EG UK Theory and Practice of Computer Graphics (2008)
Ik Soo Lim, Wen Tang (Editors)

A Haptic Rendering Algorithm for Drilling into Volume Data

Y. Liu S. D. Laycock

School of Computing Sciences, University of East Anglia
Norwich, NR4 7TJ, UK

Abstract

With the developments of volume visualisation 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. In this paper, a novel indirect haptic rendering method using a Marching Cubes algorithm is presented for volume data removal. A novel three-step haptic rendering method is presented, which can be used to provide continuous and smooth force feedback during the drilling of various types of volumetric data.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Geometric algorithms

1. Introduction

Volume visualisation has become widely utilised for applications such as Virtual Training [HBS97] and Scientific Visualisation [IN93]. The ability to visualise volume data directly is particularly important for medical applications where a correct anatomical view of the patient can prove vital for surgical planning. In this paper, a novel approach to drilling 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. 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 visualisation which must update at approximately 30Hz. One objective of this work is to create a system which can accurately render volume data at sufficient rates for both the visualisation and the haptics.

There are two threads accessing and modifying the same set of volume data. To ensure a thread-safe implementation a novel three-step haptic rendering algorithm is employed to avoid accessing the same area of memory at the same time.

2. Related work

Volume haptic rendering techniques can be categorised mainly into two classes. The first is direct volume haptic rendering, which computes the force feedback according to the information stored in the voxels. The second class is surface-based or indirect haptic rendering, which extracts an intermediate geometric representation of the volumetric datasets such as an iso-surface. The Marching Cubes algorithm is commonly used for extracting a surface from volume data. This method first partitions the volume data into voxels. The corners of each voxel are labelled exterior or interior depending on the eight neighbouring volumetric data points. The surface configuration of each cube is then described by 15 basic surface patterns [LC87]. 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 utilise a localised Marching Cube algorithm for general computer graphics applications. Shu et al. [SZK95] proposed an adaptive algorithm in 1995, which first applies the Marching Cubes algorithm to a sub-area of the volume. This sub-area was defined by a small fixed number of voxels in a grid. Later, Korner et. al. [KBSS99] presented an approach where volumetric data is locally transformed into a surface representation for haptics. For the explicit local surface extraction, the Marching Cubes algorithm is employed on a 7x7x7 voxel environment surrounding the stylus position. However, this method suffers from strong discontinuities especially during the fast movement of the stylus. More recently, Eriksson et. al. [MJ06] proposed a haptic milling surgery simulator using a localised Marching Cubes algorithm for the visualisation. Their work allows the voxels to be updated in real-time for a restricted area of the volume data and they employ high resolution volumetric data (512x512x174) to construct approximately 2.4 million triangles for the visualisation. To overcome the stability issues discussed in the previous papers, the work presented in this paper utilises an indirect haptic rendering approach, employing a fast local marching cubes algorithm and haptic rendering procedure to smooth the transitions between new and old surfaces during drilling. To reduce triangle edge aliasing a mean filtering technique is applied to the extracted surface [KBSS01].

Most of the previous work, particularly in the medical field, employs relatively simple tools such as single points or spheres. These simple tools are not sufficiently realistic to replicate the interactions that are present in many applications. The proposed method strives for a general implementation by enabling tools of arbitrary shape to be employed. The next section details how the surface can be extracted from the volume data and how this surface can be modified efficiently in real-time.

3. Surface extraction and modification of the Volume data

A surface is extracted using the Marching Cubes algorithm. Many applications, especially medical training simulators, require more information, such as the internal structure and material properties of the virtual object. Therefore in these applications a surface representation alone does not provide sufficient information. A volume-based representation can be used to meet these requirements by rendering a volumetric object from a regular or irregular three dimensional array of data. This data may be derived from a collection of digital images such as from Magnetic Resonance Imaging (MRI) or Computed Tomography (CT).

3.1. Volume data modification

Extracting the global isosurfaces from the volume data can be time consuming especially when the volume data is derived from many high resolution digital images. However, in this work a local Marching Cubes algorithm is employed when the data is modified. The values of the volume data surrounding the haptic stylus can be adjusted depending on the application. The simplistic approach can just involve setting the values of the volume data surrounding the stylus to less than a surface threshold value. However, by considering the material properties of the data contained within a voxel the rate at which the data is removed can be adjusted. Once the data has been updated the local Marching Cubes approach recomputes the surface surrounding the stylus. The volume that is updated depends on the resolution of the volume data and the shape of the tool used for the interaction.

3.2. Tool modelling and tool-object interaction

The efficiency of the volume data modification step is, in part, dependent on the volume of the interaction tool. As the user manipulates the haptic device the interaction 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 and their density value is less than the user specified iso-value threshold then the points are labelled to ensure that they will be updated during the local Marching Cubes algorithm. This approach is fully extendable to different shaped tools by representing them using three dimensional implicit functions.

