on the triangle's face normal. If there is more than one triangle in the voxel, the averaged face normal will be used. Some force values in the old surface might also need to be updated since the triangles forming the surface in the voxel may have changed.

Calculating the force feedback

The force vectors stored in the data must be combined appropriately before being returned to the haptic device. When the virtual drilling tool moves into the volume data, a haptic test point checks the surrounding eight data values in the three dimensional space. These eight data values are referred to as the force cube in this work. The corners of the force cube contain the force vectors stored in the data. Tri-linear interpolation is employed here to enable an interpolated force vector to be calculated for any position inside the force cube. Another advantage of using the tri-linear interpolation method is that the haptic test point can be smoothly moved from one force cube to another without any force discontinuities occurring between them.

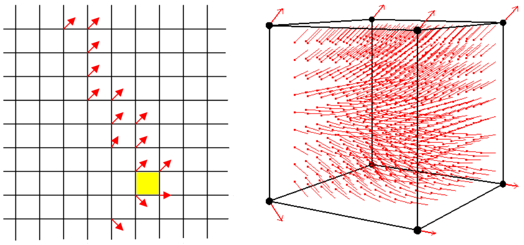

Figure 1. Force-Map haptic rendering, the red arrows represent the force vector. (a) The yellow square indicates one force cube represented in two dimensions. (b) The same single force cube in three dimensions.

For any real application, drilling with a single point does not lead to a realistic result. An approach involving multiple test points approximating the drilling tool is usually preferred. In this work, a number of hapic points are distributed approximately around the surface of the drilling tool. At each time step, each haptic point is tested in the constructed Force-Map to calculate the contribution to the overall haptic force. In many applications, the properties of the simulated materials differ depending on the location being drilled. This is particularly the case in medical and dental applications where the material properties of each voxel must be considered. For example drilling through soft tissue should be very different to drilling through rigid bone. The Force-Map method can easily incorporate this issue by simply setting a scaled force vector where the scaling factor is related to the neighbouring voxel data.

5. RESULTS

Figure 2 illustrates a procedurally generated sphere along side a surface representation of a human pelvis. The surface was extracted from 87 CT slices obtained at the Norfolk and Norwich University Hospital, UK.

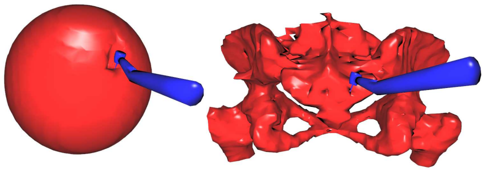

Figure 2. The left sphere-like object is created procedurally whilst the right hand image was extracted from 87 CT image slices. Each slice contains 256 x 256 pixels.

The volume on the right of Figure 2 has been calculated from a data set comprising of 256 x 256 x 87 data points. For most drill tools tested, the haptic rendering loop is updated at a rate which exceeds 1000Hz. Figure 3 shows the time required to perform the surface modification and Force-Map updates during rendering. Table 1 illustrates the average update times during drilling for the selection of tools. The sphere drilling tool is the most efficient since its volume occupies on average the fewest voxels.

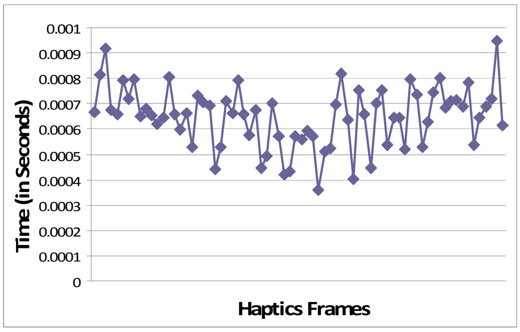

Figure 3. A graph presenting the time taken to update the surface during drilling with a sphere

Sphere	Cylinder	Combination	Dental Drill
1550Hz	877Hz	770Hz	1378Hz

Table 1. A table showing the average haptic rates whilst drilling with a selection of tools.

The work has been tested on a Quad Core 2.4GHz processor PC with a NVIDIA Geforce 8800GTX graphics card. To provide haptic feedback a PHANToM Omni device has been employed.

6. CONCLUSIONS

In this paper a Force-Map haptic rendering algorithm is proposed to achieve real-time drilling of volumetric objects. In order to gain more realistic force feedback for drilling applications, arbitrary tool model selection has been implemented in this work, for tools based on implicit equations. The future work will implement the tangential force which is an important property of the drilling application by using the Force-Map haptic rendering algorithm.

7. REFERENCES

[Eri05] Eriksson, M., Flemmer, H., and Wikander, J. A Haptic and Virtual Reality Skull Bone Surgery Simulator. Proc. WorldHaptics, Italy, 2005.

[Iwa93] Iwata H., Noma H. Volume haptization. In Proc. of IEEE Symp. On Research Frontiers in Virtual Reality, pp. 16–23, 1993.

[Kob99] Kobbelt L, Botsch M, Schwanecke U, Seidel P. Haptic volume rendering with an intermediate local representation. pp. 79–84.

[Lor87] Lorenson W.E., Cline H.E. Marching cubes:A high resolution 3d surface construction algorithm. SIGGRAPH. 21, pp. 163–169, 1987.

[Mor04] Morris D., Sewell C., Blevins N., Barbagli F., and Salisbury K. A collaborative virtual environment for the simulation of temporal bone surgery. In MICCAI, 2004.

[Mcn99] McNeely WA., Puterbaugh KD., and Troy JJ. Six degree-of-freedom haptic rendering using voxel sampling. Proceedings of ACM SIGGRAPH, pp. 401-408, 1999.

[Pet02] Petersik A., Pflesser B., Tiede U., Hohne K. H., and Leuwer R. Haptic volume interaction with anatomic models at sub-voxel resolution. In IEEE Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 66–72, 2002.

[Zil95] Zilles C., Salisbury J. A constraint based god-object method for haptic display. In Proc. of the IEEE Conference on Intelligent Robots and Systems, pp. 146–151, 1995.