2011 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 25-30, 2011. San Francisco, CA, USA

Configuration-based Optimization for Six Degree-of-Freedom Haptic Rendering Using Sphere-trees

Xin Zhang Dangxiao Wang Yuru Zhang Jing Xiao

Abstract— **This paper presents a novel constraint-based six degree-of-freedom (6-DoF) haptic rendering algorithm for simulating both contact forces and torques between interacting rigid bodies. We represent an object using a hierarchy of spheres, i.e., a sphere-tree. Such a representation allows fast detection of multiple contacts/collisions among objects and facilitates contact constraint formulation. Given a moving graphic tool as the avatar of the haptic tool in the virtual environment, we constrain its position and orientation, i.e., its six dimensional configuration, by solving a constrained optimization problem. The constraints in the 6-D configuration space (C-space) of the graphic tool is obtained and updated through on-line mapping of the non-penetration constraint between the spheres of the graphic tool and those of the other objects in the three dimensional physical space, based on the result of collision detection. The problem is further modeled as a quadratic programming problem and solved by classic active-set methods. Our algorithm has been implemented and interfaced with a 6-DoF Phantom Premium 3.0. We demonstrate its performance in dental surgery simulations involving complex, multi-contact virtual environments. Our method enables stable operations and realistic feel of haptic sensation.**

I. INTRODUCTION

Haptic rendering has important potential applications in surgery simulations and training [1], [2], [3], [4], [5]. Fig. 1 shows a periodontal operation in dental surgery, which is quite common. A good haptic simulation of such an operation can be used to train dentists. However, haptic rendering based on 3-DoF algorithms for a single point-object interaction cannot simulate multiple contact scenarios that often occur in surgery operations. Moreover, no torque feedback can be provided. Thus, 6-DoF haptic rendering is needed.

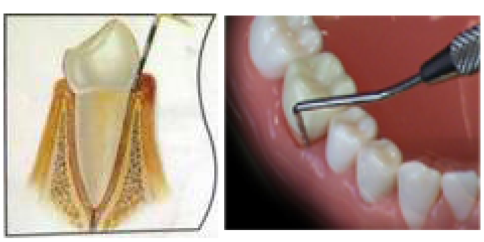

Fig. 1. Periodontal operation: Multiple contacts (between the probe and the tooth/gingival) occur when the dentist examines the depth of the periodontal pocket.

Xin Zhang, Dangxiao Wang and Yuru Zhang are with the State Key Lab of Virtual Reality Technology and Systems, Beihang University, Beijing, China. hapticwang@bubaa.edu.cn

Jing Xiao is with the Department of Computer Science, University of North Carolina-Charlotte, USA. xiao@uncc.edu

In 6-DoF rendering, the graphic tool is treated as a rigid object rather than a point, and both contact force and torque are simulated and rendered.

Several 6-DoF approaches have been proposed in recent years, and we can group them according to the differences of collision response methods they used as: *penalty-based*, *impulse-based*, and *constraint-based* approaches.

In penalty-based approaches, the dynamic model is a dynamic equation based on Newtonian principle. McNeely *et al.* proposed a VPS model which models a dynamic rigid object as a point-shell and a static rigid object as a voxel-map [6], [7]. Barbic and James [8] extended the VPS model into a point-shell model with a signed-distance field. They solved a quasi-static equation with numerical integration to get the position and orientation of the graphic tool. Weller [9] proposed a novel data structure, called inner sphere-tree, based on which a novel penalty-based collision response scheme is defined for providing continuous forces and torques. Johnson *et al.* [10] described a 6-DoF haptic rendering algorithm for arbitrary polygonal models. Local extrema in distance between the haptic tool and the obstacle was computed. The haptic rendering algorithm computed forces and torques on the moving model based on these local extrema. Otaduy and Lin presented a modular haptic rendering algorithm for stable and transparent 6-DoF manipulation, solving the governing equation of the graphic tool using an implicit integration [11]. Penalty-based algorithms may allow interpenetration between the graphic tool and virtual environments or introduce some forms of virtual coupling [12] to maintain the device's stability, which may introduce haptic artifacts by changing the orientation of forces [13].

In impulse-based approaches, contact states between a moving object and a static object are classified as separation and impulse contacts, and a continuous contact state is regarded as a series of micro-impulses [14], [15]. It is applicable to real-time simulations of collisions, but the forces may not be valid for bodies in resting contact.

