2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

Configuration-based Optimization for Six Degree-of-Freedom Haptic Rendering for Fine Manipulation

Dangxiao Wang*, Xin Zhang*, Yuru Zhang* and Jing Xiao†

Abstract*—Six-degree-of-freedom (6-DOF) haptic rendering for fine manipulation in narrow space is a challenging topic because of frequent constraint changes caused by small tool movement and the requirement to preserve the feel of fine-features of objects. In this paper, we introduce a configuration-based constrained optimization method for solving this rendering problem. The six-dimensional configuration (position and orientation) of the graphic tool is defined as the solution variable of the optimization problem. Contact constraints are obtained based on identifying principal contacts between the graphic avatar of the haptic tool, called the graphic tool, and the virtual task environment. In order to maintain stability during contact switch, a hybrid method combining collision detection, local search and parallel optimization is introduced. Based on parallel optimization and selection of local solution, we can maintain the local solution of the optimization model. Our method has been validated in experiments of moving a convex tool to probe a narrow cavity with or without bulges. Force rendering is stable even when the free space is very small and involves fine features of objects. Non-penetration between the tool and the object forming the cavity are maintained under frequent contact switches. Update rate of the simulation loop including the optimization and constraint identification is maintained at about 1kHz.

***Index terms*—configuration–based optimization, 6-DOF haptic rendering, constraint identification, non-convex free space**

I. INTRODUCTION

6-DOF haptic rendering is an important topic that has potential applications in virtual prototyping, surgical simulator for virtual training, and games [1-6]. For example, as shown in Fig. 1 a), in periodontal simulation, 6DOF interaction between the dental probe and oral tissues (teeth and gingival) need to be simulated for calculus removal. As shown in Fig. 1 b), during maintenance of an air-engine, a spline shaft need to be inserted into splined hole and the force feeling against fine-features need to be preserved during the insertion. Fine manipulations in those examples require dealing with complex contact scenarios with high accuracy in haptic rendering in addition to stability.

A. Related Work

There are two components in a typical 6DOF haptic rendering framework (connected with an impedance type haptic device) [7]: an optimization module to compute the visual avatar, and a module to compute force and torque.

*State Key Lab of Virtual Reality Technology and Systems, Beihang University, Beijing, China.
†Department of Computer Science, University of North Carolina-Charlotte, USA.

To compute the visual avatar of the haptic tool, which we call graphic tool, there are three kinds of collision response methods: penalty-based methods, constraint-based methods, and impulse-based methods.

a) Narrow cavities produced by teeth and gingival (b) a spline shaft need to be inserted into a splined hole

Fig.1 Two examples of fine manipulation.

In the penalty-based approach, the dynamic model is a dynamic equation based on Newtonian principle. There are two subclasses within this method: dynamic simulation with numerical integration [1] [2], and quasi-static equilibrium with numerical integration (QSA approach) [3] [8, 9].

In contrast with the penalty-based approach, constraint-based approach is an analytical and global method for collision response. There are two subclasses within this approach: using velocity as the solution variable [10][11] and quasi-static simulation with numerical integration[12]. In [12], the authors tried to extend the 3DOF God-object method [13] to 6DOF. They combined quasi-static and continuous collision detection to obtain the 6DOF God-object configuration.

In the impulse-based approach, 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].

To compute feedback force and torque signal, there are two methods: penalty force using a direct-rendering architecture, or virtual coupling of either quasi-static model or dynamic model with velocity.

Ortega et al. summarized existing 6-DOF constraint-based haptic rendering methods [12] and pointed out that none seemed to preserve all of the properties of the 3-DOF approach introduced by Zilles and Salisbury [13]. These methods may allow the virtual objects to interpenetrate, or they use some form of virtual coupling that can lead to disturbing force artifacts by modifying the force orientation applied to the user. These methods do not attempt to prevent the interpenetrations between virtual objects, which may lead to missing collisions and the well-known pop-through effect, where the graphic tool can traverse through thin objects or features, thereby degrading the perception of geometric details.

