Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS (2015)
F. Jaillet, G. Zachmann, and F. Zara (Editors)

# Accurate Contact Modeling for Multi-rate Single-point Haptic Rendering of Static and Deformable Environments

Thomas C. Knott[1] and Torsten W. Kuhlen[2]

[1] Virtual Reality Group, RWTH Aachen University, Germany
[2] Jülich Supercomputing Center, Germany

**Abstract**

*Common approaches for the haptic rendering of complex scenarios employ multi-rate simulation schemes. Here, the collision queries or the simulation of a complex deformable object are often performed asynchronously on a lower frequency, while some kind of intermediate contact representation is used to simulate interactions on the haptic rate. However, this can produce artifacts in the haptic rendering when the contact situation quickly changes and the intermediate representation is not able to reflect the changes due to the lower update rate.*

*We address this problem utilizing a novel contact model. It facilitates the creation of contact representations that are accurate for a large range of motions and multiple simulation time-steps. We handle problematic convex contact regions using a local convex decomposition and special constraints for convex areas. We combine our accurate contact model with an implicit temporal integration scheme to create an intermediate mechanical contact representation, which reflects the dynamic behavior of the simulated objects. Moreover, we propose a new iterative solving scheme for the involved constrained dynamics problems. We increase the robustness of our method using techniques from trust region-based optimization. Our approach can be combined with standard methods for the modeling of deformable objects or constraint-based approaches for the modeling of, for instance, friction or joints. We demonstrate its benefits with respect to the simulation accuracy and the quality of the rendered haptic forces in multiple scenarios.*

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Information Interfaces and Presentation]: User Interfaces—Haptic I/O, Benchmarking

## 1 Introduction

Haptic force cues are an important feedback channel in humans' interaction with their environment. Haptic rendering algorithms are employed to synthesize these forces computationally. In many applications a physically realistic interaction is required. Here, the haptic rendering tries to simulate the correct dynamic behavior of the involved objects. Such algorithms typically consist of two main steps: first, the contact situation between the objects is determined and second, a dynamics problem is setup and solved to compute the movements of the objects. The second step is usually done using a fixed set of contacts resulting from the first step. In so-called "constraint-based" simulation approaches, the contacts are modeled using mathematical constraint equations, which are included into the dynamics problem [ESHD05]. These constraints restrict the solutions of the dynamics problem to the so-called *configuration space*. The latter defines all possible states the objects can take without violating the constraints.

For the simulation of an unilateral contact, a constraint is commonly defined via an inequality. It defines that the distance between two objects, or rather two geometric features, has to be larger or equal to zero. In real-time simulations, these constraints are typically approximated by *linear* inequalities before they are incorporated into the dynamics problem [ESHD05]. Such a linear inequality reduces the configuration space by an infinite half-space. [NT10] showed that this can create problems as the half-space does not approximate the geometric features that the constraint represents very well.

This is especially problematic in concave contact situations. A simple but still problematic scenario is shown in Figure 1.

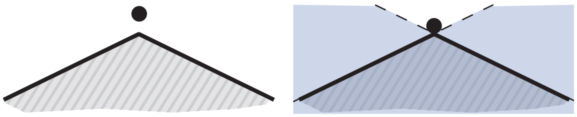

**Figure 1:** *(Left) Convex contact situation. (Right) Reduction of the configuration space by two half-space-based constraints. Unavailable space is shown in blue.*

DOI: 10.2312/vriphys.20151336

Here, the movement of the dot has to be simulated for the next time-step. To this end, first the constraint set applied to the dynamics problem during the step needs to be determined. However, it is not clear how to model the actual geometry using half-space-like constraints. In the example, one constraint is used for each surface side. Due to their infinite extend, they reach into the actual free space and reduce the configuration space more than necessary. As the constraint set is fixed for the time-step, the point is restricted to stay in the non-blue area until the next step. Although it is possible to choose a different constraint set, it will always forbid movements which are valid from a *kinematic* point of view. [NT10] demonstrated that this is in concave contact situations always the case. This over-reduction of the configuration space is a problem if it inhibits to simulate the correct *dynamic* behavior of the objects. In our example, this would be the case if the dot not just falls down but has additionally a slight momentum to the left. Here, the contact set would lead to an abrupt stopping of the dot outside of the geometry. The correct dynamic behavior could be simulated using a constraint set, which *only* contains a constraint for the left face. In general, a badly chosen constraint set can impair the simulation accuracy significantly [ESHD05]. Therefore, the choice of the constraint set for the time-step is a critical decision.