In contrast to the penalty-based and impulse-based approaches, constraint-based approaches are analytical and global methods for collision response. Constraint-based methods can prevent interpenetration by constraining the graphic tool in the free space or contact space whether the haptic tool interpenetrates the virtual environments or not. In other words, the graphic tool will remain on the surface of an object when contact occurs. Furthermore, the direction of the force is never ambiguous (orthogonal to surface of contact), which allows a more realistic generation of the forces arising from touching an object [16]. A classic 3-DoF

978-1-61284-456-5/11/$26.00 ©2011 IEEE

god-object approach was proposed by Zilles and Salisbury in 1995 [16], which appeared to be the first constraint-based approach for generic polygonal models. Ortega *et al.* [13] extended this approach to 6-DoF, using acceleration as variables and simulating the motion of the god-object by solving the Gauss' projection problem. It is a high quality algorithm, but the use of continuous collision detection [17] is time consuming for complex scenarios involving multiple contacts. Furthermore, explicit Euler integration is utilized to compute the target configuration of the god-object. This integration can cause enough inaccuracy in the configuration of the god-object to make it penetrate the surface of a cavity or even vibrate between two surfaces in a cluttered environment.

We introduce a configuration-based optimization model to optimize the configuration of the graphic tool directly rather than optimizing its acceleration or velocity [18] to avoid the computation cost [13] and inaccuracies in the tool configuration due to numerical integration. Such inaccuracies lead to penetration between the graphic tool and object features. Our approach assumes quasi-static motion of the tool, which is realistic in dental operations, where the velocity of the dental tool is relatively low. For a tool and objects in polygonal meshes, this approach produces high-fidelity rendering [19], but efficient operations are limited to small-region contacts and a simple-shape tool. In order to treat more complex contact scenarios efficiently, in this paper, we consider this approach of configuration-based optimization for the tool and objects modeled in sphere-trees. Sphere-trees facilitate a uniform expression of constraints for complex contacts involving both convex and concave geometric features, which simplifies configuration-based optimization.

The rest of the paper is organized as follows: Section II briefly shows the haptic rendering process of our algorithm. The construction of a sphere-tree is described in Section III. Section IV presents the process of collision detection. Section V describes how the configuration of the graphic tool is obtained as the solution of the constraint optimization problem and how the contact force and torque are computed. Section VI presents the results of applying our approach to dental surgery simulations. Finally, Section VII concludes the paper.

II. Overview of Our Haptic Rendering Method

Our method first obtains the haptic tool's configuration from the device, and then performs collision detection which uses the hierarchies of the sphere-trees to find the intersected leaf spheres and map this information to the C-space of the graphic tool to obtain contact constraints for optimizing the tool's configuration in each iteration of haptic rendering. In order to accelerate the process of optimization, we linearize the constraints using first-order Taylor expansion. We solve for the configuration of the graphic tool by using classic active-set methods. Finally, our methods computes the force/torque by a spring force model and renders the result to the human operator.

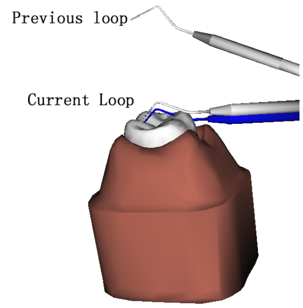

Fig. 2. The blue tool represents the haptic tool and the white tool represents the graphic tool.

In Fig. 2, the haptic tool was coincident with the graphic tool in the previous loop since no contact occurred. During the current loop, the haptic tool has contacted with the tooth and penetrated into it. However, the graphic tool remains on the surface. Imagining that there is a spring between the haptic tool and graphic tool, the latter one always has a trend to approximate the former one but is subject to the constraints.

III. Construction of Sphere-trees

We approximate an object using spheres. As far as we know, the sphere-tree algorithm, which is based on Hubbard's medial-axis theory [20] is one of the best approaches to approximate an object with spheres. The key element of this algorithm is to find the skeleton of an object using a Voronoi diagram and create spheres from the skeleton to give a tight fit to objects. Bradshaw and O'Sullivan [21] have proposed several other algorithms (Merge, Burst, Expand, Spawn) based on Hubbard's and have extended all these algorithms in an adaptive manner. We use Bradshaw's Sphere Tree Construction Tool-kit [22], which contains a number of different algorithms to construct sphere-trees of 3D objects. In particular, we use a combined algorithm to generate sphere-trees with, at most, eight children per node. The combined algorithm allows the use of different sphere reduction algorithms in conjunction and chooses the one that results in the best fit. The sphere-trees of the probe and one of the teeth we will use in Section VI are shown in Fig . 3.

