Adaptive 6-DoF Haptic Contact Stiffness Using the Gauss Map

Hongyi Xu, *Member, IEEE,* Jernej Barbič, *Member, IEEE,*

Abstract—The penalty method is a popular approach to resolving contact in haptic rendering. In simulations involving complex distributed contact, there are, however, many simultaneous individual contacts. These contacts have normals pointing in several directions, many of which may be parallel, causing the stiffness to effectively accumulate in a temporally highly-varying and unpredictable way. Consequently, penalty-based simulation suffers from stability problems. Previous methods tackled this problem using implicit integration, or by scaling the stiffness down globally by the number of contacts. Although this provides some control over the net stiffness, it leads to large penetrations, as small contacts are effectively ignored when compared to larger contacts. We propose an adaptive stiffness method that employs the Gauss map of contact normals to ensure a spatially uniform and controllable stiffness in all contact directions. Combined with virtual coupling saturation, penetration can be kept shallow and simulation remains stable, even for complex geometry in distributed contact. Our method is fast and can be applied to any penalty-based formulation between rigid objects. While used primarily for rigid objects, we also apply our method to reduced deformable objects. We demonstrate our approach on several challenging 6-DoF haptic rendering scenarios, such as car engine and landing gear virtual assembly.

Index Terms—haptics, 6-DoF, penalty contact, stiffness, virtual coupling saturation

1 Introduction

Simulating distributed contact between geometrically complex objects is a challenging task for haptic rendering, both from the point of stability and the speed of computation. Because they are simple and fast, penalty-based methods are often employed in haptic simulations of real-world tasks such as engineering design assessment and maintenance training [1]. For large-scale complex geometry, *distributed* contact is often formed, where contact sites are many and have varying sizes both in space and time [2]. In such contact scenarios, penalty methods are less reliable due to the unpredictable stiffness variation and accumulation.

To improve haptic stability, stiffness scaling is often applied to prevent excessively high stiffness (Figure 2). A simple stiffness scaling method was proposed in [1]. It works by scaling the stiffness, globally at all contacts, with a factor proportional to the inverse of the number of contacts, so that the total stiffness is not too high for stable simulation. Such an approach makes the penalty forces temporally more smooth and improves haptic stability. However, because all contacts are scaled with the same value, such scaling will cause contacts with small contact areas to be de-emphasized/ignored relative to contacts with large contact areas. In real-world scenarios, however, even a single isolated contact can and should drastically alter the contact simulation (see Figure 3). The global stiffness scaling of [1] causes such small, but essential contacts to be ignored, resulting in unwanted penetration and invalid virtual assembly paths. In this paper, we address this problem by proposing a better stiffness scaling method which not only produces temporally coherent penalty forces, but also assures uniform, stiff and stable contact stiffness in all directions, regardless of the specific number of contact sites, their size, sampling density and normal orientations. Our method properly scales both small and large contacts so that they contribute equally to the overall contact simulation. It works by assigning spatially-varying scaling weights to each contact point, based on the force and torque directions analyzed under the Gauss map. Our method makes it possible to employ high contact stiffness, and keeps the penetration depths under control. We demonstrate the effectiveness of our method on both 3-DoF and 6-DoF haptic simulations, involving challenging 6-DoF haptic simulations of complex real-world airplane and car geometry. We demonstrate that such penalty simulations are not possible without our method (Figure 1(a), Figure 11(a)). Our method can be added to any penalty-based simulator, and applies both to admittance and impedance haptic rendering modes. Stability and avoidance of penetration can be further improved using virtual coupling saturation. Although previous work already employed saturation, saturation becomes a much more powerful and reliable tool when using our new adaptive stiffness scaling (Figure 10).

• *Both authors are with the Department of Computer Science, University of Southern California, Los Angeles, CA, 90089.*
E-mail: hongyixu@usc.edu, jnb@usc.edu.

2 Related Work

Haptic rendering has been an active area of research over the last decade [3]. It requires high-update rates, which are challenging to achieve in complex collision scenarios. One of the oldest and still most efficient methods in terms of computation time is the Voxmap PointShell (VPS) method [1], [4], [5]. It regularly samples the surface of the object with a collection of points with normals. The environment is modeled as a map of voxels (*voxmap*) where each voxel contains a fixed-width integer value denoting whether the object is outside or inside the object, and the distance to contact. For collision detection, the points are queried against the voxmap, and then penalty forces are computed when collisions are detected. The VPS method can typically provide reliable 1000 Hz haptic refresh rate without resorting to multi-rate simulation [6]–[8]. VPS has been later improved by many authors [9], [10]. Barbič and James [11] extended the method to deformable objects and improved contact stability using distance

