Six Degrees of Freedom Implicit Haptic Rendering

Konstantinos Moustakas

To cite this version:

Konstantinos Moustakas. Six Degrees of Freedom Implicit Haptic Rendering. 10th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI), Sep 2014, Rhodes, Greece. pp.546-555, 10.1007/978-3-662-44654-6_54 . hal-01391357

Six Degrees of Freedom Implicit Haptic Rendering

Konstantinos Moustakas

Electrical and Computer Engineering Department, University of Patras, Rio-Patras, Greece, {moustakas@upatras.gr}

Abstract. This paper introduces a six degrees of freedom haptic rendering scheme based on an implicit support plane mapping representation of the object geometries. The proposed scheme enables, under specific assumptions, the analytical reconstruction of the rigid 3D object's surface, using the equations of the support planes and their respective distance map. As a direct consequence, the problem of calculating the force feedback can be analytically solved using only information about the 3D object's spatial transformation and position of the haptic probe. Several haptic effects are derived by the proposed mesh-free haptic rendering formulation. Experimental evaluation and computational complexity analysis demonstrates that the proposed approach can reduce significantly the computational cost when compared to existing methods.

Keywords: Haptic rendering, implicit representation, six degrees of freedom.

1 Introduction

Human perception combines information of various sensors, including visual, aural, haptic, olfactory, in order to perceive the environment. Virtual reality applications aim to immerse the user into a virtual environment by providing artificial input to its interaction sensors (i.e., eyes, ears, hands). The visual and aural inputs are the most important factors in human-computer interaction (HCI). However, virtual reality applications will remain far from being realistic without providing to the user the sense of touch. The use of haptics augments the standard audiovisual HCI by offering to the user an alternative way of interaction with the virtual environment [1]. However, haptic interaction involves complex and computationally intensive processes, like collision detection or distance calculation [5], that place significant barriers in the generation of accurate and high fidelity force feedback.

Seen from a computational perspective, haptic rendering can be decomposed in two different but heavily interrelated processes, namely collision detection and force calculation. Initially, collisions have to be identified and localized and then the resulting force feedback has to be estimated so as to accurately render the force that will be fed back to the user using specific assumptions on the physical model involved. Concerning haptic rendering research can be divided into three main categories [2]: Machine Haptics, Human Haptics and Computer Haptics [3]. Machine Haptics is related to the design of haptic devices and interfaces, while Human Haptics is devoted to the study of the human perceptual abilities related to the sense of touch. Computer Haptics, or alternatively haptic rendering, studies the artificial generation and rendering of haptic stimuli for the human user [4], [5]. It should be mentioned that the proposed framework takes into account recent research on human haptics, while it provides mathematical tools targeting mainly the area of computer haptics.

The simplest haptic rendering approaches focus on the interaction with the virtual environment using a single point. Many approaches have been proposed so far both for polygonal, non-polygonal models, or even for the artificial generation of surface effects like stiffness, texture or friction, [6]. The assumption, however, of a single interaction point limits the realism of haptic interaction since it is contradictory to the rendering of more complex effects like torque. On contrary, multipoint, or object based haptic rendering approaches use a particular virtual object to interact with the environment and therefore, besides the position of the object, its orientation becomes critical for the rendering of torques. Apart from techniques for polygonal and non-polygonal models [6], voxel based approaches for haptic rendering including volumetric haptic rendering schemes [7] have lately emerged. Additionally, research has also tackled with partial success the problem of haptic rendering of dynamic systems like deformable models and fluids [8].

In general, with the exception of some approaches related to haptic rendering of distance or force fields, one of the biggest bottlenecks of current schemes is that haptic rendering depends on the fast and accurate resolution of collision queries. The proposed approach aims to widen this bottleneck by providing a free-form implicit haptic rendering scheme based on support plane mappings able to provide a six degrees of freedom haptic feedback. In particular, a 3D object is initially modelled using the associated support plane mappings [9]. Then the distance of the object's surface from the support plane is mapped at discrete samples on the plane and stored at a preprocessing step following the same procedure presented in [10]. During run-time and after collision queries are resolved, estimation of the force feedback can be analytically estimated, while several haptic effects, like friction and texture can be easily derived. This results in constant or linear time complexity haptic rendering based only on the 3D transformation of the associated object and the position of the haptic proxy.