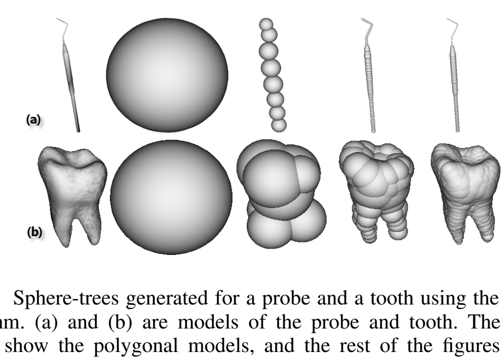

Fig. 3. Sphere-trees generated for a probe and a tooth using the combined algorithm. (a) and (b) are models of the probe and tooth. The left most figures show the polygonal models, and the rest of the figures show the sphere-trees varying from level 0 to level 3.

From Fig. 3 we can see level 3 fits the objects as well as the polygonal model does. We adopt level 3 as the sphere approximation to the objects in implementation, and the collision detection is efficient with such trees of only four total levels. As shown in the attached video, our sphere-tree model of an object appears to provide as good an approximation as the polygonal mesh model of the object.

IV. Sphere-Tree Based Collision Detection

Algorithm 1 outlines the collision detection process between two sphere trees: one for the haptic tool and the other for an object. First, intersection is checked between the two root spheres. If the root spheres intersect, intersections check is conducted at the next lower level and so on until either no intersection is found or all leaf level intersections are found. The output is a set of intersected pairs of leaf spheres, which can be empty if no intersection is detected. Fig. 4 shows an example, where each sphere tree has only two levels for simplicity.

Algorithm 1 **Collision Detection between Two Sphere Trees**

input: sphere-trees of two objects, with root spheres s_1 and s_2
stack $S \leftarrow \emptyset$
$S_{leaf} \leftarrow \emptyset$
if (s_1 intersects s_2)
  push (s_1, s_2) to S
while (S is not empty)
  pop (s_1, s_2) from S
  $S_1 \leftarrow$ children of s_1
  $S_2 \leftarrow$ children of s_2
  for each pair (s'_1, s'_2), $s'_1 \in S_1, s'_2 \in S_2$
    if (s'_1 intersects s'_2)
      if (s'_1 and s'_2 are leaf-nodes)
      then add (s'_1, s'_2) to S_{leaf}
      else push (s'_1, s'_2) to S
output: S_{leaf}

The set S_{leaf} includes all the pairs of intersected spheres of the sphere-trees' leaf-nodes. The next section will show how to map these intersected spheres to constraints in the configuration space (C-space) of the graphic tool.

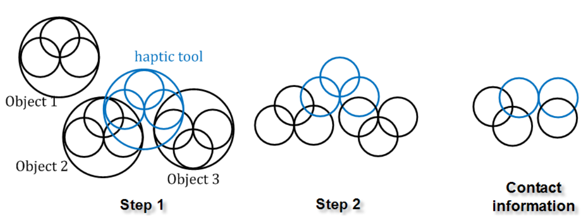

Fig. 4. The schematic of collision detection.

V. Constraint-based collision response

The collision response consists of two different parts. We first construct an optimization model and solve it. Then we model the force/torque applied to the human operator.

A. Configuration-based optimization problem

As we know, an optimization problem consists of an objective, variables, and constraints. These are described below for our problem.

1) Variables: A 6 dimensional vector

$$\mathbf{q}_g^t = (x_g^t, y_g^t, z_g^t, \gamma_g^t, \beta_g^t, \alpha_g^t)^T \tag{1}$$

is introduced to define the configuration of the graphic tool [1], where ^{t} represents the current simulation loop and $_g$ represents the graphic tool.

2) Objective: The objective of our optimization can be defined as minimizing a least-square difference:

$$Minimize : \frac{1}{2}(\mathbf{q}_g^t - \mathbf{q}_h^t)^T G (\mathbf{q}_g^t - \mathbf{q}_h^t) \tag{2}$$

where $\mathbf{q}_h^t$ is the configuration of the haptic tool in the current simulation loop, and G is a diagonal stiffness matrix with $\{k_t, k_t, k_t, k_r, k_r, k_r\}$ in the diagonal [2]. The objective is based on the principle of minimum total potential energy. When the haptic tool intersects objects in the virtual environment, the graphic tool, which is considered connected to the haptic tool by a translational spring and a torsional spring, should stay in contact with the objects in a configuration that minimizes the total potential energy of the springs, as expressed in equation 2.