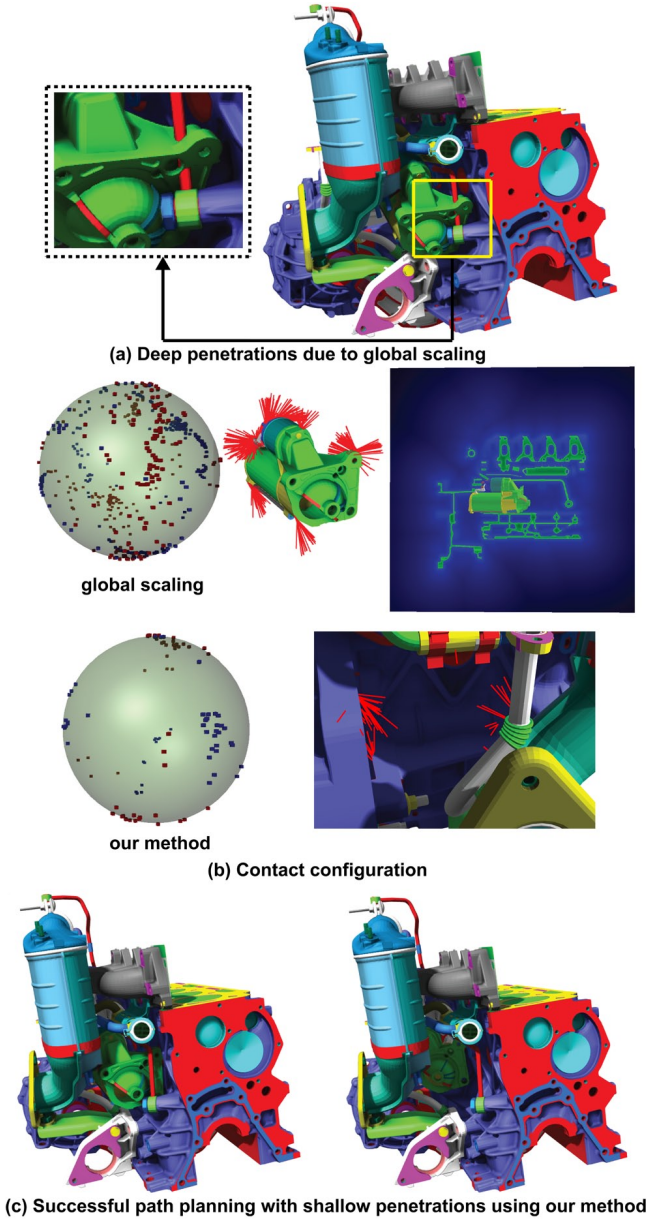


Fig. 1: Haptic assembly of the car engine model. *A starter motor (major car manufacturer; real geometry) is inserted into its proper place in the engine compartment. (a) With previous methods (global scaling [1]), pop-throughs occur on the thin tubes (shown in yellow rectangle in the top-left figure). (b) Multiple contacts are formed, with various contact normal directions, when manipulating the starter motor inside the car engine compartment. Under global scaling, there are* 430 *contacts (shown on the car starter motor), whereas our method produces* 55 *contacts. Our method produces fewer contacts than global scaling because it does not excessively decrease stiffness in all contact directions. Gauss maps are displayed on a unit sphere. A vector from the sphere center to each red and blue dot represents a single contact force and torque direction, respectively. All the thin and small features are represented with our high-resolution distance field (*$1024 \times 1024 \times 1024$*; one voxel corresponds to approximately* $0.9mm$*). One slice of the signed distance field is shown (blue indicates outside, and green is inside). (c) Our method keeps the penetration depth shallow and finds a real, non-penetrating path to manipulate the starter motor through the narrow passage into its place. As such, it increases the quality and accuracy of the virtual assembly experience.*

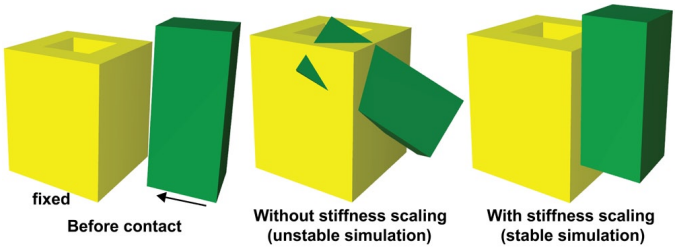
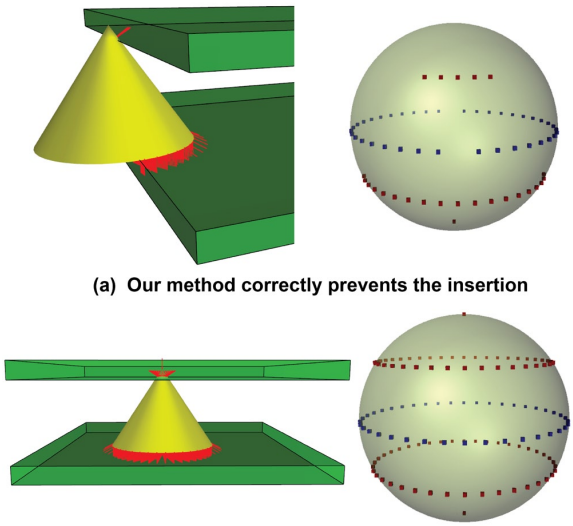


Fig. 2: The need for contact stiffness scaling: *A large contact area between the hole and the peg is formed (*178 *contacts). If stiffness is unscaled, high contact stiffness occurs due to the large number of contacts, and the haptic simulation becomes unstable, despite using pseudo-admittance. Under otherwise identical simulation conditions, stiffness scaling greatly improves simulation stability, and permits the contact stiffness to be* $8\times$ *higher. In this example, we use global stiffness scaling. Because all the contacts have equal normals in this example, our method gives the same result as global stiffness scaling.*

fields. The VPS method resolves distributed contacts very efficiently, albeit at reduced collision accuracy. Unlike penalty-based methods, constraint-based methods resolve contact via modeling contacts as unilateral constraints [12]–[16]. These methods can avoid penetrations and pop-throughs, with possibility to correctly compute contact friction forces [17], albeit at the cost of much more computation. In this paper, we follow the penalty method of [11] for fast collision detection and resolution.

Penalty methods are prone to numerical instabilities due to the high and unpredictable stiffness variations encountered at any timestep. This is especially pronounced with haptic rendering of complex and conforming distributed contact under a fixed numerical timestep. Contact clustering alleviates the instability problem by grouping contacts depending on the Euclidean distance between contact points [18]–[21]. However, the individual contact points can be very close to each other but have significantly different contact directions. Similar to global scaling [1], the under-sampled contact directions can have undesirable low contact stiffness when they are clustered and averaged with dominant contact points with different contact directions. Clustering also requires a careful selection of the number of clusters or the clustering threshold, typically performed using trial and error. Our method improves stability and ensures that all contacts participate in the simulation, even if they are represented only by a small number of points, or even by a single point. Our approach of mapping the collision normals onto a unit sphere is similar to the normal cone methods [22], [23]. These methods, however, accelerate collision detection, whereas we use the normals to improve the resolution of contact. Virtual coupling is often used for haptic rendering, either as a mass-spring-damper [1] or as a quasi-static spring [4]. It is often combined with saturation for large displacements to avoid deep penetrations [4], [24]. When integrated with the VPS global scaling method, the virtual coupling saturation often fails to prevent deep penetration due to improperly scaled stiffness. In contrast, our spatially-varying adaptive stiffness scaling greatly improves the ability of virtual coupling saturation to prevent unwanted penetrations. Employing contact gradients and implicit integration is one possible way to improve the passivity of haptic penalty-based simulations [2], [25], [26]. Our adaptive scaling is orthogonal to the specific integration method applied, whether



(a) Our method correctly prevents the insertion

(b) Global scaling incorrectly permits the insertion

Fig. 3: 6-DoF haptic rendering of inserting a cone into a gap between two walls. *The gap (rendered transparently) is smaller than the height of the cone. The contact normals are shown as red arrows. Note that multiple contact points have the same force or torque directions and thus a dot on the Gauss map sphere may represent multiple contact points, such as the red dot on the south pole of the unit sphere. (a) Large area of contact points are formed at the bottom of the cone while only a few contacts occur at the top. Our method properly scales the stiffness at the top contacts, and therefore the cone cannot slide inside the gap. (b) Using global scaling method, the cone can easily incorrectly slide into the gap. Much deeper penetrations occur at the top of the cone.*

explicit or implicit. We demonstrate that our method combined with implicit integration outperforms implicit integration with previous scaling methods.

3 Adaptive Contact Stiffness

In our method, we follow the penalty-based method of [11], which computes contact forces and torques between a pointshell object and a distance field object. The sample points are organized in the nested bounding sphere hierarchy. Our collision detection algorithm traverses the hierarchy in breadth-first order. We do not use interruptible collision detection and always complete all levels. Once a point P is detected to be in collision, we compute the contact force F and torque τ as

$$F = -kdN, \quad \tau = r \times F, \tag{1}$$

where $d < 0$ is the penetration depth, k is the contact stiffness, $r \in \mathbb{R}^3$ is the torque handle and $N \in \mathbb{R}^3$ is the point's inward contact normal. We note that this model avoids distance field gradient discontinuities by using the point normal as the direction of the penalty force.