4. Haptic rendering

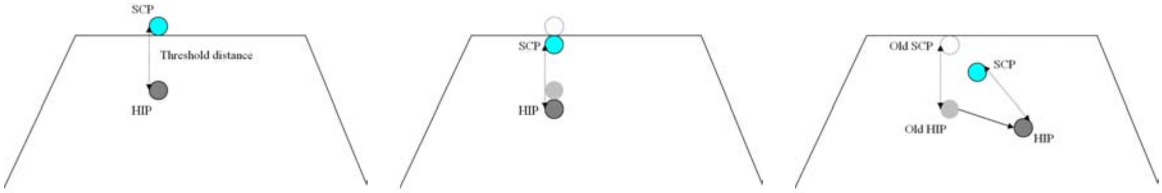
The Direct Volume Haptic rendering approach defines a direct mapping from voxel properties to force feedback. In contrast the indirect approach utilises an intermediate geometric representation of the volumetric datasets, such as an isosurface [KBSS99]. The geometry of the surface can then be easily haptically rendered by utilising a standard constraint-based method [RKK97] [ZS95] [HBS97].

In this work Ho and Basdogan's [HBS97] constraint-based technique for haptic rendering is employed to enable force feedback to be determined from the polygonal surface. This rendering algorithm can be adapted for the data structures employed for the voxel grid and extracted polygonal surface. The basic idea of this method is that when the haptic probe or Haptic Interface Point (HIP) moves beneath the surface of the object, a corresponding proxy point remains on the object's surface. The proxy position tracks the HIP and the resulting force vector is calculated using the difference vector between the proxy and the HIP's position. The four steps below outline the constraint-based haptic rendering method adapted for a surface representation of dynamically changing voxel data.

1. Detecting collisions.

The closest points of contact between the interaction tool and the extracted surface must be efficiently computed. Often a single point is sufficient to act as the main contact point or haptic interface point, HIP. If the HIP has moved beneath a triangle and therefore, now lies inside the object, a corresponding Surface Contact Point (SCP) on the closest triangle must be obtained.

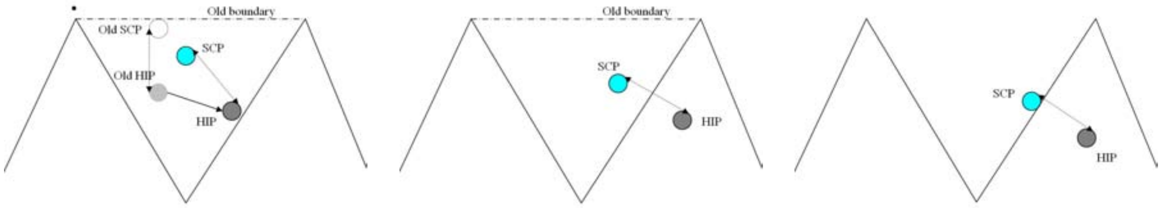
2. Determine the SCP position on the closest triangle.



(a) At the point of initial contact the SCP is constrained to the surface.

(b) Once the specified threshold is exceeded the volume data may be modified in a separate thread.

(c) In subsequent frames of the rendering method the SCP is dragged behind the HIP.



(d) A new boundary will be constructed based on the modified volume data.

(e) The procedure continues to move the SCP behind the HIP until it collides with a new triangle.

(f) The SCP and HIP return to a standard haptic rendering approach, enabling drilling to continue.

Figure 1: *Haptic Rendering Procedure for Drilling, illustrated on a two dimensional slice of a polygonal object.*

When a collision is detected, the closest triangle to the HIP can be determined. If the closest point on the plane lies exterior to the triangle boundary it is clamped to the closest edge or vertex using an approach involving Barycentric Coordinates. The SCP will be initialized to this closest point.

3. Tracking the SCP on the surface.

The Marching Cubes approach constructs a surface composed of triangles from the volume data. In each successive iteration of the haptic rendering procedure the SCP must be tracked over this surface whilst the HIP remains interior to the surface. The tracking of the SCP follows from Ho and Basdogan's approach [HBS97]. In this work, once the SCP is known to have left the previous triangle, all the surrounding voxels are searched to obtain the triangle located in either the current voxel or one of the adjacent voxels, which is closest to the new SCP.

4. Computing force feedback

After the new SCP position has been determined the force feedback is calculated in the usual way using a virtual spring-damper system, coupling the HIP to the SCP.

For high-quality, stable haptic rendering, the above mentioned process must run at a rate of 1000Hz. In order to visualise the surface the graphics are computed in a separate thread which must update at a rate of at least 30Hz. In this work a new thread safe update method is proposed to tackle the problem of the two threads accessing the same data at the same time.

4.1. Haptic Rendering during Data Modification

In this approach the data modification and surface extraction is performed in the graphics thread to avoid causing instabilities in the haptics thread. To ensure smooth haptic rendering, a mechanism is required to allow the SCP to track from the old surface to the newly created one. It also must ensure a stable force is perceived whilst the new surface is extracted from the modified data. This is achieved using the following three steps.