978-1-61284-385-8/11/$26.00 ©2011 IEEE

B. Problems in Fine Manipulation

Haptic rendering of fine manipulation presents several difficulties that have not been well taken into account in the existing literature:

1. Narrow space constraints.
   1) Often the space for the graphic tool to manoeuvre is very small, such as in dental probe: there is only a narrow cavity between the tooth and the gingival.
   2) Frequent constraint change (or contact switch): as shown in Fig. 2, a small movement (translation and/or rotation) of the tool from time t1 to time t2 will lead to a change of contact/constraint status, which includes change of the number of contact regions or change of the contact/constraint type within each region.
   3) Multiple contact regions: there exist multiple contact regions between the tool and the cavity and multiple contact points within each region.
2. Fine-feature preserving requirement.
   1) Movement around fine features including sharp features (for example, an edge-edge contact between a spline shaft and a splined hole) need to be rendered; the force feeling caused by the shape of the features should be preserved.
   2) Pop-through (i.e. tunnelling effect) against thin-objects or small-size features on the object's surface should be avoided.

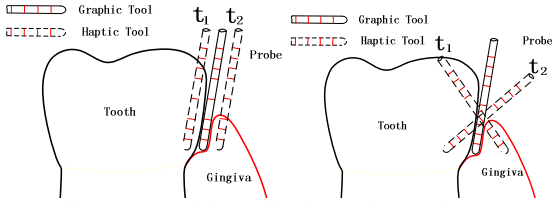


Fig.2 Small movement of the tool leads to a change of contact constraint status and force/torque discontinuities.

Because of the above difficulties, there are two challenges for haptic rendering of fine manipulation:

- How can the force artifact be prevented to maintain both the stability and fidelity of force rendering? Frequent contact switches in a narrow cavity will lead to more frequent force and torque discontinuity. With a penalty-based method, these discontinuities will make the haptic device unstable. Some previous work [1] [2] [3] [9] used virtual coupling to ensure stability but force fidelity is decreased as the result.
- How can the visual artifact be prevented to maintain the accuracy of the graphic tool configuration? When a contact occurs, neither penetration nor separation should occur between the graphic tool and objects. When the allowed free space for the tool is very small, slight computation inaccuracy in the position of the tool can make it penetrate the surface of a cavity or even vibrate between two surfaces. In previous work [1-3] [9] [12], numerical integration is used to compute the graphic tool, which causes inaccuracy.

C. Organization of This Paper

The remainders of the paper are organized as follows. In section 2, we provide an overview of our 6-DOF haptic rendering approach and propose a configuration-based optimization model. In section 3, we describe a hybrid approach for constraint identification by combining collision detection and local search in order to preserve the relevant local solution of the optimization. In section 4, we explain the implementation details of our method. In section 5, the proposed rendering method is validated inside a narrow space and under contact switches. In section 6, conclusions and future work are provided.

II. Overview of our 6-DOF Haptic Rendering Method

To avoid the computation error caused by penalty-based rendering approaches, we propose a constraint-based haptic rendering method, which extends the 3-DOF god-object method [13] to 6-DOF haptic rendering. We model both the tool and the manipulated object as triangle meshes.

A. Our Framework

The flowchart of our 6-DOF haptic rendering method is shown in Fig. 3. The contribution of our method lies in three aspects: (1) a ***configuration-based optimization method*** to compute contact configurations of the graphic tool, in order to avoid visual artifacts, i.e. no perceptible penetration or separation between the graphic tool and an object, (2) a ***constraint identification method*** to detect contact switches and formulate the set of constraints for the optimization model, and (3) a force/torque computation model ***without using virtual coupling***, to eliminate haptic artifacts and to maintain stability.

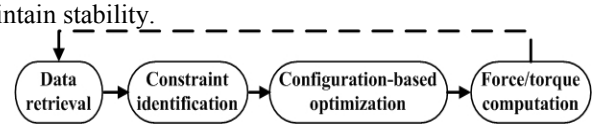


Fig.3 Flowchart of our 6DOF haptic rendering method.

B. Configuration-based Optimization Model

Under narrow cavities, the configuration, i.e., position and orientation, of the graphic tool should be strictly restricted by multiple closely-located contacts. Thus, we establish a constraint-based optimization model to optimize the configuration variable directly rather than optimizing acceleration or velocity [10][12] to avoid inaccuracies in the tool configuration due to numerical integration. Such inaccuracies lead to penetration between the graphic tool and object features.