Whenever many point contacts occur simultaneously, the net stiffness may become large and cause haptic instabilities. Previous work [1] copes with this problem by essentially replacing the vector sum of all contact forces and torques by their average. This is achieved by dividing the force and torque sums by the number of contact points n (See Figure 2), essentially globally dividing the contact stiffness by n. The force discontinuities introduced from such stiffness scaling are mitigated by deferring the scaling process until $n \geq n_0$ ($n_0 = 10$ in [1]), i.e.,

$$k_{\text{effective}} = \begin{cases} \frac{k}{n/n_0} & \text{if } n \geq n_0 \\ k & \text{otherwise.} \end{cases} \tag{2}$$

Such stiffness scaling is very useful when the contact points form a single contact area, sampled with many contact points, and producing penalty forces in identical or nearly-identical directions, such as a box colliding with a wall (Figure 2). However, the scaling operation is *global*, disregarding the contact distribution, point sampling and contact normal directions. When there are many distinct contact normal directions, as typical with complex geometry in tight contact in virtual assembly scenarios, the directions that are represented with fewer point samples will be overly softened. This will cause undesirable large penetrations. Figure 3 illustrates this process. It depicts a cone inserted into the gap between two parallel walls. Many contact points are formed at the cone's bottom while there are only a few contact points at the top. The global stiffness scaling method [1] cannot produce sufficiently large penalty forces to prevent large penetrations at the top of the cone. Another related example is a uniformly-sampled cylinder peg sliding into a cylindrical hole. In this example, there are many points are in contact, with contact forces pointing in a radial direction. Once the peg hits the bottom of the hole, new contacts appear at the bottom of the peg. With global scaling, the stiffness of all the contacts is scaled down uniformly. Because there are many more contacts on the sides of the peg, the penalty force at the bottom is relatively weak and will permit the peg to penetrate deeper into hole. An optimal scaling solution should keep the simulation stiff in all the contact directions present in the current contact configuration, while still ensuring stable numerical integration.

3.1 Adaptive 3-DoF Stiffness

We first give our spatially-varying adaptive stable stiffness scaling for 3-DoF haptic simulations, i.e., simulations where the object is only allowed to rigidly translate, but change of orientation (rotations) are not allowed. We will later extend our technique to 6-DoFs. Suppose there are currently n contacts in our simulation. Then, each contact contributes to the net stiffness in the current contact configuration. Note that, for each single contact, the

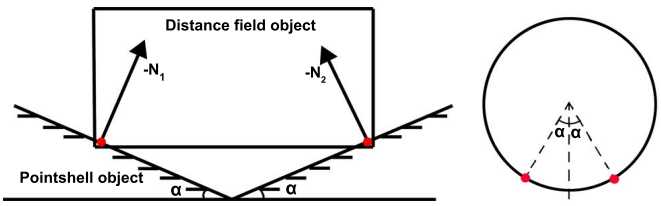


Fig. 4: Uniform stiffness rescaling does not work. *A box is in contact with two inclined planes. There are 2 contacts. When $\alpha < 45°$, moving the object up-down is stiffer than moving it left-right, and the specific amount depends on the angle α. Such a contact sensation is impossible to achieve via the naive method that simply filters the eigenvalues of the contact gradient. Our method, in turn, produces the correct result.*

contact normal N is the direction in which the object motion is met with the strongest resistance, when considering this contact in isolation. With many contact points, these effects add up, as follows. Suppose the object position is $x \in \mathbb{R}^3$. The gradient of the contact force of contact i is $dF_i/dx = -kN_iN_i^T$ [26]. Therefore, the net change of the contact force under a small displacement δx is

$$\delta F = \left(\sum_{i=1}^{n} -kN_iN_i^T\right)\delta x = A\delta x. \tag{3}$$

The symmetric matrix $A \in \mathbb{R}^{3\times3}$ is the contact gradient. The eigenvectors of this matrix give the three orthogonal directions of the maximum, medium and smallest net stiffness. In other words, the effect of having n contacts is that the stiffness has now become anisotropic. Our first attempt to make stiffness uniform was to modify the three eigenvalues of A so that they were all equal, i.e., effectively converting A into a form $A = kI$, where I is the 3×3 identity matrix. This approach, however, does not work in practice. Figure 4 gives a 2D illustration where

$$A = \begin{bmatrix} 2k\sin^2\alpha & 0 \\ 0 & 2k\cos^2\alpha \end{bmatrix}. \tag{4}$$

When $\alpha < 45°$, it should be softer to move the distance field object in a horizontal as opposed to vertical direction. However, filtering the eigenvalues of the A matrix will produce isotropic stiffness in horizontal and vertical directions, which is unrealistic. Another problem with such a naive scaling is that it requires a threshold when α is approaching 0, so that the stiffness in the horizontal direction can vanish when $\alpha = 0$. This produces a force discontinuity around the threshold.

These observations lead us to seek a method where we do not simply modify the contact gradient to make all directions equally stiff. Instead, we observe that the spatial contact stiffness is determined by the directions of the normals present in the current contact configuration, with the rule that repeated identical normals, or nearly identical normals, should behave

$$\mathcal{X} : N \in \mathbb{R}^3 \mapsto S^2, \tag{5}$$

which maps each unit contact normal $N = (x,y,z)$ onto point (x,y,z) on the unit sphere S^2 in three dimensions (Figure 3). like a single normal. Therefore, we design a method that produces equal stiffness in all the directions present in the current contact configuration, instead of isotropically in all directions. We achieve this by analyzing the Gauss map $\mathcal{X}$, The Gauss map is a great tool to reveal the "normal distribution": areas of S^2 that have many points in proximity (or on top of each other) correspond to very stiff directions due to accumulated stiffness, typically because of many parallel contact normals on a large contact area. Such areas need to be de-emphasized by weighting those contacts with a small weight. Conversely, isolated points on the Gauss map, are lonely, isolated contacts, corresponding to a single point in contact. The stiffness of these contacts should not be scaled down. Under global scaling, such points are essentially ignored

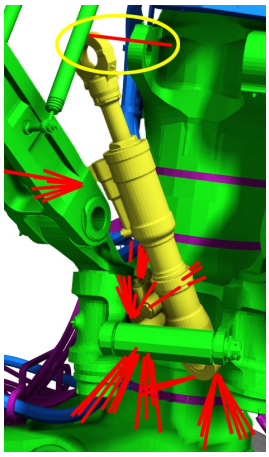


Fig. 5: *Complex contact distribution.*

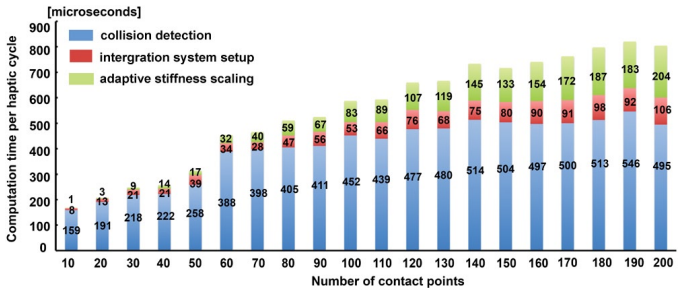


Fig. 6: Stiffness scaling on regions of unit sphere. *In the car engine example, we measured the computation time of collision detection, linear system setup for implicit integration, and our adaptive stiffness scaling per haptic cycle, under varying number of contact points. Here we use implicit dynamic virtual coupling; therefore, the linear system setup includes computing the contact and damping forces and torques, and also their gradients with respect to displacements, rotations and velocities. The linear system size is* 6×6 *and does not depend on number of contact points [2]; therefore, the computation time for the linear system solve is constant at around* $20\sim30$ *microseconds. Collision detection dominates the computation time when* $n < 200$.

