Direct Haptic Rendering of Isosurface by Intermediate Representation

Kwong-Wai Chen
The Chinese University of Hong Kong
Shatin, Hong Kong
kwchen@cse.cuhk.edu.hk

Pheng-Ann Heng
The Chinese University of Hong Kong
Shatin, Hong Kong
pheng@cse.cuhk.edu.hk

Hanqiu Sun
The Chinese University of Hong Kong
Shatin, Hong Kong
hanqiu@cse.cuhk.edu.hk

ABSTRACT

With the development of volume visualization methods, we can easily extract meaningful information from volumetric data using interactive graphics and imaging. Haptic interaction of volumetric data adds a new modality to volume visualization that has an advantage in presenting complex attributes of local region. However, the benefits of haptic rendering of volumetric data have only been recognized recently. Most traditional haptic rendering methods are developed to compute realistic interaction force with geometric primitives. Direct volume haptic rendering allows haptic palpation of volumetric data, but lacks of the ability of simulating the contact sensation of stiff embedded implicit surface.

In this paper, we propose a direct haptic rendering method for isosurface in volumetric data using a point-based haptic feedback device, without the extraction of the isosurface to geometric representations such as polygons. Our algorithm uses a virtual plane as an intermediate representation of the isosurface, and computes the point interaction force applied to the haptic interface based on this virtual plane. Using this approach, we are able to gain higher haptic servo rate for volumetric data. It makes maintenance of the stability of the simulation easier, and applicable to noisy data without preprocessing. We have developed our algorithm and tested with synthetic data and medical data, using the PHANToM™ haptic interface.

Keywords

Virtual Reality, Haptic Rendering, Volume Visualization

1. INTRODUCTION

Volumetric data are 3D entities that may have information inside them, might not consist of surfaces and edges, or might be too voluminous to be represented geometrically. Volume data are obtained by sampling, simulation, or modeling techniques in many area. For example, a sequence of 2D slices obtained from Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) is 3D reconstructed into a volumetric model and visualized for diagnostic purposes or for planning of treatment or surgery. While volume visualization methods of extracting meaningful information from volumetric data using interactive graphics and imaging have been proven quite effective for most applications, it remains worthwhile to investigate the benefit of augmenting these visualization methods with information obtained through other sensory channels. In particular, our sense of touch, in combination with our kinesthetic sense, is capable of supplying a large amount of information about the structure, location, and material properties of objects. The study of the many issues related to interaction with an environment through the sense of touch is known as *computer haptics*.

Haptic rendering refers to the computational methods used to determine the forces that resulted when we interact with virtual objects. This involves tactile feedback for sensing properties such as surface texture, and kinesthetic feedback for sensing the shape and size of objects. Traditional methods for producing convincing haptic renderings have mainly utilized scenes comprised of geometric primitives such as polygons, spheres, and surface patches. These approaches have generally focused on simulating realistic interactions with static and dynamic collections of geometric objects given the capabilities and limitations of haptic devices. The haptic rendering methods for geometric models can be classified into two groups: *penalty based methods* and *constraint based methods*. In penalty based methods, also known as vector field methods [10, 9], force proportional to the amount of penetration into a virtual volume are applied to the haptic device. However, this approach have a number of drawbacks. First of all, it breaks down for objects with complex polygonal meshes. Moreover, it also has problem when dealing with multiple objects and object with thin volume. Finally, force direction and magnitude computed may not be continuous across a sub-volume boundary. The drawbacks of penalty based methods led to solutions that keep a history of contact surfaces description of the objects, so that we always know the surfaces the haptic interface point has passed through. The algorithms are grouped together under the name of constraint based methods [12, 11], they are well-designed for generating convincing interaction forces for objects modeled as rigid polyhedral. In this approach, a virtual position is defined to indicate where the haptic interface point (HIP) will be located if the haptic interface could not penetrate the surface. The virtual haptic interface point (VHIP) is constrained by the object surface, which models real world object interaction of rigid bodies. The force computed is proportional to the displacement between the haptic interface point and the surface contact point. Moreover, a rich set of surface properties can be easily simulated by restricting or changing the motion of the VHIP.