Another benefit of this model is that we can compute the tool configuration without incorporating typical mesh-mesh continuous collision detection (such as FAST or C2A [15-17]) and thus to avoid the associated computational burden.

The input and output of our optimization problem are:

- Given: (1) the configuration of the ***haptic tool*** $\bar{q}_h$ at the previous and current time step, and (2) the shape and configuration of the object (e.g. a tooth).

- Compute: (1) the configuration of the ***graphic tool*** $\vec{q}_g$ while ensuring no-penetration between the graphic tool and the object, (2) the feedback force and torque $\vec{F}$.

We propose the following configuration-based optimization model to solve for $\vec{q}_g$.

$$\begin{cases} \min: f(\vec{q}_g, \vec{q}_h) \\ Subject \quad to: \quad V_O \cap V_T = \Phi \end{cases} \tag{1}$$

where

$$\begin{cases} \vec{q}_h = [x_h, y_h, z_h, \gamma_h, \beta_h, \alpha_h]^T \\ \vec{q}_g = [x_g, y_g, z_g, \gamma_g, \beta_g, \alpha_g]^T \end{cases} \tag{2}$$

$f(\cdot)$ is an energy function to describe the difference between configurations of the haptic tool and the graphic tool. The function is defined as follows:

$$f(\vec{q}_g, \vec{q}_h) = \frac{1}{2}(\vec{q}_g - \vec{q}_h)^T M (\vec{q}_g - \vec{q}_h) \tag{3}$$

where M is a 6×6 diagonal matrix to describe weight ratios between translational and rotational components [18].

In equation (1), V_O denotes the volume of the object, V_T denotes the volume of the graphic tool. The constraint is to maintain exact contact between the graphic tool and the object, i.e. there is no penetration and no perceptible separation when contact force or torque occurs. The biggest challenge in our optimization model is how to formulate the constraint in equation (1). We will discuss this issue in section 3.

After we compute the graphic tool, the 6-DOF feedback force and torque can be derived using the following model

$$\vec{F} = K_e(\vec{q}_g - \vec{q}_h) \tag{4}$$

where K_e is a 6×6 diagonal matrix that defines a linear relationship between the force/torque signal and the position/orientation signal. The elements in the matrix are selected to fully exploit the display capability of the haptic device, i.e. to use the maximum stiffness of the device. The elements in the first three rows represent the maximum translational stiffness that can be simulated by the haptic device (e.g. 1N/mm for Phantom Premium 3.0). The elements in the last three rows represent the maximum rotational stiffness (e.g. 30mNm/rad for Premium 3.0).

C. Local Solution of the Optimization Model

We compute the solution of the graphic tool based on the motion history of the haptic tool, and thus to get the relevant ***local solution*** instead of the global solution of the optimization problem. Since we consider motion coherency, we need to add the following constraints to the optimization model

$$\vec{q}_g^{\,i}(j) \in \left[\vec{q}_g^{\,i-1}(j), \vec{q}_h^{\,i}(j)\right] \quad j = 1, 2, \cdots, 6 \tag{5}$$

where $\vec{q}_g^{\,i}$ and $\vec{q}_g^{\,i-1}$ is the 6-dimensional configuration of the graphic tool at the current and the previous time steps respectively, as shown in Fig. 4.

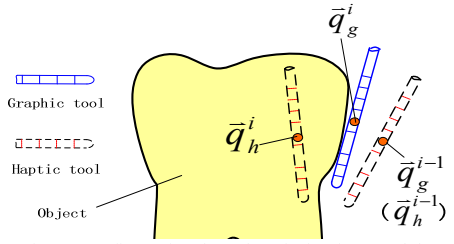


Fig.4. Configuration-based optimization model

Based on equation (5), a local minimum distance (i.e. local solution) instead of the global minimum distance (i.e. global solution) is computed. This local solution is necessary to maintain correct response under typical cases: to avoid pop-through (i.e. tunnelling effect) against thin-objects or small-sized features on an object's surface (see Fig. 5-a), and to avoid the configuration ambiguities of the graphic tool against convex corners or sharp features (see Fig. 5-b). As shown in Fig. 5-b, the local solution is more appropriate than the global solution to model the contact situation while the haptic tool moves.