Step 1. It is assumed that the data will only be modified if the user pushes the HIP into the surface of the object by a specified distance. To achieve this effect the vector, Distance, between the SCP and the HIP is determined. If the magnitude of this vector is larger than a preset threshold value then the process may proceed to Step 2 (shown in Figure 1(A) and (B)), where data modification takes place. Otherwise the rendering process stays at Step 1 where the returned force is calculated using the vector between the HIP and the SCP method, as previously outlined. The preset threshold may be adjusted to vary the amount by which the HIP must penetrate the surface before drilling commences.

Step 2. In this step the data is modified in the graphics thread based on the HIP position. Since the data is being modified the SCP can not be tracked in the traditional way. At this point the SCP will effectively be dragged behind the HIP in the direction of the HIP's movement vector (shown in Figure 1(C)). During this stage the distance between the

SCP and the HIP will remain constant at a value specified by the threshold used in Step 1.

Once the data modification and surface regeneration is complete, new triangles are available (shown in Figure 1(D)). However, the HIP may not have moved into one of these new triangles (shown in Figure 1(E)). Therefore, the drag process continues until the SCP touches one of the new triangles. At this point the algorithm proceeds to Step 3.

Step 3. When the process goes into Step 3 (Figure 1(F)), it will use the traditional method to calculate the SCP on the newly obtained triangle. The scalar value of the force feedback will be the same as the threshold value, which enables the user to feel smooth force feedback.

5. Results

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, produced by SensAble Technologies has been employed. By using the system, a user can drill into rigid objects using different types of tools with the three dimensional user interface. The surface shown in the left of Figure 2 has been calculated from a data set comprising of 512 x 512 x 512 data points. The surface on the right has been extracted from 87 CT slices of the human pelvis obtained at the Norfolk and Norwich Hospital. This data has been sampled to create a number of triangles. The haptic rendering loop is updated at a rate which exceeds 1000Hz. While touching a 3D object, without drilling, a frame rate of 800Hz is achieved. Since the data modification process is computed in the graphics thread the frame rate reduces to approximately 300Hz during the drilling approach. On average the local Marching Cubes approach requires 0.4ms to update the volume data. This allows users to efficiently obtain visual and haptic cues.

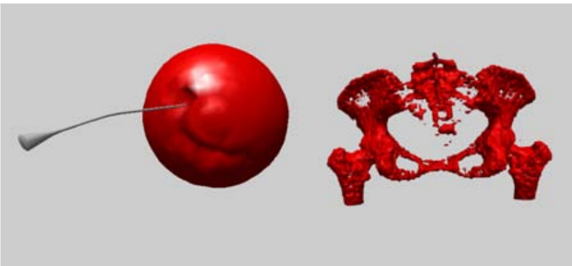


Figure 2: *The left sphere-like object is created procedurally from the data whilst the right hand image was extracted from 87 CT image slices. Each slice contains 256 x 256 pixels.*

6. Conclusions and Future Work

In this paper an algorithm based on Local Marching Cubes is proposed, to achieve real-time sculpting of volumetric objects. The proposed method enables volumetric material to be removed in a stable haptic rendering system. These objects can be derived from procedurally generated volume data or image slices, such as CT or MRI. In the future the physical properties of the material will be considered to facilitate the drilling of deformable objects.

A novel procedure is presented to enable a smooth response to be perceived during the drilling stage. Presently this approach is based upon a modification to a single point haptic rendering algorithm. However, this approach can easily be extended to three dimensional tools. Future work will focus on multi-point haptic rendering algorithms, which could enhance the fidelity of the application.

References

[HBS97] Ho C., Basdogan C., Srinivasan M.: Haptic rendering: Point - and ray - based interactions. In *Proc. of the Second PHANToM Users Group Workshop* (October 1997).

[IN93] Iwata H., Noma H.: Volume haptization. In *Proc. of IEEE Symp. on Research Frontiers in Virtual Reality* (October 1993), pp. 16–23.

[KBSS99] Kobbelt L. P., Botsch M., Schwanecke U., Seidel H.-P.: Haptic volume rendering with an intermediate local representation. pp. 79–84.

[KBSS01] Kobbelt L. P., Botsch M., Schwanecke U., Seidel H.-P.: Feature-sensitive surface extraction from volume data. In *SIGGRAPH 2001, Computer Graphics Proceedings* (2001), Fiume E., (Ed.), ACM Press / ACM SIGGRAPH, pp. 57–66.

[LC87] Lorensen W. E., Cline H. E.: Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Computer. Graphics Proceedings. 21*, 4 (1987), 163–169.

[MJ06] M E., J. D. M. W.: A haptic vr milling surgery simulator using high resolution ct data. *Proceedings of Medicine Meets Virtual Reality. 119* (2006), 138–143.

[RKK97] Ruspini D., Kolarov K., Khatib O.: The haptic display of complex graphical environments. In *Proc. of ACM Siggraph 97* (Los Angeles, CA, 1997), pp. 345–352.

[SZK95] Shu R., Zhou C., Kankanhalli M. S.: Adaptive marching cubes. *The Visual Computer 11*, 4 (1995), 202–217.

[ZS95] Zilles C., Salisbury J.: A constraint based god-object method for haptic display. In *Proc. of the IEEE Conference on Intelligent Robots and Systems* (1995), pp. 146–151.