Haptic rendering of the geometric models by constraint based methods allow a rich set of touch sensation of the virtual object. However, volumetric data are not comprised of geometric primitives, the traditional methods developed are not directly applicable without appropriate conversion of volumetric data into geometrical descriptions. Recently, the benefits of haptic rendering of volumetric data have been recognized, but this area of research has not yet been fully explored. Haptic interaction of volumetric data adds a new modality to volume visualization [4, 5, 8]. Visual information has an advantage in presenting whole image of the object, on the other hand, haptic information has an advantage in presenting complex attributes of local region.

There are two types of haptic interaction with volumetric data, which depends on the purpose of the haptic simulation. The force generated can either be constructed to approximate a realistic feeling of a virtual object or to convey meaningful full structural information for data exploration. The first one is to simulate the contact force of the isosurfaces contained in the volumetric data. Isosurface extraction is the technique used in volume visualization to explicitly represent the isosurface structure by a geometric model. The geometric model can be haptically rendered by the geometric haptic rendering methods. The disadvantage of this approach is that the geometric model extracted by isosurface extraction methods generally contains a large number of polygons which challenge the performance of the haptic rendering. Moreover, it requires a preprocessing of the volumetric data, thus the simulations that dynamically change the structure of the volumetric data are impossible. This motivates researchers to develop algorithms that direct haptically render the stiff isosurfaces in the volumetric data. The second is analogous to the direct volume rendering of volume visualization, where the reaction force is directly computed by using the information stored in the voxels.

In the remainder of this paper, **Section 2** describes the approaches of haptic rendering of volumetric data as a survey. In **Section 3**, we propose a method for direct haptic rendering of the isosurface in volumetric data by an intermediate representation of virtual plane. Then, we present the implementation of the algorithm in **Section 4**. **Section 5** is the discussion and conclusion.

2. HAPTIC RENDERING OF VOLUMETRIC DATA

2.1 Volume Haptization

Just like direct volume rendering methods used in volume visualization, direct haptic rendering of the volumetric data has the advantage that it convey more information of the volumetric data which maybe useful for data exploration. For example, the user may wish to explore internal structures of an organ, such as lung. The method of direct haptic rendering is known as volume haptization [2, 7], where its basic idea is to define a direct mapping of voxel value to force and/or torque. The mapping are based on two principal requirements. First, the interaction forces must be calculated fast enough to be used within an interactive system. Second, in order to have a better consistence of visual and haptic feedback, the forces imparted to the user should have a direct relation with the visual appearance of the volumetric object. Therefore the force transfer function is generally defined corresponding to the transfer function of the opacity defined for visual rendering. Moreover if we employ a segmentation step to determine the visual appearance of volumetric data, we should also introduce a similar step to haptic rendering.

2.2 Isosurface Haptic Rendering

A volumetric data may contain an implicit surface inside it, which is defined by an isovalue (for example, the surface of an organ in a CT-scanned medical data). In virtual surgical training and planning system, it is useful that the contact sensation of the isosurface can be haptically simulated. As motioned before, it is undesirable to apply haptic rendering to the geometric approximation generated by isosurface extraction algorithms. This motivates researchers to develop algorithms which directly haptically render the isosurface within the volumetric data [2, 3].

In [2], Avila and Sobierajski try to calculate the stiffness and motion retarding forces when interacting with volumetric isosurfaces, by relating the force transfer functions with the sample density. The stiffness computation requires that the penetration distance of the HIP below the isosurface is available at every location in the volumetric data. While it is possible to precompute the distance to an isosurface for every sample of the data, the technique used is to approximate the stiffness and retarding forces based only on the density field. There are two reasons for this. First, if the simulation allows interactive modification of the volumetric data, creating a new distance map for the data would be prohibitive. Second, for small penetration distances, the density field itself can give a reasonable approximation of the distance to an isosurface.

However, the distance of a point from the isosurface may not have a direct relation with the density difference. Moreover, contact sensation of some object such as bone cannot be realistic simulated by a penetrable shell model. Blezek and Robb [3], try to simulate the feeling of stiff structure by using a method which is similar to the constraint based haptic rendering methods for geometric model.