2 Support plane mappings

Support planes are a well studied subject of computational geometry and have been employed in algorithms for the separation of convex objects [9, 11, 12]. From a geometrical perspective, a support plane E of a 3D convex object O is a plane such that O lies entirely on its negative halfspace H_E^-. Support planes have become useful in previous algorithms based on the concept of support mappings. A support mapping is a function that maps a vector $\mathbf{v}$ to the vertex of $vert(O)$ of object O that is "most" parallel to $\mathbf{v}$ [9, 13]. As a direct consequence, a support plane can be defined as the plane that passes through $s_O(\mathbf{v})$, the support mapping of $\mathbf{v}$, and is parallel to $\mathbf{v}$.

The importance of support planes is intuitively apparent: they provide an explicit way of deciding whether another object could possibly intersect with the one that the support planes refers to. Based on this simple but important feature of support planes, a slightly more generalized formulation has been derived [14] introducing the concept of support plane mappings for collision detection. The approach described in [14] is used in the proposed framework for collision detection.

After collision is detected, the force feedback provided to the user through the haptic device has to be calculated. In the present framework, force feedback is obtained directly from the model adopted for collision detection, thus handling collision detection and haptic rendering in an integrated way, as described in the sequel.

Let the parametric form of the support plane equation $S_{SP}\left(\eta,\omega\right)$ be:

$$\mathbf{S_{SP}}\left(\eta,\omega\right)=\begin{bmatrix} x_0+\eta u_1+\omega v_1 \\ y_0+\eta u_2+\omega v_2 \\ z_0+\eta u_3+\omega v_3 \end{bmatrix}, \forall \eta,\omega \in \Re \tag{1}$$

where $\mathbf{u}$ and $\mathbf{v}$ constitute an orthonormal basis of the support plane and (x_0, y_0, z_0) its origin.

Assuming now a dense discretization of the η, ω space, we can define a discrete distance map of the support plane SP and the underlying manifold mesh surface S_{mesh}, by calculating the distance of each point of SP from S_{mesh}:

$$D_{SP}\left(\eta,\omega\right)=ICD\left(S_{SP},S_{mesh}\right) \tag{2}$$

where ICD calculates the distance of every point sample (η, ω) of the support plane SP, alongside the normal direction at point (η, ω), from the mesh S_{mesh} and assigns the corresponding values to the distance map $D_{SP}(\eta, \omega)$. The distance map is used in the sequel to analytically estimate the force feedback.

It should be mentioned that the above procedure results in scalar distance maps that accurately encode the surface if and only if there is a one to one mapping of all surface parts with at least one support plane. If such a mapping does not exist, then vectorial distance maps can be used that include information about the distance of all sections of the ray cast in the normal direction of the support plane to the object mesh.

3 Point-based (3DoF) haptic rendering

Referring to Figure 1, let point $\mathbf{H}_p$ be the position of the haptic probe and S_{mesh} represent the local surface of the object.

Let also S_{SP} represent the distance of point $\mathbf{H}_p$ from the support plane, which corresponds to point $\mathbf{P}_M$ on the SP.

If collision is detected, the absolute value of the force fed onto the haptic device is obtained using a spring model as illustrated in Figure 1. In particular:

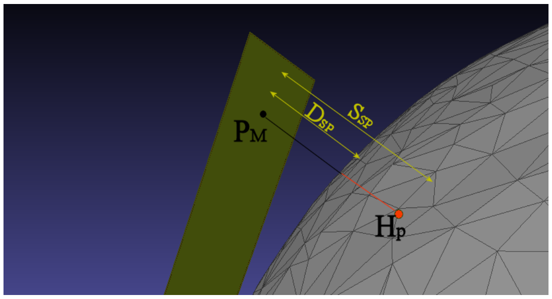

Fig. 1. Distance calculation using distance maps over support planes

$$\|\mathbf{F}\| = k \cdot |S_{SP} - D_{SP}(\mathbf{P_M})| \tag{3}$$

where k is the spring constant. $D_{SP}(\mathbf{P_M})$ is the distance of point $\mathbf{P_M}$ from the mesh and is stored in the distance map of the support plane. Notice that the term $|S_{SP} - D_{SP}(\mathbf{P_M})|$ is an approximation of the actual distance of $\mathbf{H}_p$ from the mesh that becomes more accurate if the support plane surface approximates well the mesh.

