Haptic Rendering of Milling

Yang Zhengyi and Chen Yonghua

Mechanical Engineering Department, The University of Hong Kong,
Pokfulam Road, Hong Kong
yangzy@hkusua.hku.hk, yhchen@hkucc.hku.hk

Abstract. This paper presents a volume-based haptic rendering method for a haptic shape modeling system based on simulated milling process. Previous proposals on point-based and ray-based haptic rendering methods cannot provide the force feedback in a milling process. The method proposed in this paper adopts a volumetric force model in material removal process. The force is calculated based on the relationship of material removal rate and machining power. Object and tool are both represented by Spatial Run-Length Encoding developed by the authors. Experiments on 5-axis milling process are carried out to determine the coefficients in the force model.

1 Introduction

Rapid developing virtual reality techniques have opened up an opportunity for research in the proposed applications. The haptic shape modeling system allows the complex 3D shape modeling of a product by removing, adding, or deforming virtual materials using virtual tools that have real-time force feedback. With such a system for shape modeling, the user is able to see, touch, feel, model, and manipulate objects with a sense of immersion. Furthermore, since an object is created by virtual machining processes, the manufacturing information of an object becomes an inherent part of the product development processes. This makes it possible to design for manufacturing and plan process directly.

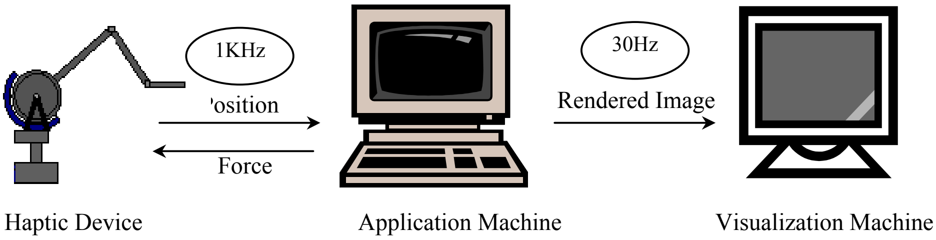


Fig. 1. Logical architecture of a haptic system

In order to emulate the natural surface constraint satisfaction of the milling process, force feedback must be introduced into shape modeling system. With the help of force feedback, the burden of physical constraint satisfaction is transferred to a haptic system, and the designer becomes free to concentrate on higher-level problems such as path planning and engineering rule satisfaction.

Haptic rending of milling process is crucial in the proposed haptic shape modeling system. This paper is emphasized on haptic rendering of the milling process (can be extended to all material removing processes). A volume-based haptic rendering methodology is proposed, in which the cutting force is computed based on the relationship of material removal rate and machining power. Such a volumetric force model is process independent.

Haptic rendering has progressed rapidly from very simple models such as cubes and spheres to very general models made from tens of thousands of polygons or full-featured, trimmed, NURBS models. Polygonal methods, although limited as a surface representation due to compactness and smoothness concerns, currently dominate haptic rendering.

Based on the result of analyzing the human factors, in order to be able to feed the appropriate sensorial input to the human perceptual system, the system needs to update data at two very different frequencies: about 30 Hz for the visual rendering, and above 1KHz for the haptic response [1]. A logical architecture of a haptic rendering system is illustrated in Fig. 1. Therefore, the system should:

- Maintain a high update rate in the force servo loop;
- Present high quality forces without detectable artifacts;

According to the model representing the generic probe of haptic device, haptic rendering techniques are classified into two categories: point-based and ray-based [1].

- Point-based technique: The haptic interaction is surface contact point, the computation of touching surfaces is based on the penetration depth of the end-effector of the haptic device beneath the surface;
- Ray-based technique: The haptic interactions with virtual objects are simulated by using ray-tracing techniques. The probe of the haptic device is modeled as a line segment. Not only forces are generated, but also torques.

In their research, both point-based and ray-based haptic rendering methods based on surface representation are reported.

In point-based method, 3-DOF force feedback is implemented. Because of its simplicity, many haptic rendering systems are point-based [1, 3-5]. Some systems use surface model to represent object and others use volume model.

In ray-based method, 6-DOF force feedback is achieved. A voxel-based approach to haptic rendering is presented in [6], that enables 6-DOF manipulation of a modestly sized rigid object within an arbitrarily complex environment of static objects. A dual-Phantom system is developed by Ho *et al.* based on a ray-based method [2]. Two Phantoms hold both ends of a probe at the same time. Forces and torques due to collisions of the tip and/or side of the probe with object are calculated. An incremental method is used for contact determination between convex primitives in [7]. The resulting contact information is used for calculation of the restoring forces and torques and thereby used to generate a sense of virtual touch.