Just as constraint based rendering methods for geometric model, to haptically render stiff objects, the virtual point of contact with the surface (surface contact point or virutual haptic interface point, VHIP) must be maintained. Since isosurface is implicitly contained in volumetric data, a different approach is proposed to move the VHIP along the isosurface. The VHIP is interactively moved to a new location on the surface, which follows in three dimensions the contours of the surface. VHIP must be chosen such that the surface normal at VHIP must be equal to the vector from

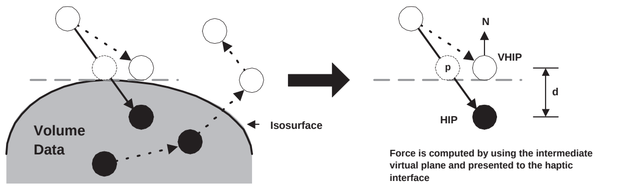

Figure 1: Direct haptic rendering of Isosurface through an intermediate representation

the HIP to VHIP in order to approximate the motion of real objects moving across one another. In the general case, the point at where the surface normal and the vector are equal may have several solutions at disparate points on the surface. To select the correct VHIP, an "inching" algorithm is used to move the VHIP incrementally along the surface. This constrains the VHIP to the surface and prevents ambiguities.

The approach works well for synthetic quadric implicit surfaces, but was not sufficiently realistic in simulation with complex anatomic object. The surfaces felt rough and unwanted instabilities in the haptic device is produced, especially in the high frequency region of the volumetric data. The force artifact is more serve if the data are noisy, which is inevitable in medical data. The rough feeling of the data can be reduced by applying smoothing algorithms such as low-pass filtering and morphologic closing operation. Another problem of this approach is that the normal to the surface was approximated rather than exactly calculated, the algorithm occasionally fails to maintain the correct surface contact point. When this occurs, the algorithm provides a zero length force vector, effectively rendering the object haptically transparent. Finally, the incrementally moving of the VHIP and approximation of the surface normal by central difference may require too much computation time for a servo loop of the haptic rendering. We developed a direct haptic rendering method for isosurface in volumetric data by an indirect geometric representation.

3. INTERMEDIATE REPRESENTATION APPROACH

The computation of a single haptic control loop is strongly influenced by the complexity of the object shape. For complex object, it may requires a large amount of computation and long computational time. Consequently, a single haptic control loop may last for too long, and stiff surfaces cannot be represented because of the low servo rate. For this problem, intermediate space had been introduced first by [1], and improved by [9]. The feature of intermediate space is that a virtual environment and the control loop of haptic rendering exchange the information necessary by using a virtual plane which is frequently recomputed. By using the intermediate space, it facilitates the collision detection and reaction force computation, thus stiff surfaces can be represented even if the virtual objects have complex shapes.

The proposed approach, as shown in **Figure 1**, is to use a virtual plane as an intermediate representation of the isosurface in the volumetric data, and compute the point interaction force based on this virtual plane. Most of the time, the simulation is not required to update the virtual plane as fast as the haptic control loop. Some servo loops will thus only use the virtual plane to generate the interaction force which can be quickly computed as the virtual plane structure is so simple that collision detection and force computation can be done instantly. Therefore, we are able to gain higher servo rate for complex volumetric data with this method. Another advantage of this approach is that, when it is applied to a medical volumetric data, the intermediate representation of the isosurface locally by a virtual plane have an effect of smoothing the noisy volumetric data. Finally, since the volumetric data is transferred to an intermediate geometric representation, the algorithm is able to haptically simulate hybrid virtual environments that consist of both volumetric and geometric models.

In the following subsections, we will first discuss how to compute the intermediate virtual plane which captures the local information of the isosurface in the volumetric data. Then we discuss how to update the virtual plane according to various factors, including the underlying surface structure and movement of HIP. However, the simple approach of updating the virtual plane will lead to force discontinuity artifacts. The solutions of the problems will be discussed. Finally, we present the implementation and results.

3.1 Intermediate Virtual Plane

The algorithm of our approach is summarized in **Figure 2**. In each servo loop of the haptic impedance control, the position of the HIP is traced. At the same time, the voxel value v at the position of the HIP is computed by tri-linear interpolation. Applying the binary segmentation function $B(v)$ to the value v, yields 0 if the HIP is located at the background (outside the defined isosurface) and 1 if it is inside the defined isosurface. A ray is constructed from the position of the VHIP (resultant position computed in pervious servo loop) to the current HIP. If two end-points of the ray are both background ($B(v) = 0$), or part of the object ($B(v) = 1$), the ray does not intersect with the isosurface. The VHIP is then allowed to move directly to the HIP and no force will be generated in these cases. Note that even when the ray is completely inside the object (both end-points have $B(v) = 1$), it is regarded as not intersecting with the isosurface and therefore no force will be generated. The reason of this is to avoid large force which exceed the bandwidth of the haptic device be generated, if the HIP is deeply inside the object when the simulation commences.