The direction of the force should in general be perpendicular to the local area, where collision is detected. An obvious solution to the evaluation of the direction of this force would be to detect the surface element (i.e. triangle) where the collision occurred and to provide the feedback perpendicularly to it. This approach is not only computationally intensive, but also results in non-realistic non-continuous forces at the surface element boundaries. In the present framework the analytical approximation of the mesh surface is used utilizing the already obtained SP approximation and the distance map. Based on this approximation the normal to the object's surface can be approximated rapidly with high accuracy. In particular, assuming that (η, ω) form an orthonormal basis of the plane then if $D_{SP}(\eta, \omega)$ is the scalar function of the distance map on the support plane, as previously described, the surface S_{mesh} of the modelled object can be approximated by equation (4) (Figure 1):

$$\mathbf{S}_{mesh}\left(\eta, \omega\right) = S_{SP}\left(\eta, \omega\right) - D_{SP}\left(\eta, \omega\right) \mathbf{n}_{SP} \tag{4}$$

where S_{SP} is the surface of the support plane, D_{SP} the associated distance map and $\mathbf{n}_{SP}$ its normal vector that can be easily evaluated through $\mathbf{n}_{SP} = \mathbf{u} \times \mathbf{v}$.

Now the calculation of the force feedback demands the evaluation of the normal vector $\mathbf{n}_S$ on the object's surface. That is obtained through equation (5). In the following the brackets (η, ω) will be omitted for the sake of simplicity.

$$\mathbf{n}_S = \frac{\partial \mathbf{S}_{mesh}}{\partial \eta} \times \frac{\partial \mathbf{S}_{mesh}}{\partial \omega} \tag{5}$$

where

$$\frac{\partial \mathbf{S}_{mesh}}{\partial \eta} = \frac{\partial \mathbf{S}_{SP}}{\partial \eta} - \frac{\partial D_{SP}}{\partial \eta}\mathbf{n}_{SP} - D_{SP}\frac{\partial \mathbf{n}_{SP}}{\partial \eta} \tag{6}$$

Since $\mathbf{n}_{SP}$ is constant over SP, equation (6) becomes:

$$\frac{\partial \mathbf{S}_{mesh}}{\partial \eta} = \mathbf{u} - \frac{\partial D_{SP}}{\partial \eta}\mathbf{n}_{SP} \tag{7}$$

A similar formula can be extracted for $\frac{\partial \mathbf{S}_{mesh}}{\partial \omega}$:

$$\frac{\partial \mathbf{S}_{mesh}}{\partial \omega} = \mathbf{v} - \frac{\partial D_{SP}}{\partial \omega}\mathbf{n}_{SP} \tag{8}$$

All above terms can be computed analytically, except from $\frac{\partial D_{SP}}{\partial \eta}$ and $\frac{\partial D_{SP}}{\partial \omega}$ that are computed numerically.

Substituting now equations (4), (6), (7), (8) in equation (5) the normal direction $\mathbf{n}_S$ can be obtained.

Since, the direction of the normal along the surface of the modelled object is obtained using equation (5), the resulting force feedback is calculated through:

$$\mathbf{F}_h = k\left|S_{SP} - D_{SP}(\mathbf{P}_M)\right| \frac{\mathbf{n}_S}{\|\mathbf{n}_S\|} \tag{9}$$

4 6DoF haptic rendering

Let us now assume that the haptic interaction point is actually a haptic interaction object that is also modelled using support plane mappings and distance maps.

Referring to Figure 2 that for simplicity depicts the 2D case, let M_1 and M_2 be the mesh areas of two different objects possibly involved in collision and S_1 and S_2 their respective support planes. Based on the approach presented in [14] it can be decided whether M_1 and M_2 are probably involved in collisions using their SPMs S_1 and S_2. Now the question is: Is it possible to identify the collisions and calculate 6DoF force feedback without entering the computationally intensive narrow-phase [14] of the collision detection algorithm?

This question reduces to the problem of calculating the impact volume (3D case) or impact surface (2D case) S as depicted in Figure 2. Let us consider for sake of simplicity and without loss of generality the 2D case. If $\mathbf{d}$ is a material density function then the mass corresponding to the collision area S can be described by the following formula.

$$V = \int\limits_{S}\int \mathbf{d}dS \tag{10}$$

Now referring again to Figure 2 let W be a sampling point of the support plane S_2 and W_2 the point of mesh M_2 that can be trivially reconstructed by projecting W along side the normal direction n_2 as far as the distance map