Besides haptic rendering of rigid objects, researches are carried out on deformable objects. An algorithm called 3D ChainMail is reported in [8]. The algorithm is used for interactive manipulation of deformable objects. Unlike other work where deformation is modeled with complex calculations on a small number of elements, this algorithm performs simple calculations on a very large number of elements to achieve complex behavior. Jansson *et al.* presented a discrete mechanics model for deformable bodies [9]. The model can be viewed as extended mass-spring model. A haptic rendering method of surface-to-surface sculptured model interaction is introduced in [10]. Their work extended the finger-surface interaction to surface-to-surface interaction.

Applications of haptic rendering method can be found in medical field and engineering field. A surgery simulation system is reported in [11]. Volumetric methods are developed to model phenomena such as the deformation, cutting, tearing, and repairing of soft tissues. Thomas *et al.* developed a force feedback dental simulator to train dental students in the haptic skills of dentistry [4]. A haptic scissors is designed to simulate cutting biological tissues [12]. Other medical applications can be found in [13,14]. Applications in engineering field can be found in [4,15-19].

The haptic rendering methods mentioned above are based on tool contact simulation, surface tracing, or object cutting. None of them considered actual material removing processes. This paper proposed a volume-based haptic rendering of milling process. Both the object and the milling tool are represented by a volumetric data structure called Spatial Run-Length Encoding (S-RLE) developed by the authors. The haptic response is implemented with a Phantom haptic arm from SensAble® Technologies.

2 Volume-based haptic rendering

2.1 Haptic shape modeling based on simulated machining processes

The presented volume-based haptic rendering methodology is developed for a haptic shape modeling system based on simulated machining processes. In the authors' research, shape modeling is simulated as a virtual material removal process similar to the milling process. When interactively removing material using a virtual tool such as a ball-end milling cutter, a user can feel the physically realistic presence of the material with force feedback throughout the process. For rough shape modeling, tool paths are un-constrained like a free-hand sculpturing; for more accurate modeling, a tool path is constrained by a structured set of machining parameters.

2.2 Object and tool represented by S-RLE

A volumetric object is a regular or irregular 3D array of data, with each data element representing a sample point in the volume. Volumetric representation has seen several decades of very active research and development [20-24]. However, most of the achievements are concentrated on the visualization based on volumetric model. Haptic rendering is not thoroughly investigated. An extendable volumetric representation based on run-lengths called spatial run-length encoding (S-RLE) is developed by the authors. The data structure consists of two cross-referenced database: one is a stack of lists in geometrical domain, recording the runs describing the space occupation of the object; the other is a table in physical domain, describing the physical properties of each element. The former is called *position array* and the latter *property list*. The property list is extendable to include more physical properties. The element of position array is called voxel. Each voxel has a property index pointing to the property list. If the properties of two voxels are identical, they are called *homogenous voxels*. Each run in position array contains homogenous voxels only. The structure of S-RLE is depicted in Figure 2. Algorithms for haptic and visual rendering are developed. The proposed S-RLE data structure is suitable for haptic rendering because of its efficient memory usage, quick collision detection, inherent representation for heterogeneous objects, and fast visual rendering.

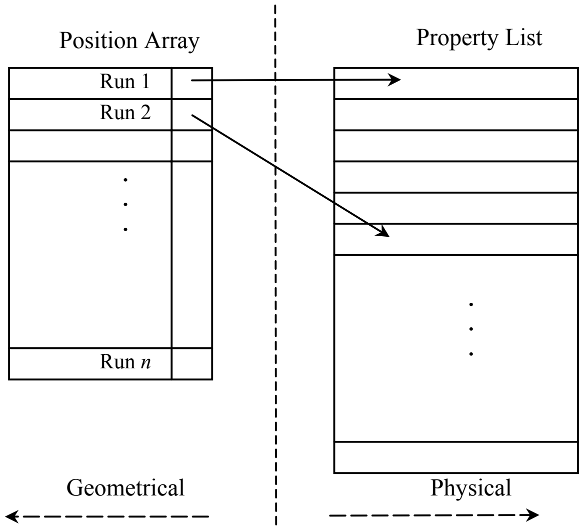


Fig. 2. Structure of S-RLE

In geometric domain, S-RLE is analogous to 2D run-length encoding (RLE). An example of 2D RLE is pictured in Figure 3. The image is represented by 2D RLE as follow,