If the VHIP has $B(v) = 1$ and the HIP has $B(v) = 0$, the HIP is moving out of the object. No force should be generated for this case neither, even the end-points of the ray have different values of $B(v)$. The only case that has force output is when the VHIP has $B(v) = 0$ and the HIP has $B(v) = 1$. This case happens when the ray penetrates the defined isosurface from outside. In order to calculate the interaction force in this case, a tangential plane including the intersection point on the isosurface is worked out. The virtual plane is defined by the equation:

$$\mathbf{N} \bullet (\mathbf{x} - \mathbf{p}) = 0 \quad (1)$$

where $\mathbf{p}$ is the intersection point computed by linear interpolation along the ray and it corresponds to the position where $v = \tau$, the threshold of the isosurface. $\mathbf{N}$ is the normal vector at the intersection point approximated by using central difference. The computation of the virtual plane transmits the local information of the volumetric data to the intermediate space which will be used for servo loop of haptic control. Generally, the position and the orientation of the virtual plane should be frequently re-computed according to the movement of the HIP.

The interaction force is calculated by using the intermediate virtual plane representation of the volumetric data just as constraint based approach for geometric models. The VHIP is moved to a point on the surface where the distance between it and the HIP is the minimum, computed by:

$$\mathbf{VHIP} = \mathbf{HIP} + ((\mathbf{p} - \mathbf{HIP}) \bullet \mathbf{N})\mathbf{N} \quad (2)$$

The reaction force is proportional to the distance between the newly computed VHIP and HIP, by using a virtual spring-damper model, as shown in **Figure 3**.

$$\mathbf{F} = K\mathbf{d} - B\mathbf{v_d} \quad (3)$$

where $\mathbf{d} = \mathbf{HIP} - \mathbf{VHIP}$ and $\mathbf{v_d}$ is the velocity along $\mathbf{d}$. Force computed in this way has a direction in the normal of the surface only. Although it gives user an important sense of realness to perception of objects, we rarely experience frictionless surfaces in real life. As discussed in [11], the surface effects such as static, dynamic, viscous friction, stiffness and texture can be created by solely restricting the movement of the VHIP. These methods are implemented in our system to simulate the static and dynamic, viscous friction.

```
Input:      The current HIP, and previous VHIP.
Output:     The interaction force F.
Algorithm:  Update the virtual plane if necessary.
            Move the VHIP according to the constraint
            of the virtual plane.
            Compute the interaction force based on
            the constraint of the virtual plane.

IF VirtualPlane ≠ (0,0) THEN
    IF c ≥ n THEN
        c = 0
        IF Density(HIP) ≤ τ THEN
            VirtualPlane = (0,0)
        ELSE
            p = LinearInterpolation(VHIP, HIP, τ)
            N = SurfaceNormal(p)
            VirtualPlane = (N, p)
        END IF
    ELSE
        c = c + 1
    END IF
ELSE
    IF Density(HIP) > τ & Density(VHIP) ≤ τ THEN
        p = LinearInterpolation(VHIP, HIP, τ)
        N = SurfaceNormal(p)
        VirtualPlane = (N, p)
    END IF
END IF

IF VirtualPlane ≠ (0,0) THEN
    IF N • (HIP − p) < 0 THEN
        VHIP = HIP + ((p − HIP) • N)N
        d = HIP − VHIP
        v_d = Velocity(d)
        Return F = Kd − Bv_d
    ELSE
        VHIP = HIP
        Return F = 0
    END IF
ELSE
    VHIP = HIP
    Return F = 0
END IF
```

Figure 2: Algorithm HRI for haptic rendering of an isosurface by an intermediate representation

3.2 Updating Virtual Plane

If there is enough processing power, we can just set the update rate of the virtual plane same as the haptic control loop. If the update rate of the virtual plane is moderately fast, the user can feel curved surfaces. However, low update rate makes a bumpy surface felt like a surface of polyhedron. Besides, various kinds of factors also make a bumpy surface, including the curvature of the underlying isosurface and the velocity of the HIP. In general, the update of the virtual plane should be fast enough to correctly capture the curvature of the isosurface when the HIP moves. The transition of the virtual planes is shown in the **Figure 4**. In order to have a smooth feeling of the surface, **D** should be as small as we cannot perceive it.