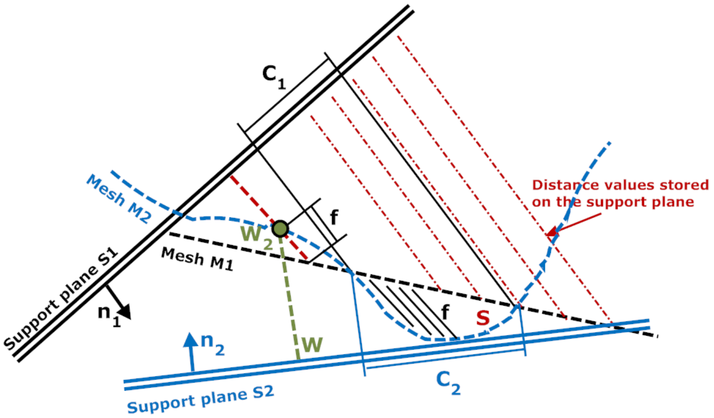

Fig. 2. Distance calculation using distance maps over support planes

$D_{SP}(W)$ dictates. Now, W_2 can be also trivially projected on the support plane S_1 and retrieve the corresponding distance value $D_1(M_1)$ of S_1 from the mesh M_1. Now, omitting the notation (η, ω) in the following for simplicity and assuming the function f is defined as follows:

$$f = D_1\left(W_2\right) - D_1\left(M_1\right) \tag{11}$$

where $D_1\left(W_2\right)$ is the distance of point W_2 from S_1 and $D_1\left(M_1\right)$ the corresponding distance of S_1 from mesh M_1. Now, collision can be reported only for cases where function f is positive.

Let now C_1 and C_2 (Figure 2) denote the projection of the colliding volume on the support planes S_1 and S_2 respectively. It is obvious that C_1 is the support set of positive values of function f. Then, equation (10) can be transformed as follows:

$$V = \int\limits_{t_1 \in C_1} \mathbf{d} \cdot f \, dt_1 = \int\limits_{t_2 \in C_2} \mathbf{d} \cdot f \cdot \cos\left(\mathbf{n}_1 \mathbf{n}_2\right) \, dt_2 \tag{12}$$

A similar analysis can be followed for the three dimensions and the corresponding formula is:

$$V = \int\int_{\eta, \omega \in C_2} \mathbf{d} \cdot f \cdot \cos\left(\mathbf{n}_1 \mathbf{n}_2\right) d\eta d\omega \tag{13}$$

Now since the volume of the penetrating object part and its centre of inertia is known, 6DoF haptic feedback can be easily estimated. In the context of the proposed framework a single force is estimated for the penetrating part of the haptic interaction object that is applied on the gravity centre of the colliding volume. Other approaches for approximate 6DoF haptic rendering like the one presented in [15] can be also implemented.

5 Haptic effects

The analytical estimation of the force feedback based only on the object 3D transformation, the probe position and the distance maps, provides the opportunity to develop closed form solutions for the rendering of physics-based or symbolic force effects; the following sections indicatively describe some of them.

By applying a local smoothing operation to the distance map, the resulting force feedback is smooth in the areas around the edges, without being overrounded as is the case with the force shading method [16]. A typical example of distance map preprocessing so as to achieve force smoothing using a Gaussian kernel is given by the following equation:

$$D'_{SP}\left(\eta,\omega\right) = D_{SP}\left(\eta,\omega\right) * G_{\sigma}\left(\eta,\omega\right) \tag{14}$$

where G_{σ} is a 2D Gaussian kernel and "$*$" denotes convolution. It is evident that different smoothing operators can be easily applied. A very useful operator that can be implemented is the force smoothing only in areas that are not smooth due to the finite tessellation (sampling) and not in object and surface boundaries, following the popular in computer graphics "crease angle" concept. A haptic "crease angle" rendering can be easily performed by applying anisotropic diffusion or even an edge-preserving smoothing operator on the distance map.

Similarly using the proposed framework for haptic rendering, haptic texture can be also simulated easily by applying appropriate transformations on the acquired distance map. An example for simulating surface roughness is provided below, where Gaussian noise is added on the distance map. No computational cost is added, since the procedures for calculating the force direction are not altered due to the existence of haptic texture. The only difference lies in the evaluation of the magnitude of $\mathbf{F}_{texture}$, which now yields from:

$$\mathbf{F}_{texture} = k\left|S_{SP} - (D_{SP}(\mathbf{P_M}) + n_g)\right| \frac{\mathbf{n}_S}{\|\mathbf{n}_S\|} \tag{15}$$

where n_g denotes the gaussian noise.