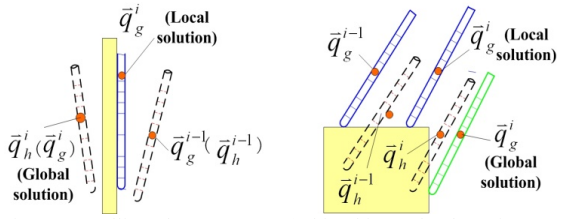


Fig.5. Tunnelling phenomenon: (a) thin object, (b) sharp features

III. CONSTRAINT IDENTIFICATION

In our optimization model, the main challenge is to express the constraint in equation (1) in terms of a set of easy-to-compute constraint equations. The challenge is due to the following:

(1) Variety of the contact scenarios: Because both the tool and the object are modelled by triangle meshes, different contact scenarios will occur between the tool and the object [19]. Constraint equations have to be established for these different contact scenarios.

(2) Modelling of non-convex free space: Typical contact modelling between a vertex and a face cannot handle cases where the shape of the free space in the neighbourhood of the contact point is non-convex [20].

(3) Transitions among contact states during the tool's movement: between adjacent simulation steps, contact status may change. Not only the number of contact regions will increase or decrease, but also the contact points within each region will change.

(4) Multiple-point contacts: Simultaneous contacts in multiple regions and multiple contact points within each region.

A. Types of Contacts for Single Contact Region

We use a matrix of contact types to denote different pairs of contact primitives between the graphic tool and the object. For a mesh-mesh contact, the contact type in each contact

region can be described by contact pairs between the geometric primitives, i.e. vertex, edge and triangle on each mesh. A convex edge is characterized by that the angle between the two adjacent surfaces (of the edge) is greater than 180 degrees. A convex vertex is formulated by three or more convex edges. Restricting our focus on the interaction between a convex-shaped tool and arbitrary shaped objects, we may classify the contact types into 12 possible types of principal contacts (PCs), as shown in table 1 and Fig. 6.

Table 1 Types of principal contacts between a convex tool and an arbitrary object

Object / Tool	Vertex		Edge		Triangle
	convex	concave	convex	concave	
Vertex (convex)	*Type 9*	Type 4	*Type 8*	Type 3	**Type 1: V-F type**
Edge (convex)	*Type 10*	-----	**Type 2: E-E type**	Type 5	Type 6
Triangle	Type 11	----	Type 12	----	Type 7

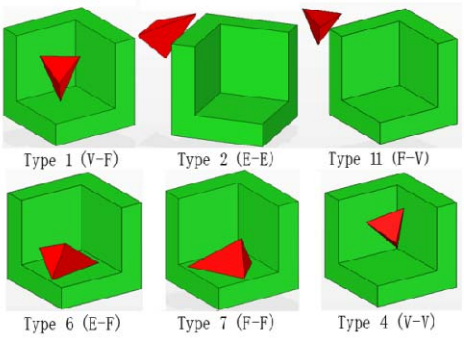


Fig. 6 Some principal contacts between a convex tool (in red) and an arbitrary object (in green).

The vertex-surface contact (V-F) and edge-edge contact (E-E) can be called single PCs. Other 10 types in Table 1 can be derived as combinations of these two types or by symmetry:

- ***Single PC***: Type 1 (V-F), Type 2(E-E);
- ***Combination PCs***: Types 3-7 and Types 11-12. We can convert E-F/F-E types into combinations of V-F/F-V, or convert F-F type into combinations of V-F;
- ***Degenerate PCs***: Types 8-10.

Constraint equations for these PC types can be established by the following method.

Type 1: vertex-surface contact type (V-F type)

The constraint in equation (1) can be expressed as

$$g[\bar{p}_t] = n_o^T \cdot (\bar{p}_t - \bar{p}_o) \geq 0 \tag{6}$$

where $g[\cdot]$ represents a uni-lateral constraint formed by each triangle on the object's surface. $\bar{p}_t$ refers to a boundary point or a vertex on the surface of the graphic tool. $\bar{p}_o$ refers to one of the three vertices of the constraining triangle. $\bar{n}_o$ refers to a unit normal vector of the constraining triangle.