In our implementation, the update rate of the virtual plane is set to $1/n$ of the update rate of the control loop. The parameter n can be adjusted by the user, according to the isosurface structure, the impedance controller parameters,

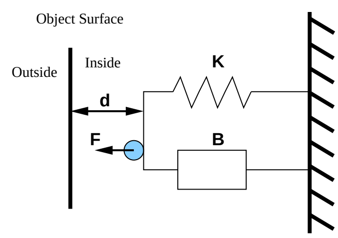

Figure 3: Force is computed by using a virtual spring-damper model.

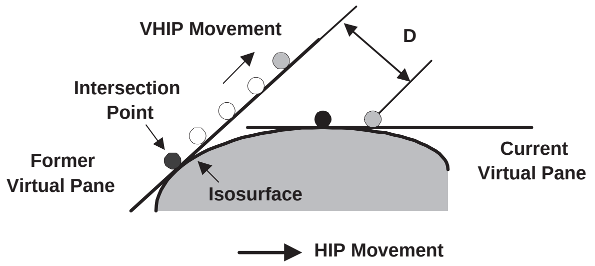

Figure 4: Transition of the virtual planes

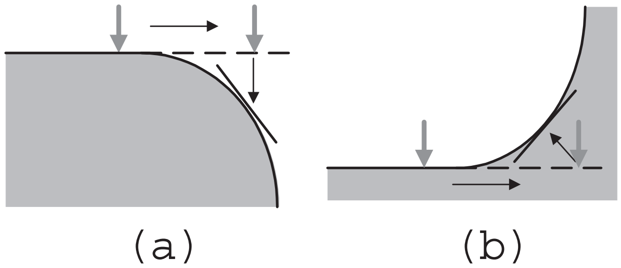

Figure 5: Force discontinuity results if the update rate of the virtual plan is too slow. (a) VHIP suddenly drops, (b) VHIP is embedded in the surface.

and the system load. At the beginning of the simulation, a counter (c) is set to be zero and is incremented at every loop of the haptic control. When c equals n, a new virtual plane should be computed if the HIP is still inside the isosurface. The counter (c) is reset to zero, and the interaction force is computed using the new virtual plane.

3.3 Preventing Force Discontinuity Artifacts

An pointed out above, the intermediate space method works well only with the virtual plane is updated frequently compared to the velocity of the HIP. As shown in **Figure 5**, this will cause problem on sharply-curving surface where the update of the virtual plane is not fast enough to capture the information of the isosurface. A sharp discontinuity occurs in the force model when the HIP is allowed to move large distances before the new virtual plane approximation is computed. If the previous servo loop computation leaves the VHIP outside the surface as shown in **Figure 5a**, the VHIP drops suddenly onto the new virtual plane. Worse, if the VHIP is embedded in the new surface as shown in **Figure 5b**, it is violently accelerated until it leaves the surfaces. Since the computed force is proportional to the distance of this movement, a force with large magnitude will be produced. The force may cause severe artifact or even exceed the bandwidth of the haptic device.

To solve the problem of the force discontinuity when the VHIP is outside the surface, we can increase the update rate of the virtual plane, so that the simulation is fast enough that the dropping distance of the VHIP is unnoticeable. Another remedy is to apply the force shading method discussed at [11]. The additional constraint plane is computed using the gradient approximated by central different at the VHIP of the previous servo loop. This constraint plane is first ap-

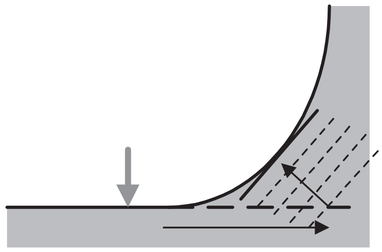

Figure 6: "Recovery time" solution to the problem of extreme force where VHIP is embedded in the surface.

plied to find the temporary HIP, and then the true virtual plane to find the final VHIP. However, this method requires similar computation time for a new virtual plane at each servo loop.

To solve the problem of extreme force when the VHIP is embedded in the new surface, the *recovery time* method [9] can be used. This method is applied during the time immediately after the new virtual plane is calculated. The normal direction for the force is unchanged but the magnitude is reduced so as to bring the VHIP out of the surface over a period of time, rather than instantaneously. The method is illustrated in **Figure 6**. The recovery time is adjustable, and serves to move the VHIP out of the surface gradually in order to smooth the simulated interaction force.