and the contact is missed. In practice, such isolated contacts can be very important and should not be missed. This is very pronounced in our car engine demo, where such isolated contacts are in abundance and global scaling produces incorrect virtual assembly paths that violate such contacts.

To achieve equal stiffness in all contact directions, we scale the contact stiffness k at each contact with a weight that varies at each contact. Essentially, we compute weights so that when one looks at the region sampled by the normals on the Gauss sphere, the total net weighted accumulated stiffness everywhere is approximately k. Our weights are therefore inversely proportional to the local "density" of contacts on the Gauss sphere. We compute the weights as follows. For each contact point i, we form a "hat" function

$$\phi(d_{ij}) = \begin{cases} 1 - \frac{d_{ij}^2}{\varepsilon^2} & \text{if } d_{ij} < \varepsilon, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

$$\phi(d_i) = \sum_{j=1}^{n} \phi(d_{ij}) \geq 1, \qquad d_{ij} = \|N_j - N_i\|, \tag{7}$$

$$w_i = \frac{\phi(d_i)}{\phi_{max}}, \qquad \phi_{max} = \max_{1\leq i\leq n} \phi(d_i). \tag{8}$$

We evaluate the total contact forces and torques as

$$F_i = -w_i k d_i N_i, \quad F = \sum_{i=1}^{n} F_i, \quad \tau = \sum_{i=1}^{n} r_i \times F_i. \tag{9}$$

Unlike [1], we do not defer the scaling process until a certain number of contact points n_0 is reached. We set the parameter ε to 0.01 in all our examples. In the special case where all the contact normals point in the same direction (i.e., flat contact with a wall), $\phi(d_i)$ will be equal to n for each contact point and therefore our scaling is the same as the global scaling of [1], if deferring process is also applied. However, when the contact normals are distributed irregularly, which is quite typical in distributed contact scenarios, our method assures that no contact direction is scaled down too much, preventing unwanted large penetrations. We note that we can also use the geodesic distance between points for d_i, but

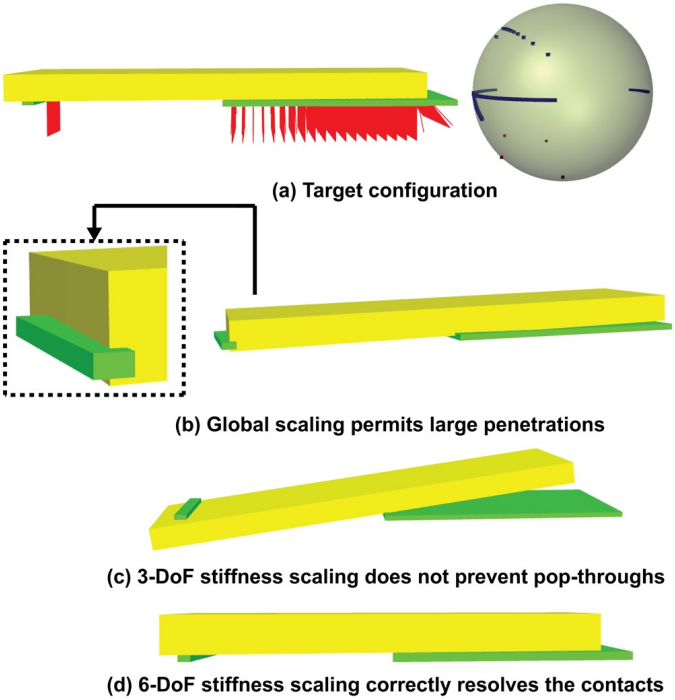


Fig. 7: 6-DoF stiffness scaling is necessary to avoid penetrations. *(a) A yellow box is placed on two thin green planes. Two contact areas of different sizes are formed ($n = 1{,}014$), and thin features are involved. (b) Using the global scaling method, deep penetrations occur. (c) 3-DoF stiffness scaling behaves similarly as the global scaling method, since most of the contact normals are pointing in the same direction. Pop-through occurs on the wall with fewer contact points. (d) Our 6-DoF adaptive stiffness resolves the problem by also accounting for the differences of the contact torque directions.*

Euclidean distance already works well in practice. Our method is simple to implement and its computational overhead is negligible compared to collision detection when $n < 100$ (Figure 6). Although the computation complexity of scaling is $O(n^2)$, collision detection still dominates the computation even when $n = 200$ (25.6% computation time for the scaling). Note that using adaptive stiffness scaling, stiffness is maintained reasonably high in all the contact directions. It will thus significantly reduce the number of contact points compared to using global scaling (Figure 1 (b)). In our examples, we typically have $n < 150$, and therefore our adaptive scaling method does not impose a large overhead.

3.2 Adaptive 6-DoF Stiffness

We now extend our stiffness scaling to 6-DoF haptic simulations. For 6-DoF haptic rendering, we need to consider the penetrations caused not only by improperly scaled penalty forces, but also by the torques. We define the normalized torque directions T as

$$T = \frac{r \times N}{\|r \times N\|_2}. \tag{10}$$

We extend the Gauss map to also map unit torque directions T, in addition to normals:

$$\mathcal{X} : (N, T) \in \mathbb{R}^6 \mapsto S^2 \times S^2. \tag{11}$$

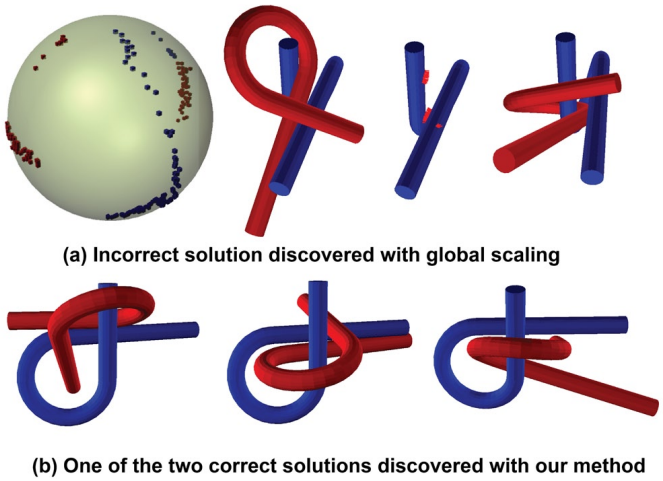


Fig. 8: Haptic path planning ("alpha puzzle") [28], [29]. *(a) Contacts are formed on the two sides on the passage with opposite contact normals ($n = 118$). With global scaling, stiffness is scaled down by $11.8\times$ at all the contact points. The red alpha-shape tube can be inserted into the narrow gap simply due to lower stiffness, incorrectly "solving" the puzzle ("cheating"). (b) With our scaling, the demo is completely stable and unbreakable: our method does not permit "cheating" by exploiting any limitation of the penalty-based model. Instead, our method only permits the two correct collision-free solutions to resolve the puzzle.*