$y = 2$: (2, 2) (5, 6) (8, 9)
$y = 3$: (2, 2) (5, 6) (8, 8)
$y = 4$: (2, 2) (5, 6) (8, 8)
$y = 5$: (2, 7)
$y = 6$: (2, 2) (5, 6) (8, 8)
$y = 7$: (2, 2) (5, 6) (8, 8)
$y = 8$: (2, 2) (5, 6) (8, 9)

A volumetric 3D object denoted by V can be represented by a 3D array of point voxels $p(x, y, z)$ with integer coordinates as follow,

$$V = \{p\,(x, y, z) \mid 1 \le x \le \boldsymbol{X},\ 1 \le y \le \boldsymbol{Y},\ 1 \le z \le \boldsymbol{Z}\} \qquad (1)$$

where $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$ are the maximum coordinates in the three axis respectively.

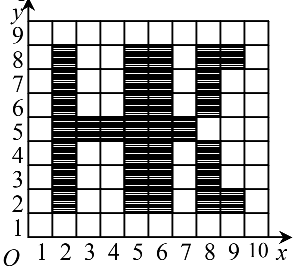


Fig. 3. 2D RLE

In RLE, a 3D object is considered as a stack of 2D slices parallel to a coordinate plane in the coordinate system. The object is represented by enumerating its space occupation in each slice, respectively.

Given an object ***V*** represented by an array of 3D voxels in 3D space ***W***, its position array of S-RLE is constructed by the following steps,

1. Retrieve the first slice at z;
2. Retrieve the first line at y;
3. Retrieve the first voxel at x, store its coordinates in position array as $pa(x_{start}, y, z)$, store its property in property list as $pl[1]$;
4. Retrieve the next voxel at $x = x + 1$ and its property, if voxel $x = 1$ and $x = x + 1$ are homogenous voxels, repeat step 4 until a heterogeneous voxel appears and store the coordinates of last homogeneous voxel scanned into position array as $pa(x_{end}$, y, z);
5. Repeat step 3 until the end of current line is approached, set $y = y + 1$;
6. Repeat step 2 until the end of current slice is approached, set $z = z + 1$;
7. Repeat step 1 until the end of the 3D voxel array is approached.

In physical domain, S-RLE employs a list to store all the concerned properties of a voxel. In the application of haptic rendering of machining, material properties are included.

For modeling based on real machining processes, more detailed description of cutting tool is needed. For example, a milling cutter is divided into two parts: one is the part with cutting edge which is called effective cutting part, the other the cutter shank which is modeled as a cylinder. This is more complicated than existing practice of modeling the haptic probe as a point or a line segment.

2.3 Machining force modeling

Many efforts have been taken on modeling the cutting force in various machining processes, such as turning, drilling, and milling. But not every force model is feasible to be used for haptic rendering that features a relatively high update rate.

The simplest model relates the cutting power P to the material removal rate (MRR) by the following equation [25]:

$$\mathrm{P} = K \cdot (\mathrm{MRR}) \qquad (2)$$

Where K is the unit power consumption. The spindle motor power (P) is equal to the tangential cutting force times the tooth velocity. Therefore, the tangential cutting force is easily found by the following:

$$F_t = K_t \cdot (\mathrm{MRR}) / f \tag{3}$$

where f is the feedrate. The radial force is calculated by multiplying the tangential force by a constant K_r:

$$F_r = K_r \cdot F_t \tag{4}$$

The volumetric model is simple to implement and can be calculated efficiently to meet the need of high update rate in haptic rendering. The values of K_t and K_r depend on workpiece material, cutting tool geometry and cutting conditions.

The estimation of MRR is also simple:

$$\mathrm{MRR} = \mathrm{VR} / \mathrm{T} \tag{5}$$

VR is the material removed in a period of haptic cycle. T is the period of haptic cycle (generally, smaller than 1*ms*).

2.4 VR calculation based on S-RLE

The operations in geometric domain of S-RLE consist of two mains tasks, one is collision detection and the other is volume-removed calculation. Before calculating the VR, collision detection is conducted because it is comparatively simple.