4. IMPLEMENTATION

The Personal Haptic Interface Mechanism (PHANToM™, [10]), distributed by SensAble Technologies, Inc. has evolved as a result of a research at the MIT Artificial Intelligence Laboratory at 1993. The PHANToM™ is a convenient ground-based device which provides a point-based force feedback interface between a human user and a computer. By stressing design principals of low mass, low friction, low backlash, high stiffness and good backdrivability, the system is capable of presenting convincing sensations of contact, constrained motion, surface compliance, surface friction, texture and other mechanical attributes of virtual objects. We have developed and tested our algorithm using the PHANToM™ haptic device.

Figure 7: The PHANToM™ Haptic Interface.

	Intermediate Representation		Blezek and Robb [3]		Avila and Sobierajski [2]	
	Average Time (microsecond)	Servo Rate (kHz)	Average Time (microsecond)	Servo Rate (kHz)	Averate Time (microsecond)	Servo Rate (kHz)
sphere (64x64x64)	39.00	25.64	47.10	21.23	38.30	26.11
knot (64x64x64)	42.00	23.81	49.10	20.37	41.20	24.27
head (128x128x64)	48.00	20.83	59.70	16.75	47.40	21.10
lung (128x128x64)	46.80	21.37	57.60	17.36	46.20	21.65

Table 1: Quantitative comparison of various algorithm of direct isosurface haptic rendering (For the intermediate representation approach, the update rate of the virtual plane is set to be five times slower than the achieved servo rate).

GHOST SDK [6] (which is the development environment of the PHANToM™ haptic device), is a C++ object-oriented toolkit that represents the haptic environment as a hierarchical collection of geometric objects and spatial effect. The direct haptic rendering approach is implemented by extending the GHOST SDK classes and is written in the C++ programming language. **Table 1** shows the quantitative result of the proposed algorithm with the comparsion with the other two approaches. The visual rendering and haptic rendering processes are decoupled, running as separated processes which communicate by only passing the necessary information, in a SGI™ Octane®/^{MXE} R10000 workstation with 384Mb main memory.

The algorithm was first tested with several sets of simple synthetic data which contain implicit surfaces. A box, a sphere and a volumetric data with three spheres of different densities, were rendered haptically and graphically. The isosurfaces rendered by our algorithm are felt correctly based on the visual image. Slower update rate caused the feeling of "step" along the surfaces, increasing the update rate will generally improve the feeling of smoothing. We had set the update rate of the virtual plane to 50 to 10 times slower than the servo loop of the haptic control during the experiment, and at most of the time (if the haptic interface point was not moving rapidly), the curvature of the surfaces was felt smooth.

We then tested the algorithm with the "knot" data defined by a mathematical function. The isosurface structure contained in the volumetric data can be identified correctly. However, there was force discontinuity felt at concave regions if the methods discussed in the **Section 3.3** was not applied. At some place of high curvature, a large reaction force that even exceed the bandwidth of the haptic device was generated. The artifact was removed by applied the solution discussed in the **Section 3.3**.

We also have applied our algorithm to two sets of medical CT-scanned images. The first was a head model, the dataset was thresholded to the level of bone, and rendered both haptically and graphically. The hard structure of bone surface could be clearly felt, even we set the update rate of the virtual plane to tenth of the haptic control loop. The other medical data is a CT-scanned lung model. The soft tissue the organ was able to be haptically located by the algorithm. It is however, the experience was not as realistic as with the skull, due to the inability of the algorithms to cope with the detailed variations on the surface. Same as other volume haptic rendering methods in the literature, the performance of our algorithm towards the surface of soft tissues, can be improved by applying smoothing algorithms to the data (such as spatial low-pass filtering and morphological operation) before haptic rendering.