The distance d_i definition is now extended to be [27]

$$\phi(d_{ij}) = \max(1 - \frac{\|N_j - N_i\|^2}{\varepsilon^2}, 0) + \max(1 - \frac{\|T_j - T_i\|^2}{\varepsilon^2}, 0) \tag{12}$$

Another possible alternative would be to use unnormalized torque directions, which also incorporate the handle lengths, as

$$\hat{T} = r \times \frac{F}{\|F\|_2}. \tag{13}$$

However, for large scenes with long torque handles, the force direction differences may then become overshadowed by the large torque directional differences, producing incorrect stiffness scaling. We therefore prefer unit torque directions T since they avoid the need of introducing an additional parameter to weight the contributions of the forces and torques in Equation 12. Given d_i, we use Equation 8 to compute the weight of each contact. We note that we do **not** scale forces and torques separately. Instead, we only scale the stiffness of each contact, which affects both the force and torque at that contact simultaneously. Such stiffness scaling makes forces and torques consistent with each other, which is good for stability.

Adaptive stiffness that incorporates the torque directions T is essential when the contacts have similar (or identical) normals, but there are large differences in torque directions. In such situations, the Gauss map of the normals alone cannot properly scale the stiffness at each contact. For example, Figure 7 depicts a box resting on two thin walls. One wall has many fewer contact points than the other wall. Almost all the contact normals are pointing upward and therefore, based on 3-DoF stiffness scaling, scaling of the contact stiffness would be approximately uniform. Such a haptic simulation would be correct if it is 3-DoF only. However, when rotations are enabled, the 3-DoF scaling will

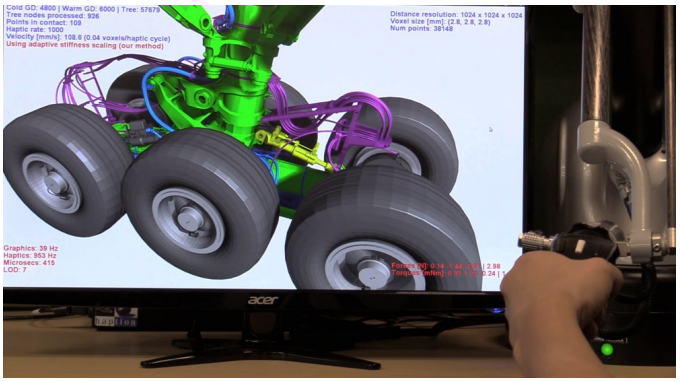


Fig. 9: 6-DoF haptic rendering of contact between the landing gear and a hydraulic actuator (yellow). *We are using the 6-DoF Haption Virtuose 6D device.*

produce large rotations and cause deep penetration into the wall with fewer contact points. Our 6-DoF Gauss map incorporates torque directions T and resolves this problem, since the contacts on the two walls lie at the opposing poles of the torque sphere S^2. Each wall's contact are therefore properly rescaled. The contacts on the wall with dense contact points are scaled down relatively to the other wall, and the total net torques are balanced.

3.3 Virtual Coupling Saturation

In haptic rendering, the penalty forces and torques are typically not rendered directly to the user (a.k.a., "direct rendering") [18], [30]. Instead, it is customary to separate the simulation position of the haptic object from the position imposed by the haptic manipulandum, and connect the two with a spring (virtual coupling [25], [31]). Virtual coupling decreases penetrations, improves stability, and enables a better control of the rendered stiffness. The virtual coupling force F_{VC} is designed to be linear in displacement between manipulandum position x_m and simulation position x_s as

$$F_{VC} = k_{VC}(x_m - x_s), \tag{14}$$

where k_{VC} is the virtual coupling stiffness. However, under such a linear formalism, the virtual coupling force can always be made to overpower the contact forces, causing deep penetrations and pop-throughs. Therefore, the virtual coupling force is often set to saturate to a constant maximum value F_{VC}^{max}, once displacement reaches a certain value x_{max} [4], [26]. The x_{max} typically corresponds to shallow penetrations, such as half a voxel, and we have

$$F_{VC}^{max} = kx_{max}. \tag{15}$$

In our system, we also incorporate virtual coupling torque saturation by limiting it under some user-specified constant values.

Now let us consider the contact configuration with many contact points but with scattered contact normal directions. With global scaling, the stiffness k is scaled to a much smaller value and therefore the total contact net force in the less-sampled contact directions is overpowered by F_{VC}^{max}, causing penetration deeper than d_{max}. One possible solution would be to vary F_{VC}^{max} according to the current stiffness k. However, this either produces too small virtual coupling forces to move the object (such as when all the contact points have the same contact normal), or produces haptic instabilities due to fast variation of the virtual coupling force as

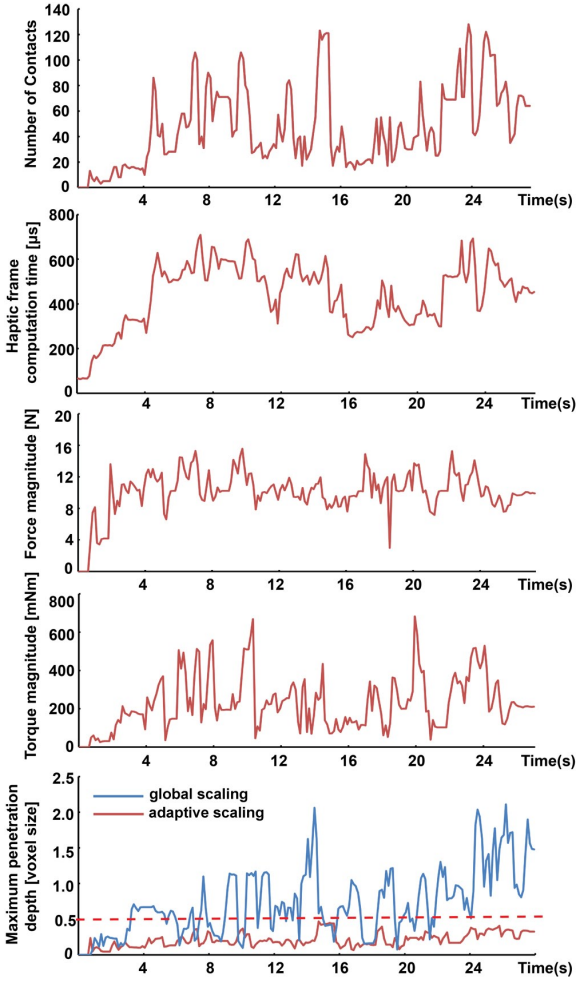


Fig. 10: Analysis of our method (car engine). *The first row shows the number of contacts during the haptic simulation. Our method is stiffer and produces a smaller number of contacts (under* 120*) than global scaling (Figure 1(b)). The computation time (second row) is under* 700 *microseconds per haptic frame, and enables haptic update rates of at least* 1,000 *Hz. Force and torque magnitudes are shown in the third and fourth row, respectively. In the last row, we compared the maximum penetration depth of global scaling vs our method. We observed that when coupled with virtual coupling saturation, our method keeps the maximum penetration depth below half of a distance field voxel (shown dashed), while the global scaling produces much deeper penetrations.*