Collision Detection. In machining processes, collisions among tool, tool holder, workpiece, and fixtures must be avoided to prevent from damage. In haptic shape modeling system, the mechanics model changes when collision occurs. Therefore, collision detection is an important function in haptic rendering of machining.

Collision detection is an issue in geometric domain. Once collision between the surface voxels of two objects occurs, the two objects collided with each other. Obviously, use only surface voxel to perform collision detection would be computational efficient. In the position array of an object $\boldsymbol{V}$, the voxels with at least one empty voxel e in their 26-neighborhood defines the surface $\boldsymbol{S}$ of the object,

$$\boldsymbol{S} = \{p \in \boldsymbol{V} \mid \exists\, e \in 26nh(p),\, e \notin \boldsymbol{V}\} \tag{6}$$

where $26nh(p)$ means the 26 neighborhood of the voxel p. Inherently, $\boldsymbol{S}$ is closed in 3D space.

In haptic rendering of machining, the cutting tool and workpiece are represented in S-RLE in 3D space $\boldsymbol{W}$ as $\boldsymbol{V_t}$ and $\boldsymbol{V_w}$ respectively. The surface of cutting tool $\boldsymbol{S_t}$ is represented by,

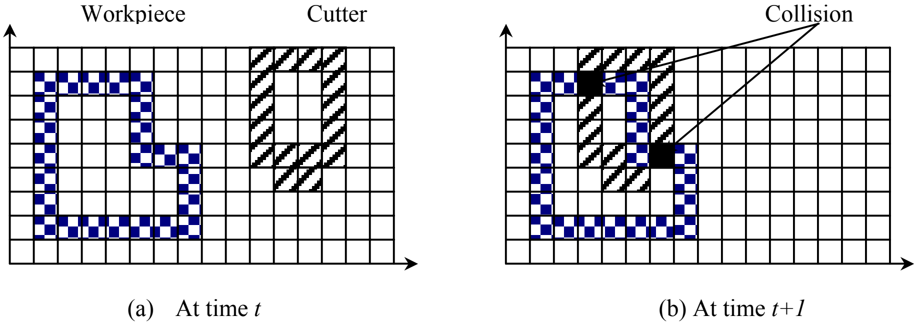
$$\boldsymbol{S_t} = \boldsymbol{S_{te}} \cup \boldsymbol{S_{ts}} \tag{7}$$

where $\boldsymbol{S_{te}}$ and $\boldsymbol{S_{ts}}$ are the surfaces of effective cutting part and cutter shank, respectively. The surface of workpiece is $\boldsymbol{S_w}$.

Collision occurs when $\boldsymbol{S_{ts}}$ and $\boldsymbol{S_w}$ have at least one common voxel p in $\boldsymbol{W}$, and the boundary of collision area $\boldsymbol{C}$ is,

$$\boldsymbol{C} = \{p \in \boldsymbol{W} \mid p \in \boldsymbol{S}_{ts} \text{ and } p \in \boldsymbol{S}_w\} \quad \textbf{(8)}$$

Figure 4 is a 2D illustration of the collision occurrence between the tool and the workpiece. $\boldsymbol{S}_w$ is updated in each haptic rendering cycle because some materials may be removed and new in-process surface generated. The updating is quite simple when the collision area boundary is acquired.



(a) At time *t*

(b) At time *t+1*

Fig. 4. Collision in 2D

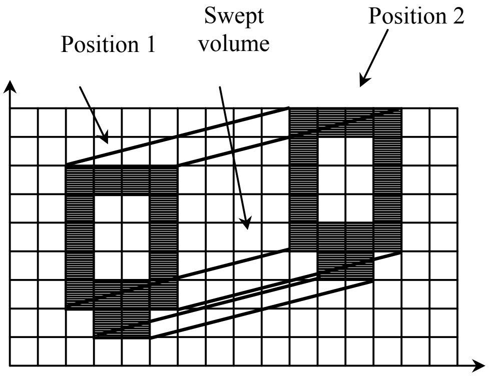


Fig. 5. Cutter swept volume

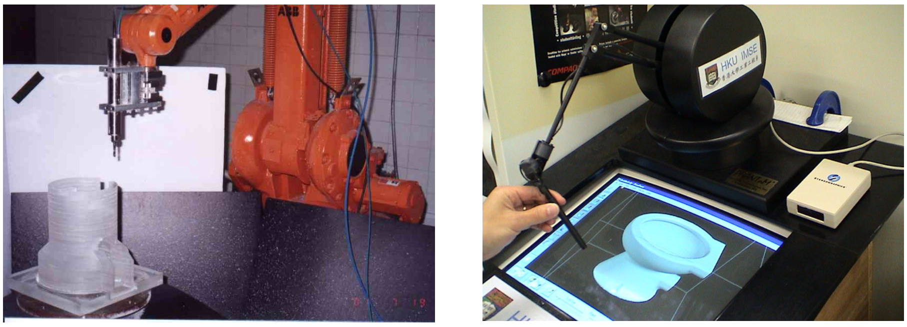
When collision is detected, a brake force is feedback to the user via the haptic device and an alarm is sent to the user visually.