3) Constraints: We construct a mapping from the contact information in terms of intersecting sphere pairs to the constraints on the graphic tool's configuration in its C-space in the following way.

We choose a pair of intersected spheres $s_T = (x_T, y_T, z_T, r_T)$, $s_O = (x_O, y_O, z_O, r_O)$, where (x, y, z, r) represents the center and radius of a sphere in global coordinates, and the subscript $_T$ and $_O$ represent spheres from the tool and the object respectively. Then the constraint can be written as:

$$(x_T - x_O)^2 + (y_T - y_O)^2 + (z_T - z_O)^2 \geq (r_T + r_O)^2 \tag{3}$$

Since s_O is fixed in the virtual environment, and only s_T moves as the graphic tool moves, we can write (3) in a function form below:

$$C(x_T, y_T, z_T) \geq 0 \tag{4}$$

The variables in (4) are also the functions of $\mathbf{q}_g^t$, and we can establish their mapping using the coordinate transformation (*cf.* equation (7) in Appendix.). Then the constraints in C-space can be expressed as:

$$C(x_T(\mathbf{q}_g^t), y_T(\mathbf{q}_g^t), z_T(\mathbf{q}_g^t)) \geq 0 \tag{5}$$

In order to speed up the convergence of our optimization, we linearize every constraint using Taylor expansion. (*cf.* equation (8) in Appendix)

[1] (x_g^t, y_g^t, z_g^t) represents the graphic tool's centroid, and $(\gamma_g^t, \beta_g^t, \alpha_g^t)$ represents the orientation of the graphic tool respect to the global coordinate system

[2] We choose stiffness $k_t = 1N/mm$ and $k_r = 1000mN \cdot m/rad$ in implementation to fully exploit the ability of our device.

B. Solver of the optimization problem

The objective of the optimization model is quadratic, and the constraints are linear. So this is a typical quadratic programming (QP) problem. Classic active-set methods, which are generally the most effective methods for small- to medium-scale QP problems, perform well in our algorithm.

We can always get the optimal solution in a finite number of iterations, but a good start can shorten this process greatly. The variation of the graphic tool's configuration from one rendering cycle to the next is small due to the high update rates of rendering. In other words, the two optimal points (in C-space) are near each other, so the optimal point in the previous loop can be a good initial point for iteration in the current loop. We choose $\mathbf{q}_g^{t-1}$ as the iteration point $\mathbf{q}_0$, then the active-set methods find a step from one iteration to the next by solving a quadratic subproblem in which a subset of the constraints in (5) are imposed as equalities. Our algorithm continues to iterate in this manner until it reaches $\mathbf{q}^*$ that minimizes the quadratic objective function.

C. Force and Torque Computation

After the configuration of the graphic tool is obtained, the 6-DoF feedback (i.e. force and torque) can be derived using the following model:

$$\mathbf{F} = G(\mathbf{q}_g^t - \mathbf{q}_h^t) \tag{6}$$

VI. Experiments and Discussion

A Phantom Premium 3.0 6DOF is utilized as the haptic device to provide 6 dimensional forces and torques. The specifications of the computer are: Intel(R) Core(TM) 2 2.20GHz, 2GB memory, X1550 Series Radeon graphical card.

A. Periodontal Operation

One important procedure of periodontal operation is periodontal pocket probing examination. Periodontal tissue consists of gingiva, periodontal ligament, alveolar bone and root cementum. Fig. 5 shows the comparison between healthy periodontium and periodontium involved in inflammation. As shown in Fig. 5 c), a periodontal pocket is pathologically deepened gingival sulcus (the shallow crevice between free gingiva and tooth surface). Therefore, periodontal pocket probing is an important operation for periodontal diagnosis. During the examination, a dentist uses a periodontal probe to check the depth value of the pocket, and thus to determine the degree of the periodontal inflammation.