5. DISCUSSION AND CONCLUSION

We have developed a direct haptic rendering method for isosurface in volumetric data using a point-based haptic feedback device. Our approach uses a virtual plane as an intermediate representation of the isosurface in the volumetric data, and computes the point interaction force based on this virtual plane. We are able to gain higher servo rate for complex volumetric data with this method, as there is less computation for each control loop. Moreover, the update rate of the indirect geometric representation of the volumetric data can be changed according to the system working load and thus easily to maintain the stability of the simulation. Another advantage of this approach when applied to the real volumetric data is that the intermediate representation of the isosurface locally by a virtual plane has an effect of smoothing the noise. Finally, since the volumetric data is transferred to an intermediate geometric representation, the algorithm is able to haptically simulated in hybrid virtual environments that consist of both volumetric and geometric models. By experinment, the approach works well for synthetic implicit surfaces and rigid surface in medical images. When complex soft tissue surfaces are under investigation, if the virtual plane is updated sufficient fast, the method should convey the same haptic information as the volumetric data is first converted to geometrical representation and then rendered by traditional constraint based haptic rendering methods.

6. REFERENCES

[1] Y. Adachi, T. Kumano, and K. Ogino. Intermediate representation for stiff virtual objects. In *Proc. IEEE Virtual Reality Annual Intl. Symp.*, pages 203–210, 1995.

[2] Ricardo S. Avila and Lisa M. Sobierajski. A haptic interaction method for volume visualization. In *Proceedings of the Conference on Visualization*, pages 197–204, October 27–November 1 1996.

[3] Daniel J. Blezek and Richard A. Robb. Haptic rendering of isosurface directly from medical images. In *Medicine Meets Virtual Reality*, pages 67–73, 1999.

[4] F. Brooks, J. Ouh-Young, J. Batter, and P. Kilpatrick. Project grope–haptic displays for scientific

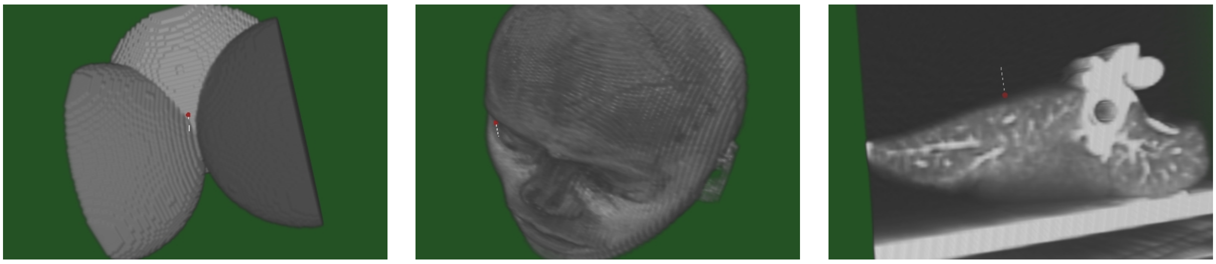

Figure 8: Haptic interaction with various volumetric data.

visualization. *Computer Graphics*, 24(4):177–185, 1990.

[5] S. Gibson, J. Samosky, A. Mor, C. Fyock, and et al. Simulating arthroscopic knee surgery using volumetric object representations, object representations, real-time volume rendering and haptic feedback. In *First Joint Conference on Computer Vision, Virtual Reality, and Robotics in Medicine and Medical Robotics and Computer Assisted Surgery*, pages 369–378, 1997.

[6] SensAble Technologies Inc. *GHOST Software Developer's Toolkit*, 1997.

[7] H. Iwata and H. Noma. Volume haptization. In *Virtual Reality*, pages 16–23. IEEE 1993 Symposium on Research Frontiers in, 1993.

[8] N. Langrana, G. Burdea, K. Lange, D. Gomez, and S. Deshpande. Dynamic force feedback in a virtual knee palpation. *Journal of Artifical Intelligence Medicine*, 6:321–333, June 1994.

[9] William Mark, Scott Randolph, Mark Finch, James Van Verth, and Russell M. Taylor II. Adding force feedback to graphics systems: Issues and solutions. In *SIGGRAPH 96 Conference Proceedings*, pages 447–452, August 1996.

[10] Thomas H. Massie and J. K. Salisbury. The PHANToM haptic interface: A device for probing virtual object. In *Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, volume 1, pages 295–301, November 1994.

[11] Diego C. Ruspini, Krasimir Kolarov, and Oussama Khatib. The haptic display of complex graphical environments. In *SIGGRAPH 97 Conference Proceedings*, pages 345–352, August 1997.

[12] C. Zilles and J. K. Salisbury. A constraint-based god-object method for haptic display. In *Proceedings of the International Conference on Intelligent Robots and Systems*, volume 3, pages 146–151, August 1995.