Volume Removed. In machining processes, the volume removed in one cutting cycle is the intersection of workpiece and the tool swept volume.

In haptic rendering, however, the position of the tool is checked by the haptic device in discrete time. The time interval is the period of haptic cycle. If the time interval is 1*ms*, the feedrate is 10*mm*/*s*, and then the tool will move 0.01*mm* in one haptic cycle. Such a distance might be larger than the voxel dimension as illustrated in Figure 5. Thus, the swept volume of cutting tool V_s cannot be calculated directly from the union of cutting tool volume at time *t* and *t*+1. In this paper, tool position interpolation method is employed to compute the tool swept volume.

The interpolation of tool position is calculated from the tool position at time t and time t+1. The beginning position is recorded as a vector $P_t = (x_t, y_t, z_t, \rho_t, \theta_t, \phi_t)$ and the ending is $P_{t+1} = (x_{t+1}, y_{t+1}, z_{t+1}, \rho_{t+1}, \theta_{t+1}, \phi_{t+1})$. Then the tool position $P_{t+\Delta}$ at time $t+\Delta$ is calculated as,

$$P_{t+\Delta} = P_t + (P_{t+1} - P_t)\cdot M \qquad \mathbf{(9)}$$



(a) Robot machining system in real world

(b) Haptic device – Phantom® in virtual world

Fig. 6. Similar configuration between an articulated robot and Phantom®

where M is the transform matrix calculated from P_t and P_{t+1} ,

$$M = \begin{bmatrix} T & S \\ R & 1 \end{bmatrix} \qquad \mathbf{(10)}$$

where T is the translation matrix; R is the rotation matrix; S is the scaling factor (here is set to 1).

According to the feedrate, the number of intermediate tool position n can be decided to satisfy the following inequality,

$$\mathit{Max}(\text{translation of voxel}) < \text{voxel dimension} \qquad \mathbf{(11)}$$

After all the intermediate tool positions are obtained, the tool swept volume $\boldsymbol{V_s}$ is the union of them. And the following calculation of VR is trivial,

$$\mathrm{VR} = \boldsymbol{V_s} \cap \boldsymbol{V_w} \qquad \mathbf{(12)}$$

3 Milling force measuring

In order to determine the coefficients K_r and K_t in the volumetric force model, milling force experiments are conducted based on the combinations of materials, cutting depths, spindle speeds, feed-rates and tool orientations. All current CNC milling machines are significantly different from commercially available haptic devices. However, the robot machining system developed by the authors as shown in Figure 6 is very similar in physical configuration to Phantom®, currently the most popular haptic device. The force measurement from the robotic milling system is a more realistic reflection of haptic devices with articulated arms. The main part of the robotic system is an ABB IRB1400 articulated robot with six-degree-of-freedom, which has a similar configuration to the 6DOF Phantom®. Therefore, dynamics data from the robotic machining system may directly be used for the haptic force feedback in the Phantom device.