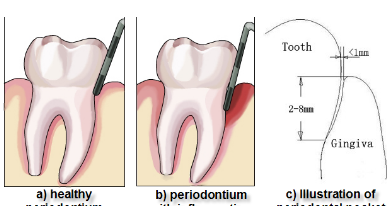

Fig. 5. Periodontal pocket

It is quite challenging to simulate this operation as multiple contacts happen frequently. We checked the stability of our method in this experiment. As shown in Fig. 6, first, the user contacts the front side of the gingival (step 1) and then slides the tool to the front side of the tooth (step 2). When the user slides the the tool from the surface of the tooth to its top side, the change of contact force direction is felt (step 1$\sim$ step 3). And that is why the force changes suddenly in y direction around 9 seconds. Then the user manipulates the tool into the sulcus between the tooth and gingiva at around 20-22 seconds (step 4). During this period of time, contact switches occur frequently; we can see that the force and torque fluctuate. However, the user cannot feel any vibration of the haptic device. A video is provided in the attachment of this paper.

B. Oral Cavity Inspection

In oral cavity inspection, a dentist uses a dental probe to examine possible diseases such as angular stomatitis, glossitis, swollen or bleeding gums, and decayed teeth. In order to simulate this task, a complete oral virtual environment is necessary to simulate the movement of the probe within the oral cavity, which includes all teeth, lower jaw, upper jaw, tongue and cheek. Because the tongue will occlude the teeth during visual inspection of some specific teeth, the dentist has to make sure that no other tissue is damaged when the target tooth is inspected.

From TABLE I, we can see the number of triangles of each object mesh is rather large in this experiment:

TABLE I

Number of polygons and leaf-node spheres for different objects

	Polygons	Leaf-node spheres
Probe	1088	512
Face	3138	512
Upper jaw	2280	512
Lower jaw	2280	512
Tongue	1045	512
Teeth	28×4000	28×512

It is challenging to simulate this operation at haptic rate (i.e. kHz) in this complex virtual environment. Again, our method is efficient and provides stable interaction. The attached video also demonstrates that. Fig. 7 shows our dental surgery simulator (in the right part of Fig. 7). First, we construct one sphere-tree for each object. Then, we simulate this operation in the following steps: 1) examine every single tooth to see if there is any decayed tissue on the tooth surface; 2) examine the gingiva to see if it is inflamed. In Fig. 7, we can tell that the collision detection time and optimization time are related to the number of intersecting pairs of spheres. The time cost of optimization can be maintained at 0.2ms-0.4ms because of the good initial point we choose for iteration (as explained in Section V-B). Collision detection is the major bottleneck for time cost in our computation pipeline. It can be kept within 1ms when the number of intersecting pairs of spheres is 100-200.