the contact changes at each haptic cycle. Our Gauss map scaling, however, can assure that the total stiffness in any contact direction will be approximately k and therefore virtual coupling saturation has to balance much more predictable forces. The result is that virtual coupling saturation combined with our scaling works much better than when using global scaling [1], which greatly reduces unwanted penetrations (Figure 10, Figure 11(c)).

4 Results

Our experiments were performed on an Intel Xeon 2.9GHz GPU(2×8 cores) machine with 32 GB of RAM, and a GeForce GTX 680 graphics card with 2 GB of RAM. Our haptic device is the 6-DoF Haption Virtuose 6D (Figure 9), capable of rendering both forces and torques. All the haptic examples were run at the

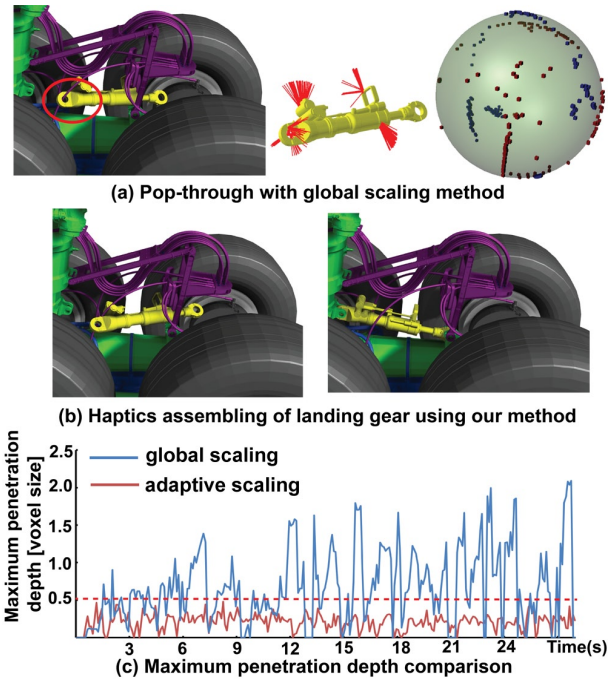


Fig. 11: Haptic assembly of the Boeing 777 landing gear. *(a) Under global scaling [1], the wire collides with the hydraulic actuator ($n = 128$) and incorrectly pops-through into the hole (shown in red circle). (b) Our method is much more robust in handling the collision between the thin features and the manipulated object. The penetration are invisible at all the stages. (c) With virtual coupling saturation, we observe that our method makes it possible to keep the penetration depth under half a distance field voxel, whereas the penetration depths under global scaling are substantially deeper.*

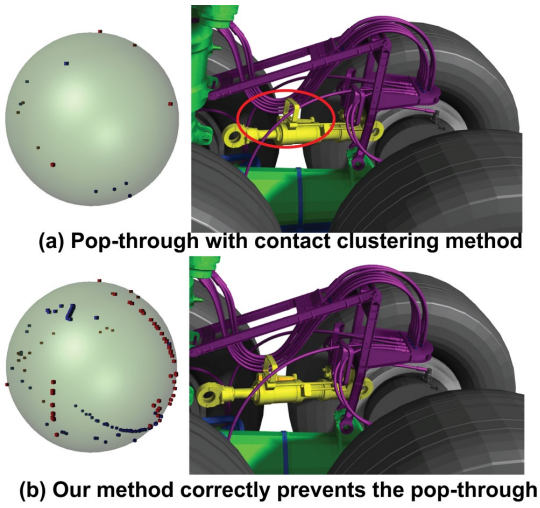


Fig. 12: Comparison to contact clustering. *There are 78 contact points between the hydraulic actuator (yellow) and the landing gear. (a) The contact clustering method [18] groups the contacts into 7 clusters. However, the downsampled contacts cannot prevent pop-throughs between the actuator and the thin wires (shown in red circle). (b) Our method keeps all the contact points, and assures uniform stiffness in all the contact directions, robustly preventing the pop-through. This example demonstrates that the contact points detected by our pointshell vs distance field method are actually needed for good simulation results, and cannot be simplified to a smaller number of contacts.*

1,000 Hz haptic rate. All the signed distance fields were computed using an octree-based method with 8 threads [32].

In our first haptic rendering example, we present the classic "alpha" path planning puzzle [28], [29]. The goal is to find the path to position the rigid red alpha-shaped tube inside the loop of the blue alpha. The red and the blue alphas have identical shapes (1,008 triangles). The blue alpha (distance field object, $512 \times 512 \times 512$) is fixed in space, whereas the red alpha (pointshell object, 25,269 sample points, 7 hierarchical pointshell levels is manipulated as a haptic object. In the real world, the narrow passage between the two sides of each alpha is too small for the blue alpha to directly squeeze into the loop. However, when the global scaling method [1] is used, the large number of contact points ($n = 118$) causes low stiffness and therefore the alpha incorrectly slips past the passage (Figure 8 (a)). In contrast, with our method, it is impossible to force the blue alpha past the gap, unless one actually correctly solves the puzzle (Figure 8 (b)).

In our second example (Figure 1), we manipulate a car starter motor (18,580 triangles, pointshell object, 56,823 points, 7 levels) into its place in the car engine compartment (505,276 triangles, distance field object, $1,024 \times 1,024 \times 1,024$). Our pointshell and distance field resolve the geometry of these objects to a tolerance of $0.9mm$, which is a sufficiently high resolution for realistic virtual assembly. The car engine compartment is fixed while the starter motor is manipulated past a narrow passage inside the car engine compartment. The car engine body has many thin features and hollow parts, which can easily be penetrated or popped-through. Using the global scaling, even with the virtual coupling saturation, it is easy for the pop-throughs to occur. This is especially so when the starter motor is inserted half-way and many contacts are formed with greatly varying normal directions and number of contacts on each contact site. As we can see in the corresponding Gauss map (Figure 1 (b)), multiple contact sites are formed with scattered force and torque directions. Our adaptive stiffness scaling controls the penetration depth well and avoids spurious penetrations (Figure 10). As such, our method greatly improves the quality of haptic virtual assembly simulations.