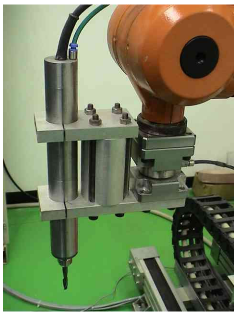


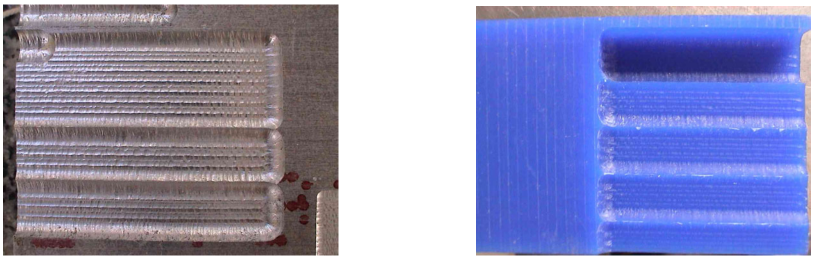
Fig. 7. Tool assembly with force sensor

A tool assembly consisting of a ball-end milling cutter, a tool holder, and a Kistler® 3DOF force sensor is fixed to the robot end-effector, as pictured in Figure 7. The force sensor contains 3 pairs of quartz rings, which are mounted between two steel plates in the sensor housing. Two quartz pairs are sensitive to shear and measure the force components F_x and F_y, while one quartz pair sensitive to pressure measures the component F_z of a force acting on the sensor. The electrical charges proportional to the different components are led via electrodes to the corresponding connectors.

Table 1. Measured milling force data

Step-over (*mm*)	0.1			
Cut depth (*mm*)	0.1			
f (*mm*)	0.1	0.2	0.5	1
F_x (*N*)	2.0103	2.2274	2.3736	3.1328
F_y (*N*)	0.8967	1.1029	1.4637	1.3029
F_z (*N*)	0.7849	1.1326	0.8746	1.1927

A sample set of force data measured is tabulated in Table 1. It is the 3 axis force data of milling Aluminum material Al-6061 with a 2-tooth 6*mm* ball-ended milling cutter. The force data in Table 1 is the average force.



(a) Al-6061

(b) Wax

Fig. 8. Milling experiments on different materials

Force measuring is conducted on milling metal and wax. Figure 8 shows the sample parts milled by a ball-end milling cutter ((a) is aluminum and (b) is model wax).

As shown in Figure 9, given a tool orientation, the vector from the tool center point along its axis is denoted as $\vec{d}$. Then F_r and F_t are calculated by the sum of the projections of the 3 axis forces onto $\vec{d}$ and $\vec{d}_{\perp}$, where $\vec{d}_{\perp}$ is a vector perpendicular to $\vec{d}$.

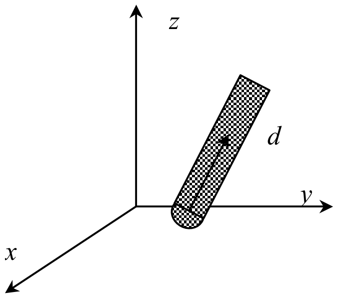


Fig. 9. Coordinate system transform

$$F_t = Proj.(F_x, \vec{d}_{\perp}) + Proj.(F_y, \vec{d}_{\perp}) + Proj.(F_z, \vec{d}_{\perp}) \qquad \textbf{(13)}$$

and

$$F_r = Proj.(F_x, \vec{d}\,) + Proj.(F_y, \vec{d}\,) + Proj.(F_z, \vec{d}\,) \qquad \textbf{(14)}$$

Thus, the two coefficients are calculated,

$$K_t = F_t f\, VR / t \qquad \textbf{(15)}$$

$$K_r = F_r \; / F_t \qquad \textbf{(16)}$$

4 Conclusions

A volume-based haptic rendering methodology is presented in this paper. The force is calculated based on the relationship of material removal rate and machining power. Object and tool are both represented by Spatial Run-Length Encoding developed by the authors to acquire fast collision and intersection computation. Experiments on 5-axis milling process are carried out to determine the coefficients in the force model. The volumetric force model is simple to implement and sufficient for the haptic rendering of machining in the application of haptic shape modeling. Comparing with other mechanistic force modeling methods, volumetric method has the advantage to bring empirical data into force model to develop look-up tables. And these tables could be updated when new material and tools are introduced. The presented rendering method is developed for a haptic shape modeling system based on simulated machining processes.

5 Acknowledgements

This project is generously supported by a grant from Hong Kong Research Grant Council with a grant code HKU 7073/02E.