Type 2: edge-edge contact type (E-E type)

For an edge-edge contact, we model it with dynamic vertex-surface contact constraints. The variation from Type 1 is how to determine the normal $\bar{n}_o$ and the point $\bar{p}_o$. As shown in Fig. 7, a dynamic constraint plane is constructed by the two intersecting edges, with normal:

$$\bar{n}_o = E_O \times E_T \tag{7}$$

where E_O is the edge vector on the object, and E_T is the penetrated-edge vector on the haptic tool's surface. Edge E_O is located on the dynamic constraint plane.

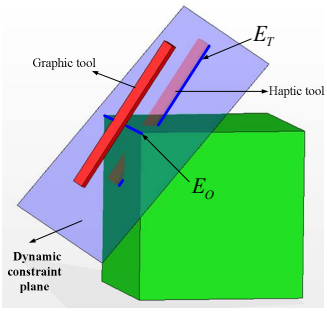


Fig. 7 A dynamic constraint plane to model an edge-edge contact

Combination types

For example, we can derive the constraint inequality for Type 4 contact (a vertex VS. a concave vertex) as follows

$$\begin{cases} g_1[\bar{p}_t] \geq 0 \\ g_2[\bar{p}_t] \geq 0 \\ g_3[\bar{p}_t] \geq 0 \end{cases} \tag{8}$$

where g_1, g_2, g_3 denotes uni-lateral constraints formed by three planes forming the concave vertex of the object.

B. Contact Formations for Multiple Contact Regions

We describe a contact state involving multiple contact regions as a contact formation (CF) [21]. For each contact region, we identify a principal contact (PC), which belongs to the 12 types defined in table 1. Thus, a contact formation is a set of PCs at any simulation step ti :

$$CF^i = \{PC_1^i, PC_2^i, \cdots PC_k^i\} \tag{9}$$

where k is the number of contact regions.

After the CF is identified, optimization with multiple constraint inequalities is used for multiple contact regions.

C. Identification of CF during Contact Switches

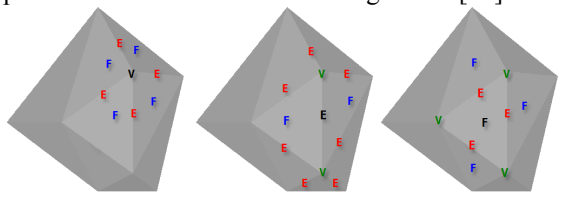
Contact switches are frequent in fine manipulations. The correct contact state needs to be determined from the relative motion between the haptic tool and the object. We proposed a hybrid method that combines collision detection and local search to detect contact switches between adjacent time steps. For convenience, we define the following concepts:

- ***First Principal Contact (FPC):*** the pair of colliding geometric primitives between the haptic tool and the object obtained from collision detection;
- ***Candidate PCs (CPCs):*** all the pairs of geometric primitives between the graphic tool and the object that are in the neighbourhoods of the corresponding primitives of the FPC. As there are three types of FPCs: V-F, E-E and F-V type, the corresponding primitives on the graphic tool's surface are V, E and F. Their neighbouring primitives are illustrated respectively in Fig. 8. Similarly, we can define the neighbouring primitives on the object's surface.

- ***Active Principal Contact (APC):*** the CPC that produces the optimized local solution in the current step;

There are three key components in our hybrid method: (1) collision detection is used to find the ***FPC*** when a ***collision*** occurs or ***separation*** first occurs; (2) local search is used to find ***CPCs***; (3) optimization is used to find the configuration of the graphic tool and the ***APC*** given the ***CPCs*** and the haptic tool's configuration. We provide a detailed flowchart of our method in section 4.

Because the update rate is high, the movement of the haptic tool is small enough in adjacent time steps; therefore, we only use the primitives adjacent to the primitives of ***FPC*** on the tool and the object to form ***CPCs.*** We construct a pre-defined lookup table to find such immediate neighbours [21].



a) V b) E c) F

Figure 8 Neighbouring primitives on the graphic tool's surface of primitives V. E. and F respectively

IV. COMPUTATION STEPS OF 6-DOF RENDERING METHOD

The computation steps of our introduced 6DOF haptic rendering method are the following:

1. Data retrieval: Get the configuration of the haptic tool in the current update loop, and get the configuration of the graphic tool in the last update loop;
2. Identification of PCs
   1) Collision detection: If no contact occurs in the previous time step, use simple continuous collision detection (simple CCD, i.e. tracking all vertices of the haptic tool) to detect possible penetrations between the haptic tool and the object. If contacts occur in the previous time step, first detect possible new ***FPCs*** in new contact regions between the tool and the object by using the collision detection packages PQP [22], and then combine them with the ***APC*** in previous time step to formulate ***FPCs***;
   2) Find all ***CPCs*** by local search;
3. Parallel optimization:
   1) Formulate the set of inequalities for each ***CPC***.
   2) Compute the goal function as equation (3).
   3) Compute a linearized model of the non-linear constraints. In the constraint inequalities (6) and (8), there exist non-linear terms (i.e. trigonometric functions) because of the orientation components in $\vec{q}_g$ [10]. We use Taylor expansion as the linear model for the nonlinear constraints. The truncation error is smaller than $O\left(\Delta\vec{q}_g^{\,2}\right)$.
   4) Construct the Jacobin matrix from the collection of constraints functions [12].
   5) Solve the optimization problem using the active set method [23] while the constraints are considered using the Lagrange factor. The initial solution/condition for each candidate is different. Since 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, the optimal points in the two cycles are close to each other. Thus, the optimal point in previous cycle can be a good initial point for iteration in the current cycle. We choose $\vec{q}_g^{\,t-1}$ as the iteration point $\vec{q}_0$, then the active-set methods find a step from one iteration to the next by solving a quadratic subproblem.
4. Local solution selection:
   1) Select the optimal solution from the optimized results of all ***CPCs***. Based on equation (5), the local solution can be maintained to reflect the desired characteristics of haptic rendering.
5. Compute the feedback force and torque using the force and torque model in equation (4).
6. Update the contact formation vector in terms of ***APCs***, and then wait for the next simulation loop.

V. EXPERIMENTS

A Phantom Premium 3.0 6DOF is utilized as the haptic device to provide 6-dimensional forces/torques (Fig. 9). The specifications of the computer are: Intel(R) Core(TM) 2 2.20GHz, 2GB memory, X1550 Series Radeon graphic card. In order to validate our algorithm, we propose three performance metrics:

- Responsiveness: when a contact switch occurs, there is no visual artifacts (i.e., no interpenetrations between tool and the object), and no haptic artifacts (i.e. forces felt at a distance, or artificial friction and sticking).
- Stability: computational stability of graphic tool and force/torque stability, defined as that big jump or vibration of the graphic tool should not occur between two adjacent simulation time steps.
- Computation update rate: the update rate should be higher than 1kHz under various contact scenarios.

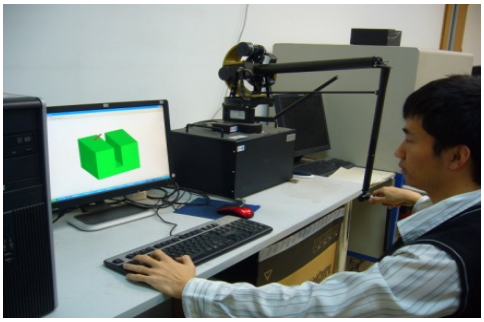


Figure 9 Interaction with a PHANToM Premium 3.0/6DOF device

A. Performance within Narrow Space

Haptic rendering experiments for a cube-shaped tool moving inside convex free space have been carried out. In Fig. 10, a cube-shaped tool can interact with a narrow cavity

produced by the internal volume of another cube-shaped object. There was a small free space (the clearance is 1mm in axis-Y direction) for the tool to move with rotation or translation within this narrow cavity. In Fig. 11, the force and torque signal during five interaction states are illustrated. The results show that the stability of the rendering can also be maintained, and no penetration occur. Even if we decrease the clearance to zero, no haptic/visual artifact occur.