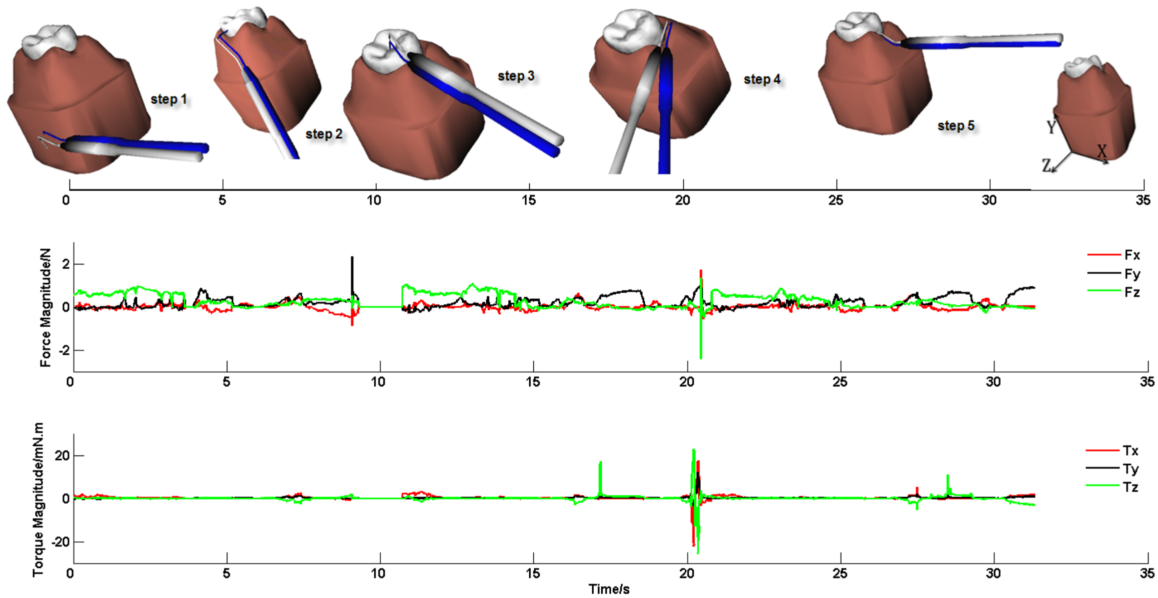

Fig. 6. Haptic simulation of dental examination and the corresponding force/torque applied to the tool as time changes during a period of 32 seconds. The graphic tool (the white one) remains just in contact even though the haptic tool (the blue one) sometimes penetrated into the tooth/gingival. During most of the time, the force and torque change smoothly. Abrupt force occurs in y direction when the tool slides from the front side of the tooth to the top of the tooth at around 9 seconds. Major force and torque occur when contact switches between the tool and tooth/gingival in the sulcus at 20-22 seconds. During the simulation time, the interaction is stable, and the human operator cannot feel any vibration of the haptic device.

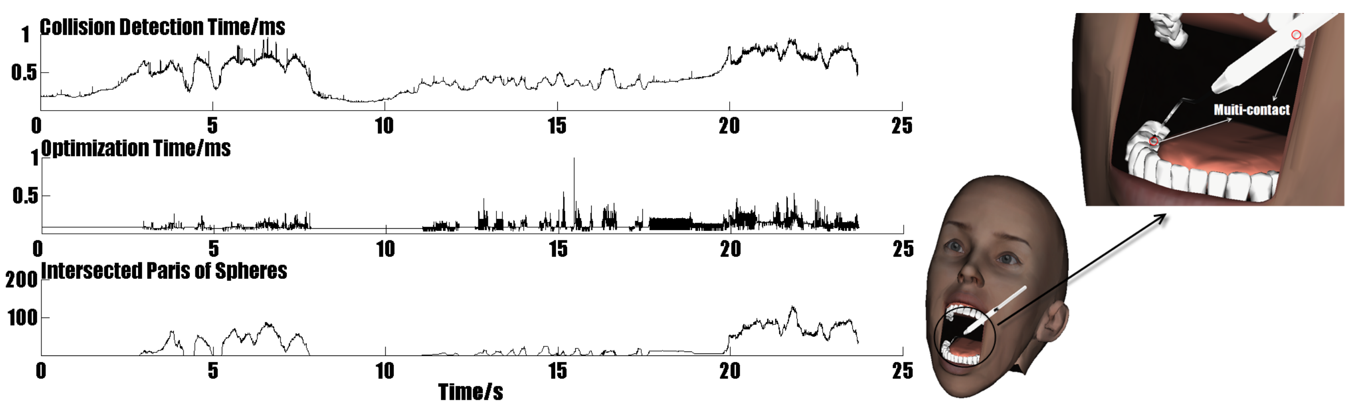

Fig. 7. The right figures show the dental simulation environment with zoomed-in details. The above charts show the computation costs recorded during a haptic interaction period of 25 seconds. The total computation time for one iteration of updates is usually within 1ms even when over 100 pairs of spheres intersect.

C. Discussion

From the above two dental simulation experiments, we can see that our method can provide stable haptic rendering for multi-contact tasks. It should be noted that the time cost of sphere-tree based collision detection will increase when the tool deeply interpenetrates into the virtual objects. We may need to introduce a multi-rate rendering architecture and decouple the collision response loop from the collision detection loop, in order to maintain 1KHz update rate of force/torque computation.

Some limitations in our current method should be addressed in the future. First, the sphere-tree model cannot provide a very accurate fit to an object with sharp features. Also, like the polygonal mesh model, some discontinuity can be felt by tracing a planar surface of an object represented by the sphere-tree model. Thus, as part of our future work, we plan to consider implicit objects to reduce the discontinuity. In addition, though the feel of the force feedback is smooth in most case (as validated in Fig. 6), we do not ensure the continuity of the force in every case, especially when the haptic tool deeply penetrates into the objects. We need to analyze it theoretically and provide a continuous force model.

Second, we need to study how to extend this method to deal with thin or small-sized objects. Although use of discrete collision detection greatly improves the real-time performance of our algorithm, it may lead to "pop through" when the user pushes the tool hard into the objects. Third, simulation of friction and dynamic effects may be another future topic to study. We will consider the method in Ortega's work [13], which produces a convincing effect of highfrequency textures to perturb the force computed. We will also adapt our previous work to simulate friction and other various dynamic effects [23].