References

1. Mark, W. R., Randolph, S. C., Finch, M., Verth, J. M. V., Taylor II, R. M.: Adding force feedback to graphics systems: issues and solutions. Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, (1996) 447-452.
2. Ho, C. H., Basdogam, C., Srinivasan, M. A.: Haptic rendering: point- and ray-based interactions. Proceedings of the second PHANToM Users Group Workshop, (1997).
3. Foskey, M., Otaduy, M. A., Lin, M. C.: ArtNova: touch-enabled 3D model design. Proceedings of IEEE Virtual Reality 2002, (2002).
4. Thompson II, T., Johnson, D., Cohen, E.: Direct haptic rendering of sculptured models. Proceedings of Symposium on Interactive 3D Graphics, Providence, RI, USA, (1997).
5. Sachtler, W. L., Pendexter, M. R., Biggs, J., Srinivasan, M. A.: Haptically perceived orientation of a planar surface is altered by tangential forces. Fifth Phantom User's Group Workshop, Aspen, Colorado, USA, (2000).
6. McNeely, W. A., Puterbaugh, K. D., Troy, J. J.: Six degree-of-freedom haptic rendering using voxel sampling. SIGGRAPH 99, (1999) 401-408.
7. Gregory, A., Mascarenhas, A., Ehmann.S., Lin, M., Manocha, D.: Six degree-of-freedom haptic display of polygonal models. Proceedings of the Conference on Visualization, IEEE Visualization 2000, Salt Lake City, Utah, United States, (2000) 139-146.
8. Gibson, S.: 3D ChainMail: a fast algorithm for deforming volumetric objects. Symposium on Interactive 3D Graphics, Povidence, RI, USA, (1997) 149-154.
9. Jansson, J., Vergeest, J. S. M.: A discrete mechanics model for deformable bodies. Computer-Aided Design 34, (2002) 913-928.
10. Nelson, D. D., Jonson, D. E., Cohen, E.: Haptic rendering of surface-to-surface sculptured model interaction. Proceedings of ASME Dynamic Systems and Control Division, (1999).

11. Gibson, S., Fyock, C., Grimson, E., Kanade, T., Kikinis, R., Lauer, H., McKenzie, N., Mor, A., Nakajima, S., Ohkami, H., Osborne, R., Samosky, J., Sawada, A.: Simulating surgery using volumetric object representations, real-time volume rendering and haptic feedback. MERL Technical Report; TR96-16. (1996).
12. Chial, V. B., Greenish, S., Okamura, A. M.: On the display of haptic recordings for cutting biological tissues. Proceedings of the 10th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, IEEE Virtual Reality 2002, (2002) 80-87.
13. Bielser, D., Gross, M. H.: Interactive simulation of surgical cuts. Proceedings of Pacific Graphics 2000, IEEE Computer Society Press, (2000) 116-125.
14. Mahvash, M., Hayward, V., Lloyd, J.: Haptic rendering of tool contact. Proceedings of Eurohaptics 2002, Edinburgh, Scotland, (2002) 110-115.
15. Tanaka, A., Hirota, K.: Virtual cutting with force feedback. Proceeding of VRAIS'98, (1998).
16. Chang, C. F., Varshney, A., Ge, Q. J.: Haptic and aural rendering of a virtual milling process. ASME 2001 Design Engineering Technical Conferences and Computers and Information in Engineering Conferences, Pittsburgh, Pennsylvania, USA, (2001).
17. Dachille IX, F., Qin, H., Kaufman, A.: A novel haptics-based interface and sculpting system for physics-based geometric design. Computer-Aided Design 33(2001) 403-320.
18. Yamamoto, K., Ishiguro, A., Uchikawa, Y.: A development of dynamic deforming algorithms for 3D shape modeling with generation of interactive force sensation. Proceedings of IEEE Virtual Reality Annual International Symposium, Seattle, Washington, USA, (1993) 505-511.
19. Galyean, T. A., Hughes, J. F.: Sculpting: an interactive volumetric modeling technique. Computer Graphics 25(4) (1991) 267-274.
20. Kim, B. H., Seo, J. W., Shin, Y. G.: Binary volume rendering using the slice-based binary shell. The Visual Computer 17(4) (2001) 243-257.
21. Lee, C. H., Koo, Y. M., Shin, Y. G.: Template-based rendering of run-length-encoded volumes. The Journal of Visualization and Computer Animation 9(1998) 145-161.
22. Ferley, E., Cani, M. P., Gascuel, J. D.: Resolution adaptive volume sculpting. Graphical Models 63(2001) 459-478.
23. Kim, T. Y., Shin, Y. G.: Fast volume rendering with interactive classification. Computers & Graphics 25(2001) 819-831.
24. Shen, X. Q., Spann, M.: 3D region representation based on run-lengths: operations and efficiency. Pattern Recognition 31(5) (1997) 575-585.
25. Choi, B. K., Jerard, R. B.: Sculptured Surface Machining Theory and Applications. Kluwer Academic Publishers, (1998).