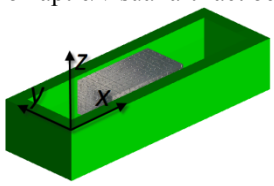


Figure 10 Motion of the tool in a convex narrow cavity

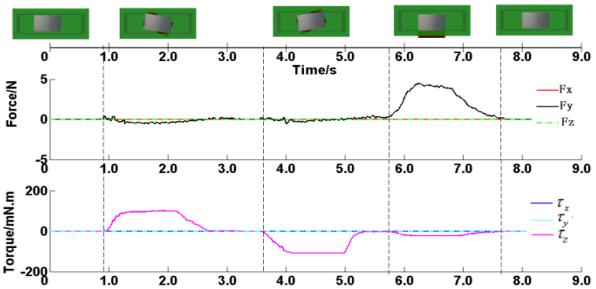


Figure 11 Force and torque signal during five interaction states (no contact, rotate clockwise, rotate anti-clockwise, translate, no contact)

Truncation error caused by Taylor expansion is analyzed. When the tool slides along the object's surface, the result of the error is recorded. Because the change of configuration in one update loop is relatively small, we can find the error values always less than 1 micrometer. This error is small enough and is negligible for causing visual artifact.

B. *Contact Switch Rendering*

It is difficult for haptic rendering when a contact switch occurs, especially if the free space for the tool is non-convex (Fig. 12). In such a case, a principal contact cannot be modeled by a simple combination of V-F contact pairs.

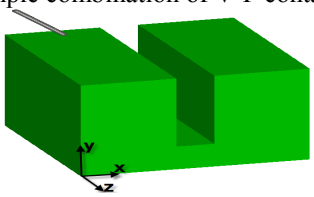


Figure 12 Motion of the tool within free space with both convex and non-convex features

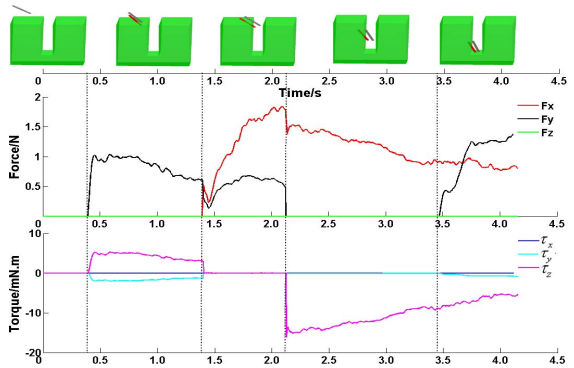


Figure 13 Force and torque signal during contact switches (no contact, V-F contact, E-E contact, V-F contact, V-F contact in two regions)

In one of our experiments, a line-shaped tool moves along both convex and concave geometric features of an object. The goal of the experiment includes two aspects: (1) to simulate different contact types in table 1, (2) to avoid the jump of the graphic tool and to avoid vibration of the haptic device when contact switches occur frequently. Our method achieves the goal well. Fig. 13 shows contact switches when the tool in Fig. 12 slides along one of the top surfaces of the object and then steps across its edge. When the above contact switches occur, the stability of the force and torque (shown in Fig. 13) is ensured. Our method is able to render the sharp feeling of the edge features in non-convex space.

During our experiments involving both convex and non-convex free space, the total time cost of all the computation modules (including collision detection, local search, parallel optimizations and local solution selection) are about 1ms (shown in Fig. 14).

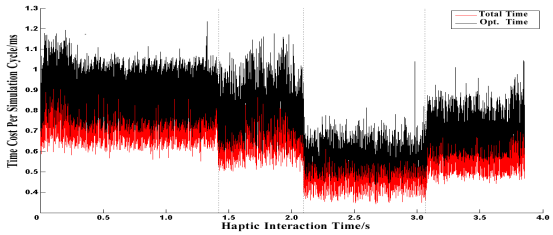


Figure 14 Total time cost and optimization time cost in each simulation loop

The haptic rendering process is illustrated by the attached video. In the video, the red image of the tool indicates the location of the haptic tool, while the gray image indicates the location of the corresponding graphic tool.

C. *Discussion*

The limitation of our method can be summarized below: (1) the region of the object for interaction with the haptic tool cannot be too large with a large number of triangles. (2) When the object includes both convex and non-convex features for the tool to interact, we can only handle a line-shaped tool instead of a complex-shaped tool. This is because a large region of interaction or a complex-shaped tool increases the size of the look-up table for local search and the number of parallel optimization. To construct a large look-up table manually can be too tedious so that automatic means should be investigated. We also need to introduce better computation mechanism to maintain efficiency of the proposed method for large-scale interaction.

VI. CONCLUSIONS AND FUTURE WORKS

In this paper, we have introduced a configuration-based optimization method to compute the configuration of the graphic tool, which is the avatar of the haptic tool, for 6-DOF haptic rendering for fine manipulation.