In our third example, we used the Boeing 777 landing gear model for haptic assembly. The pointshell object is a hydraulic actuator (25,191 triangles, 38,148 points, 7 levels, Figure 11) and is manipulated to be installed into the bottom of the landing gear (1,789,551 triangles, distance field object, $1,024 \times 1,024 \times 1,024$, corresponding to $2.8mm$ per distance field voxel). The landing gear has many thin features, such as the wires with width of only 2 voxels. When manipulating the hydraulic actuator in the environment, we observed that using global scaling, it is easy for pop-throughs to occur. Our method, coupled with virtual coupling saturation, works well, and does not produce visible penetrations.

In Figure 12, we compare our adaptive stiffness scaling method to contact clustering [18]. Such clustering groups geometrically neighboring contacts into one contact group, based on their Euclidean distance. We select the clustering threshold to be 0.5 voxel size. Each cluster is replaced with a single contact whose position and force normal are determined by averaging the contacts in a cluster, weighted by the penetration depth. The clustering alleviates the instability problem of stiffness accumulation by downsampling the contact distribution. However, under-sampled but important contact directions are ignored and

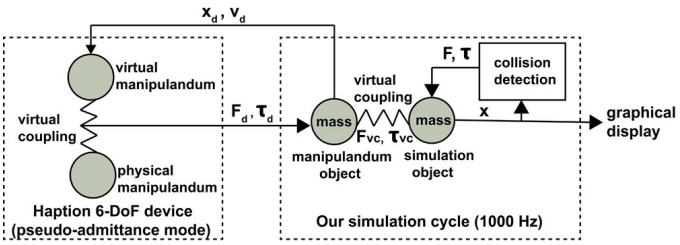


Fig. 13: Haptic simulation in (pseudo-)admittance mode. *Our simulation (1,000 Hz) reads forces F_d and torques τ_d from the device, and computes positions and velocities x_d, v_d. Internally in the Haption driver, the admittance mode is implemented using virtual coupling, hence pseudo-admittance. Both the manipulandum and simulation objects have mass and inertia (we use equal mass and equal inertia), and are timestepped as rigid bodies, under forces/torques $F_d, F_{VC}, \tau_d, \tau_{VC}$ and $F, -F_{VC}, \tau, -\tau_{VC}$, respectively.*

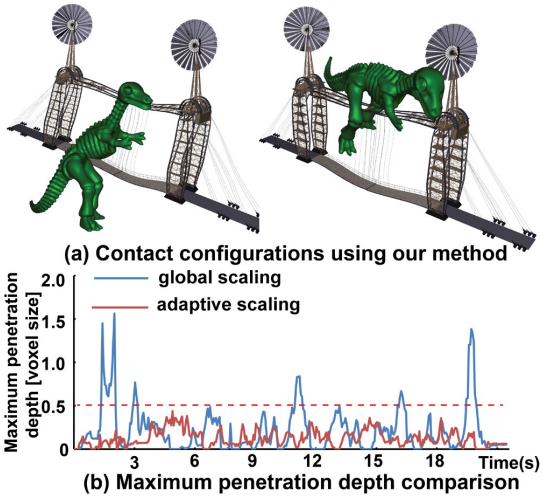


Fig. 14: Rigid vs deformable object haptic simulation with our method. *Two simulation frames with distributed contacts are shown ($n = 18, 27$, respectively). No visible penetrations occur. Our penetration depth is smaller than with global scaling.*

therefore cause deep penetrations (Figure 12 (a)). This problem is more pronounced in the presence of thin geometries and complex distributed contact. Our method keeps all the contact points and remains stable (Figure 12 (b)). Furthermore, for applications where clustering is desired, our method can improve the stability of contact clustering, by adjusting the contact stiffness for the clustered contact distribution.

Our Haption Virtuose 6D haptic device is an impedance device, but it can also operate in pseudo-admittance mode. The input to the impedance mode are the 6-DoF positions (including rotations) and 6-DoF velocities, and the output of our program sent to the haptic device are the forces and torques. The admittance mode is the opposite: the device provides the forces and torques, and the simulation software has to compute positions, orientations and 6-DoF velocities. In all of our examples, we prefer the pseudo-admittance mode over the impedance mode because our haptic device tends to generally (independently of our method) be more stable in the admittance mode. This is because pseudo-admittance is essentially (in the device driver provided by the manufacturer of the device) implemented using virtual coupling (Figure 13), which limits the stiffness that can ever be sent to the device. The downside of admittance rendering is that the forces and torques in the free space are not zero. Our method can be considered as a filter for penalty forces, torques and their corresponding gradients. Thus, it can be integrated with direct rendering or with dynamic [1] or static [4] virtual coupling. It works with explicit and implicit integration, and it can be applied either to impedance or admittance simulations. We tried all of these combinations and choose to use dynamic virtual coupling with implicit integration in admittance mode for best stability. At each haptic cycle, we read the forces and torques from the device. We use them to implicitly timestep rigid body dynamics of both the manipulated object and simulation object, subject to virtual coupling and the contact forces and torques, and their 6×6 gradients [2], [26] (see Figure 13). Implicit timestepping is performed by solving a 6×6 linear system [2]. The updated position and velocity of the manipulandum object are sent to the haptic device.

In our fourth example, we applied our method to rigid-deformable haptic simulations with static virtual coupling in impedance mode (Figure 14). The rigid dinosaur (56,192 triangles) is modeled as a $256 \times 256 \times 256$ signed distance field, while the nonlinear reduced-deformable bridge (59,630 triangles) is sampled with 85,018 points. In deformable simulations, penetrations caused by global scaling are less severe than for rigid objects, due to the extra deformable DoFs available to resolve the contact. Even in this deformable example, our stiffness scaling method produces a smaller penetration depth than global scaling, and gives quality haptic simulations under all conditions (Figure 14).

5 Conclusion

We presented a fast stiffness scaling method that uses the Gauss map to greatly improve the stability of penalty-based 6-DoF haptic rendering. The method assigns spatially-varying stiffness scaling weights to each contact point and assures equal stiffness in all the contact directions. Our method makes virtual coupling saturation very effective and can prevent deep penetrations and pop-throughs.

In the future, we would like to apply our adaptive scaling to penalty-based animations for computer graphics (as opposed to haptics). Although our haptic examples are stable, scaling the stiffness temporally generally leads to non-conservative simulations, an issue that applies equally well to previous work [1] and our work. In haptic rendering simulations, this issue is not critical because of the presence of virtual coupling. Our method makes it possible for penetrations to remain shallow even in complex scenarios with large differences in the number of samples on each contact site. It would be interesting to investigate a hybrid method combining aspects of both constraint-based and penalty-based methods that is fast, stable and avoids penetrations.

Acknowledgments

This research was sponsored by the National Science Foundation (CAREER-1055035, IIS-1422869), Sloan Foundation, Okawa Foundation, and USC Annenberg Graduate Fellowship to Hongyi Xu. We thank Bohan Wang for real-time OpenGL rendering.