6 Complexity and experimental results

In the following an analysis of the computational complexity of the proposed scheme in comparison to the typical state-of-the-art mesh-based haptic rendering scheme is discussed.

Moreover, even if an experimental analysis of the proposed support plane mapping based haptic rendering approach, in terms of timings for simulation benchmarks would not be fair for the state-of-the-art approaches, since it would encode the superiority of SPM based collision detection and would not directly highlight the proposed haptic rendering approach, two experiments are presented where the proposed haptic rendering scheme is compared to the state-of-the-art mesh-based haptic rendering in terms of timings in computationally intensive surface sliding experiments.

After collision is reported, a typical force feedback calculation scheme would need to identify the colliding triangle of the involved 3D object in $O(n)$ time, where n is the number of triangles, or in $O(logn)$ time if bounding volume hierarchies are used. Then the force can be calculated in constant $O(1)$ time. In order to avoid force discontinuities, for example force shading, and if there is no adjacency information then the local neighborhood of the colliding triangle can be found again in $O(n)$ time, where n is the number of triangles, or in $O(logn)$ time if bounding volume hierarchies are used. Finally, the mesh-based haptic rendering scheme has no additional memory requirements per se.

Process	Mesh-based	Free-form
Force	$O(n)$ or $O(logn)$	$O(1)$
6DoF force	$O(mn)$ or $O(mlogn)$	$O(m)$
Smoothing	$O(n)$ or $O(logn)$	$O(1)$
Memory	-	$O(m \cdot s)$

Table 1. Computational complexity comparison

On the other hand, concerning the proposed free-form implicit haptic rendering scheme, after a collision is detected, the resulting force feedback can be calculated in constant time $O(1)$ using equation (9). In order to avoid depth discontinuities the distance map can be smoothed, in an image processing sense, at a preprocessing phase. Even if this step is performed during run-time it would take $O(k)$ time, where k is the local smoothing region or the filtering kernel window. Concerning the 6DoF case, it can be considered requiring to execute m times the 3DoF procedure, where m is the number of support plane sampling points processed. On the other hand the proposed scheme has $O(m \cdot s)$ memory requirements, where m is the number of support planes and s the number of samples per support plane. Taking now into account that the more support planes are used the smaller their size and the less samples are necessary for a specific sampling density we can safely assume that the memory requirements are linear to the total number of samples that depends on the sampling density used.

Table 1 summarizes the computational complexity analysis of the proposed free-form haptic rendering scheme, when compared to the mesh-based approach. Concerning the quantitative results, interaction with two objects was considered, namely the Galeon and Teeth models of 4698 and 23000 triangles respectively. The objects are illustrated in Figure 3 .

Moreover, CHAI-3D was used for interfacing with the haptic devices [17]. Moreover, the force estimation algorithms were applied on a predefined trajectory of the haptic probe, so as to assure fair comparison. In particular, initially the trajectory of the haptic probe in the 3D space has been recorded while being in sliding motion over the objects' surface. Then this trajectory has been used as input for both algorithms so as to extract the timings mentioned below. Table 2 presents the mean timings and their standard deviation of the force estima-

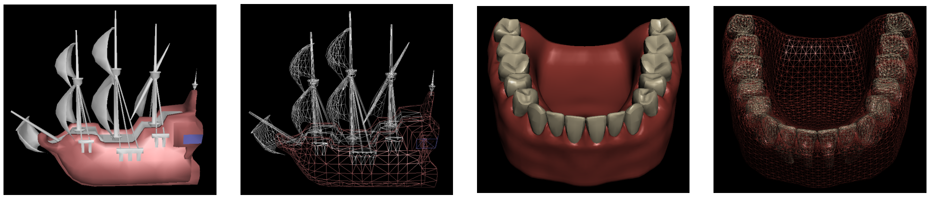

Fig. 3. Galeon model, 4698 triangles. Teeth model, 23000 triangles

tion throughout the simulation using the mesh-based,and the proposed free-form haptic rendering scheme for the case of the Galeon and the Teeth models.