With this method, visual artifacts are avoided, i.e. no perceptible penetration or separation between the graphic tool and an object. The graphic tool is exactly stopped at the surface of the object.

Furthermore, we have introduced a constraint identification method which not only models contact switches between the graphic tool and both convex and non-convex object features, but also maintains the most relevant local solution instead of a global solution of the optimization model. This local solution is useful to render haptic sensation involving sharp feature and to avoid tunnelling effect.

Finally, feedback force/torque is computed without using virtual coupling. Haptic artifacts are eliminated and stability is maintained. Stable and responsive contact switches are realized because the force/torque signal is continuous to follow the relative motion between the haptic tool and the surface of the object.

Our future work includes increasing the scale of the object mesh from tens to thousands of triangles and extending the method to rendering frictional force.

ACKNOWLEDGMENT

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.

REFERENCES

[1] 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

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

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

[4] David E. Johnson, Peter Willemsen, Elaine Cohen, Six Degree-of-Freedom Haptic Rendering Using Spatialized Normal Cone Search, *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 6, pp. 661-670, Nov./Dec. 2005.

[5] D. Wang , Y. Zhang, *et al.*, Haptic rendering for dental training system, *Science in China Series F: Information Sciences*, Volume 52, Number 3 ,MAR 2009, pp: 529-546.

[6] 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, 2005, p671-683.

[7] Ming C. Lin, Miguel Otaduy, *Haptic rendering: foundations, algorithms, and applications*, A K Peters, Ltd. 2008.

[8] M. Wan and W.A. McNeely, Quasi-Static Approximation for 6 Degrees-of-Freedom Haptic Rendering, *Proc. 14th IEEE Visualization Conf.* (VIS '03), pp. 257-262, 2003.

[9] J. Barbic, D. L. James, Six-DoF Haptic Rendering of Contact between Geometrically Complex Reduced Deformable Models, *IEEE Transactions on Haptics*, Volume: 1, Issue: 1, 2008, page(s): 39-52.

[10] C. Duriez, *et al.,* Realistic haptic rendering of interacting deformable objects in virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 12(1), 2006. 36-47.

[11] M. A. Otaduy, R. Tamstorf, D. Steinemann, M. Gross, Implicit Contact Handling for Deformable Objects ,*Computer Graphics Forum* (Proc. of Eurographics), Volume 28, Number 2 - April 2009.

[12] 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, May/June 2007, pp 458-469.

[13] 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*, Aug. 1995.

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

[15] S. Redon, Fast continuous collision detection and handling for desktop virtual prototyping, *Virtual Reality*, vol. 8, no. 1, pp. 63–70, 2004.

[16] X. Zhang, M. Lee, and Y. J. Kim, Interactive continuous collision detection for non-convex polyhedra, *Vis. Comput.*, vol. 22, no. 9, pp. 749–760, 2006.

[17] M. Tang, Y. J. Kim, D. Manocha: C2A: Controlled conservative advancement for continuous collision detection of polygonal models. *IEEE International Conference on Robotics and Automation (*ICRA*) 2009*: 849-854.

[18] L. Zhang, Efficient Motion Planning using Generalized Penetration Depth Computation, PhD Thesis, Computer Science Department, University of North Carolina at Chapel Hill,2009

[19] D. Baraff, Analytical Methods for Dynamic Simulation of Non-Penetrating Rigid Bodies, *Proc. of ACM SIGGRAPH*, 23 (3): pp. 223-232, 1989.

[20] B. Nguyen and J. Trinkle, Modeling non-convex configuration space using linear complementarity problems, *ICRA 2010*, May 3-8, 2010, Anchorage, Alaska, USA, pp. 2316-2321.

[21] J. Xiao, Q. Luo, and S. You, Haptic Modeling of Contact Formations and Compliant Motion, *ICRA 2003*, Taipei, Taiwan, Sept. 2003.

[22] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha, Fast Distance Queries with Rectangular Swept Sphere Volumes, *ICRA 2000*, pp. 3719-3726, Apr. 2000.

[23] J. Nocedal, S. J. Wright(2006), *Numerical Optimization* (2nd ed.), Berlin, New York: Springer-Verlag