VII. Conclusions and Future Work

In this paper, we have presented a constraint-based 6-DoF haptic rendering algorithm using sphere-trees. Our approach treats the simulation of the interaction between a rigid graphic tool and objects in the environments as a constrained optimization problem in the graphic tool's 6-D configuration space. Key to the approach is the mapping of constraints in the physical space to those in the configuration space. The small but effective hierarchical sphere-trees enable efficient collision detection. This approach has been applied to our dental surgery simulator, and the stability and real-time performance meet the requirements of dental surgery training.

In the future, we will improve our method to deal with the limitations discussed in the previous section. We also plan to extend this approach to deformable objects.

VIII. ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China under the grant No. 60605027, and by the research project of State Key Lab of Virtual Reality Technology and Systems of China.

Appendix

First, we use L to represent the local coordinate system of the graphic tool. The vector $\mathbf{q}_g^t$ defined in (1) denotes the position and orientation of graphic tool in the world coordinate system W.

If we transform $\mathbf{q}_g^t$ into matrix form:

$$T = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma & x \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma & y \\ -s\beta & c\beta s\gamma & c\beta c\gamma & z \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

We can construct the relationship between L and W as:

$$\mathbf{x}_W = T \cdot \mathbf{x}_L \tag{7}$$

where $\mathbf{x}_W$ and $\mathbf{x}_L$ represent the homogeneous coordinates of a point in W and L, and this point can be any sphere center of the graphic tool.

Linearizing the constraint conditions in inequality (5) will greatly speed up the iterative process of optimization. So we do it using a first-order Taylor expansion:

$$\begin{aligned}
&C(x_i(\mathbf{q}_g^t), y_i(\mathbf{q}_g^t), z_i(\mathbf{q}_g^t)) \\
&= C(x_i(\mathbf{q}_g^{t-1}), y_i(\mathbf{q}_g^{t-1}), z_i(\mathbf{q}_g^{t-1})) \\
&+ (\frac{\partial C}{\partial x_i}\frac{\partial x_i}{\partial x_g^t} + \frac{\partial C}{\partial y_i}\frac{\partial y_i}{\partial x_g^t} + \frac{\partial C}{\partial z_i}\frac{\partial z_i}{\partial x_g^t})(x_g^t - x_g^{t-1}) \\
&+ (\frac{\partial C}{\partial x_i}\frac{\partial x_i}{\partial y_g^t} + \frac{\partial C}{\partial y_i}\frac{\partial y_i}{\partial y_g^t} + \frac{\partial C}{\partial z_i}\frac{\partial z_i}{\partial y_g^t})(y_g^t - y_g^{t-1}) \\
&+ (\frac{\partial C}{\partial x_i}\frac{\partial x_i}{\partial z_g^t} + \frac{\partial C}{\partial y_i}\frac{\partial y_i}{\partial z_g^t} + \frac{\partial C}{\partial z_i}\frac{\partial z_i}{\partial z_g^t})(z_g^t - z_g^{t-1}) \\
&+ (\frac{\partial C}{\partial x_i}\frac{\partial x_i}{\partial \gamma_g^t} + \frac{\partial C}{\partial y_i}\frac{\partial y_i}{\partial \gamma_g^t} + \frac{\partial C}{\partial z_i}\frac{\partial z_i}{\partial \gamma_g^t})(\gamma_g^t - \gamma_g^{t-1}) \\
&+ (\frac{\partial C}{\partial x_i}\frac{\partial x_i}{\partial \beta_g^t} + \frac{\partial C}{\partial y_i}\frac{\partial y_i}{\partial \beta_g^t} + \frac{\partial C}{\partial z_i}\frac{\partial z_i}{\partial \beta_g^t})(\beta_g^t - \beta_g^{t-1}) \\
&+ (\frac{\partial C}{\partial x_i}\frac{\partial x_i}{\partial \alpha_g^t} + \frac{\partial C}{\partial y_i}\frac{\partial y_i}{\partial \alpha_g^t} + \frac{\partial C}{\partial z_i}\frac{\partial z_i}{\partial \alpha_g^t})(\alpha_g^t - \alpha_g^{t-1})
\end{aligned} \tag{8}$$

References

[1] D. Wang, Y. Zhang, Y. Wang, P. Lv, R. Zhou and W. Zhou, Haptic rendering for dental training system, Science in China Series F: Information Sciences, vol. 52, no. 3, MAR 2009, pages 529-546.

[2] D. Wang, Y. Zhang et al., Cutting on Triangle Mesh Local Model based Haptic Display for Dental Preparation Surgery Simulation, IEEE Transaction on Visualization and Computer Graphics, no. 6, pages 671-683, 2005.