	Galeon		Teeth	
Process	Mean time (ms)	σ	Mean time (ms)	σ
Mesh-based	1.2	0.22	4.2	0.82
Free-form	0.028	0.006	0.036	0.005

Table 2. Galeon and Teeth models: Interaction timings

It should be emphasized that the above timings need to be taken into account under the exact experimental setting. In particular, concerning the proposed approach 1000 support planes were used for the case of the Galeon and 1500 for the case of the Teeth model. Distances are estimated for all support planes and forces are calculated for the closer one. This procedure, could be optimized by partitioning the space in a preprocessing step and knowing beforehand to which support plane, each point in space "belongs to", thus reducing the search from $O(n)$ to $O(logn)$. Moreover, concerning the mesh-based approach force shading has been also implemented.

7 Conclusions

The proposed framework extends the support plane mapping concept and the corresponding collision detection scheme to direct free-form implicit haptic rendering. An analytical scheme to calculate both 3DoF and 6DoF force feedback is proposed. Its most significant property is that there is no need to enter the narrow-phase of the collision detection pipeline so as to identify the colliding triangles and subsequently estimate the reaction forces, on contrary it requires only the distance maps stored on the support planes. It is evident that the proposed scheme reduces significantly the computational cost in the performed simulations. This significant gain comes at an expense of two limitations. Firstly, special care has to be taken at the preprocessing step so that the models are well approximated using the support planes and the distance maps. For example if the objects demonstrate large complex concavities the use of vectorial distance maps is inevitable. Secondly, the proposed scheme cannot be, in its current form, directly applied to deformable models. An extension to piecewise or free-form deformable models, where deformations can be analytically expressed seems possible and remains a direction for future work.

References

1. Burdea, G., Coiffet, P.: Virtual Reality Technology. Wiley-IEEE Press, 2nd edition (2003)
2. Lin, M., Otaduy, M.: Haptic rendering: Foundations, algorithms and applications. (2008) A.K.Peters publishing
3. Srinivasan, M., Basdogan, C.: Haptics in virtual environments: Taxonomy, research status and challenges. Computers and Graphics (1997) 393–404
4. Moustakas, K., Tzovaras, D., Strintzis, M.: Sq-map: Efficient layered collision detection and haptic rendering. IEEE Transactions on Visualization and Computer Graphics **13** (2007) 80–93
5. Moustakas, K., Nikolakis, G., Kostopoulos, K., Tzovaras, D., Strintzis, M.: Haptic rendering of visual data for the visually impaired. IEEE Multimedia **14** (2007) 62–72
6. Laycock, S., Day, A.: A survey of haptic rendering techniques. Computer Graphics Forum **26** (2007) 50–65
7. Palmerius, K., Cooper, M., Ynnerman, A.: Haptic rendering of dynamic volumetric data. IEEE Transactions on Visualization and Computer Graphics **14** (2008) 263–276
8. Barbic, J., James, D.: Six-dof haptic rendering of contact between geometrically complex reduced deformable models: Haptic demo. In: In Proc. of Eurohaptics. (2009) 393–394
9. van den Bergen, G.: Collision Detection in Interactive 3D Environments. The Morgan Kaufmann Series in Interactive 3D Technology. Morgan Kaufmann (2003)
10. Moustakas, K.: Free-form implicit haptic rendering. In: Joint Virtual Reality Conference, JVRC. (March, 2005) 73–76
11. Dobkin, D.P., Kirkpatrick, D.G.: A linear algorithm for determining the separation of convex polyhedra. Journal of Algorithms **6** (1985) 381 – 392
12. Chung, K., Wang, W.: Quick Collision Detection of Polytopes in Virtual Environments. In: ACM Symposium on Virtual Reality Software and Technology 1996. (1996) 1–4
13. Ericson, C.: Real-Time Collision Detection. The Morgan Kaufmann Series in Interactive 3D Technology. Morgan Kaufmann (2005)
14. Vogiannou, A., Moustakas, K., Tzovaras, D., Strintzis, M.: Enhancing bounding volumes using support plane mappings for collision detection. Computer Graphics Forum **29** (2010) 1595–1604
15. McNeely, W., Puterbaugh, K., Troy, J.: Six degree-of-freedom haptic rendering using voxel sampling. In: Computer graphics and interactive techniques. (1999) 401–408
16. Ruspini, D., Kolarov, K., Khatib, O.: The haptic display of complex graphical environments. In: Computer Graphics (SIGGRAPH 97 Conference Proceedings). (1997) 345–352
17. Conti, F., Barbagli, K., Morris, D., Sewell, C.: Chai 3d: An open-source library for the rapid development of haptic scenes. In: IEEE World Haptics, Pisa, Italy. (March, 2005)