REFERENCES

[1] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy, "Six degree-of-freedom haptic rendering using voxel sampling," in *Proc. of ACM SIGGRAPH*. ACM, 1999, pp. 401–408.

[2] H. Xu, Y. Zhao, and J. Barbič, "Implicit multibody penalty-based distributed contact," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 9, 2014.

[3] S. Laycock and A. Day, "A survey of haptic rendering techniques," *Computer Graphics Forum*, vol. 26, pp. 50–65, 2007.

[4] M. Wan and W. A. McNeely, "Quasi-static approximation for 6 degrees-of-freedom haptic rendering," in *Proc. of IEEE Visualization*, 2003, pp. 257–262.

[5] W. McNeely, K. Puterbaugh, and J. Troy, "Voxel-based 6-dof haptic rendering improvements," *Haptics-e*, vol. 3, no. 7, 2006.

[6] O. Astley and V. Hayward, "Multirate Haptic Simulation Achieved by Coupling Finite Element Meshes Through Norton Equivalents," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1998.

[7] G. Debunne, M. Desbrun, M.-P. Cani, and A. H. Barr, "Dynamic real-time deformations using space & time adaptive sampling," in *Proc. of ACM SIGGRAPH*, 2001, pp. 31–36.

[8] F. Barbagli, D. Prattichizzo, and K. Salisbury, "A multirate approach to haptic interaction with deformable objects single and multipoint contacts," *The International Journal of Robotics Research*, vol. 24, no. 9, pp. 703–715, 2005.

[9] M. Renz, C. Preusche, M. Pötke, H.-P. Kriegel, and G. Hirzinger, "Stable haptic interaction with virtual environments using an adapted voxmap-pointshell algorithm," in *Proc. of Eurohaptics*, 2001, pp. 149–154.

[10] M. Sagardia, T. Hulin, C. Preusche, and G. Hirzinger, "Improvements of the voxmap-pointshell algorithm-fast generation of haptic data-structures," in *53rd IWK-Internationales Wissenschaftliches Kolloquium, Ilmenau, Germany*, 2008.

[11] J. Barbič and D. L. James, "Six-dof haptic rendering of contact between geometrically complex reduced deformable models," *IEEE Transactions on Haptics*, vol. 1, no. 1, pp. 39–52, 2008.

[12] C. Zilles and J. Salisbury, "A constraint-based god-object method for haptics display," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1995, pp. 146–151.

[13] P. Berkelman, R. Hollis, and D. Baraff, "Interaction with a realtime dynamic environment simulation using a magnetic levitation haptic interface device," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1999, pp. 3261–3266.

[14] 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 Trans. on Visualization and Computer Graphics*, vol. 13, no. 3, pp. 458–469, 2007.

[15] I. Peterlik, M. Nouicer, C. Duriez, S. Cotin, and A. Kheddar, "Constraint-based haptic rendering of multirate compliant mechanisms," *IEEE Transactions on Haptics*, vol. 4, no. 3, pp. 175–187, 2011.

[16] H. Courtecuisse, J. Allard, P. Kerfriden, S. P. Bordas, S. Cotin, and C. Duriez, "Real-time simulation of contact and cutting of heterogeneous soft-tissues," *Medical image analysis*, vol. 18, no. 2, pp. 394–410, 2014.

[17] D. M. Kaufman, S. Sueda, D. L. James, and D. K. Pai, "Staggered projections for frictional contact in multibody systems," *ACM Transactions on Graphics*, vol. 27, no. 5, pp. 164:1–164:11, 2008.

[18] Y. J. Kim, M. A. Otaduy, M. C. Lin, and D. Manocha, "Six degree-of-freedom haptic display using incremental and localized computations," *Presence-Teleoperators and Virtual Environments*, vol. 12, no. 3, pp. 277–295, 2003.

[19] Q. Luo and J. Xiao, "Physically accurate haptic rendering with dynamic effects," *Computer Graphics and Applications, IEEE*, vol. 24, no. 6, pp. 60–69, Nov 2004.

[20] M. A. Otaduy and M. C. Lin, "A modular haptic rendering algorithm for stable and transparent 6-dof manipulation," *IEEE Trans. on Robotics*, vol. 22, no. 4, pp. 751–762, 2006.

[21] L. Glondu, S. C. Schvartzman, M. Marchal, G. Dumont, and M. A. Otaduy, "Fast collision detection for fracturing rigid bodies," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 1, pp. 30–41, 2014.

[22] P. Volino and N. Magnenat-Thalmann, "Efficient self-collision detection on smoothly discretized surface animations using geometrical shape regularity," *Comp. Graphics Forum*, vol. 13, no. 3, pp. 155–166, 1994.

[23] S. C. Schvartzman, J. Gascón, and M. A. Otaduy, "Bounded normal trees for reduced deformations of triangulated surfaces," in *Symp. on Computer Animation (SCA)*, 2009, pp. 75–82.

[24] J. Barbič, "Real-time reduced large-deformation models and distributed contact for computer graphics and haptics," Ph.D. dissertation, Carnegie Mellon University, aug 2007.

[25] J. Colgate, M.C.Stanley, and J.M.Brown, "Issues in the haptic display of tool use," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1995, pp. 140–145.

[26] M. A. Otaduy and M. C. Lin, "Stable and Responsive Six-Degree-of-Freedom Haptic Manipulation Using Implicit Integration," in *Proc. of the World Haptics Conference*, 2005, pp. 247–256.

[27] S. LaValle, *Planning algorithms*. Cambridge Univ. Press, 2006.

[28] O. B. Bayazit, G. Song, and N. M. Amato, "Enhancing randomized motion planners: Exploring with haptic hints," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2000, pp. 529–536.

[29] J. J. Kuffner, "Effective sampling and distance metrics for 3d rigid body path planning," in *IEEE Int. Conf. on Robotics and Automation*, 2004.

[30] D. D. Nelson, D. E. Johnson, and E. Cohen, "Haptic rendering of surface-to-surface sculpted model interaction," in *ACM SIGGRAPH 2005 Courses*, 2005, p. 97.

[31] R. J. Adams and B. Hannaford, "A two-port framework for the design of unconditionally stable haptic interfaces," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1998, pp. 1254–1259.

[32] H. Xu and J. Barbič, "Signed distance fields for polygon soup meshes," in *Proc. of the Graphics Interface Conference*, 2014, pp. 35–41.

Hongyi Xu is a PhD student in computer science at the University of Southern California. He obtained his BS degree from Zhejiang University. His research interests are in computer graphics, physically based animation, contact and interactive physics.

Jernej Barbič is associate professor of computer science at USC. In 2011, MIT Technology Review named him one of the Top 35 Innovators under the age of 35 in the world (TR35). Jernej's research interests include nonlinear solid deformation modeling, model reduction, collision detection and contact, and interactive design of deformations and animations. He is the author of Vega FEM, an efficient free C/C++ software physics library for deformable object simulation. Jernej is a Sloan Fellow (2014).