[3] U. Kuhnapfel, H. Akmak and H. Maass, Endoscopy surgery training using virtual reality and deformable tissue simulation, Comput Graphics 24 (2000), pp. 671-682.

[4] V. Hayward, P. Gregorio, O. Astley, S. Greenish, and M. Doyon, Freedom-7: A high fidelity seven axis haptic device with applications to surgical training, in Experimental Robotics. Berlin, Germany: Springer-Verlag, 1998, vol. 232, Lecture Notes in Control and Information Sciences, pp. 445-456.

[5] C. Syllebranque, C. Duriez. Six Degree-of Freedom Haptic Rendering for Dental Implantology Simulation. In Proceedings of ISMBS'2010. pp.139 149.

[6] W. McNeely. K. Puterbaugh, and J. Troy, Six degree-of-freedom haptic rendering using voxel sampling, Proc. of ACM SIGGRAPH, 1999.

[7] W. McNeely, K. Puterbaugh, and J. Troy, Voxel-Based 6-DOF Haptic Rendering Improvements, Haptics-e, vol. 3, no. 7, 2006.

[8] J. Barbic, D. L. James, Six-DoF Haptic Rendering of Contact Between Geometrically Complex Reduced Deformable Models, IEEE Transactions on Haptics, vol. 1, iss. 1, pages 39-52, 2008.

[9] R. Weller and G. Zachmann. 2009. A unified approach for physically-based simulations and haptic rendering. In Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games (Sandbox '09), Stephen N. Spencer (Ed.). ACM, New York, NY, USA, 151-159.

[10] E. Johnson, P. Willemsen, E. Cohen, Six Degree-of-Freedom Haptic Rendering Using Spatial- ized Normal Cone Search, IEEE Transactions on Visu- alization and Computer Graphics, vol. 11, no. 6, pages 661-670, 2005.

[11] M. A. Otaduy and M. C. Lin, A Modular Haptic Rendering Algorithm for Stable and Transparent 6-DoF Manipulation, IEEE Transactions on Robotics, vol. 22, no. 4, pp. 751-762, 2006.

[12] E. J. Colgate, M. C. Stanley, and M. J. Brown. Issues in the haptic display of tool use. Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 1995.

[13] M. Ortega, S. Redon and S. Coquillart, A Six Degree-of-Freedom God-Object Method for Haptic Display of Rigid Bodies with surface properties, IEEE Transactions on Visualization and Computer Graphics, vol. 13, no. 3, pages 458-469 , 2007.

[14] B. Mirtich. Impulse-Based Dynamic Simulation of Rigid Body Systems. PhD thesis, University of Cali- fornia, Berkeley, CA, 1996.

[15] D. Constantinescu, S. E. Salcudean, and E. A. Croft. Haptic rendering of rigid contacts using impulsive and penalty forces. Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on], 21(3):309-323, 2005.

[16] C.B. Zilles and J.K. Salisbury, A Constraint-Based God-Object Method for Haptic Display, Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems, 1995.

[17] S. Redon, A. Kheddar, and S. Coquillart. Fast continuous collision detection between rigid bodies. In Computer Graphics Forum 21 (3) (Eurographics 2002 Proc.), 2002.

[18] C. Duriez, F. Dubois, A. Kheddar, and C. Andriot. Realistic haptic rendering of interacting deformable objects in virtual environments. IEEE Transactions on Visualization and Computer Graphics, 12(1), 2006. 36-47.

[19] D. Wang, X. Zhang, Y. Zhang, and J. Xiao, Configuration-based Optimization for Six Degree-of-Freedom Haptic Rendering for Fine Manipulation, Proc. IEEE International Conference on Robotics and Automation, Shanghai, China, May 2011

[20] P. M. Hubbard. Approximating Polyhedra with Spheres for Time-Critical Collision Detection. ACM Transactions on Graphics, 15(3):179-210, 1996.

[21] G. Bradshaw and C. O'Sullivan. Adaptive Medial-Axis Approximation for Sphere-Tree Construction. ACM Transactions on Graphics, 23(1):1-26, 2004.

[22] G. Bradshaw. Sphere-Tree Construction Toolkit. http://isg.cs.tcd.ie/spheretree/, February 2003.

[23] Q. Luo and J. Xiao, Physically Accurate Haptic Rendering with Dynamic Effects, IEEE Computer Graphics and Applications, Special Issue - Touch-Enabled Interfaces, 24:60-69, Nov/Dec. 2004.