Indeed, there are several different approaches to specify the constraint set; each with its own advantages and drawbacks. [AP97], for instance, employ only constraints for geometric features, which are already in a touching or penetrating state. However, [NT10] showed that this results in inter-penetrations, which get worse with an increasing time-step length and higher object velocities. Other approaches use a temporal back-tracking. Here, the constraint set is iteratively refined until a set is found, which allows to simulate the correct dynamic behavior of the objects [ESHD05]. This includes the solving of multiple intermediate dynamics problems. As such a process is computationally expensive, it is usually not employed for haptic rendering. A further option is to choose the contact set using an anticipation of the objects' movements. Such an anticipation is typically based on the unconstrained movement of the objects [DD06]. Consequently, it does not reflect any collision response, which can be problematic when constraints influence each other.

The choice of the constraint set gets even more difficult in case of multi-rate haptic rendering methods where the contact configuration is determined on a frequency lower than the one of the haptic simulation. This is a quite common approach, in case collision queries or soft tissue simulations are computationally too expensive to be done on a haptic simulation rate [OL06, DD06, DPD*13]. Here, a constraint set is determined on a low rate and then used in multiple time-steps on the haptic rate to simulate the user's interaction. Therefore, the constraint set must be appropriate for a *sequence* of simulation steps. As the user interacts with the simulated objects during this sequence, the anticipation of their movements to select a good constraint set is rather difficult. Furthermore, there are situations for which no appropriate constraint set exists. This is the case when the user moves an object over a convex corner during such a sequence. Consider our example and a user that controls the dot and moves it from left to right over the surface; at the beginning, *only* the left constraint should to be in the contact set, while at the end, *only* the right one should. In consequence, there is no appropriate *single* constraint set for the complete sequence. This can result in severe artifacts in the haptic rendering as we will show in our experiments (see sec.5).

The problems originate from the bad approximation of the actual object geometry by the linearized constraints. The induced reduction of the configuration space by complete infinite half-spaces makes the choice of the constraint set for a time-step a crucial and hard decision. The problems get even more severe in a haptic multi-rate simulation when the set is in use for more then one time-step.

To resolve these problems, we propose a new efficient contact model in this paper, which allows to accurately model the actual object geometry during the solving process of the dynamics problem. It facilitates the creation of configuration spaces, which are accurate for a larger range of motions and multiple simulation time-steps. We build upon a method presented by [KK14] which employs so-called geometrically limited constraints (GLC). These do not reduce the configuration space by an infinite half-space but only by an amount which roughly reflects the geometric features they represent. Nevertheless, the method has some problems as described by [KK14] and are further analyzed in this paper. One major problem is that the GLCs only approximate the geometry. As result, they still reduce the configuration space more than necessary in convex regions and create artifacts in the haptic rendering. We resolve this and other problems of the contact model by incorporating a special treatment of convex regions. This is done based on a convex decomposition, which is performed locally around contacts during the solving process of the dynamics problem. We furthermore increase the robustness of our method by applying techniques from trust region-based optimization.

We apply the proposed contact model in a multi-rate haptic rendering for point-based interaction with static and deformable scenes. To this end, we employ a mechanical intermediate contact representation, which is created for the contact situation on a lower rate and used to simulate interactions on a haptic rate. The intermediate representation is based on our contact model and therefore, reflects the actual geometry in contact over multiple time-steps. Furthermore, it incorporates the dynamic behavior of the involved objects, especially their compliance. Thereby, it allows, on the haptic rate, to simulate the changes in the contact situation, which result from the compliant behavior of objects, without updating the representation. As result, a high simulation quality is also guaranteed if the update of the representation is much lower than the haptic frame rate. In our evaluation, we show that the proposed methods allow an artifact-free haptic rendering of static and deformable scenarios, and demonstrate their haptic real-time capabilities.

The main contribution of this paper is a new contact model and its incorporation into an intermediate mechanical contact representation for single-point multi-rate haptic rendering. The approach allows an accurate contact simulation for physics-based interaction with static and deformable objects. One major advantage is that it does not require a specific technique to model deformable or rigid objects and it is compatible with other constraints used, for instance, to model friction or joints.

The remaining paper is organized as follows: after describing the related work, we will introduce our proposed methods in section 3; a more detailed overview is given at the beginning of the section. In section 4, we describe how the techniques are combined with an efficient simulation of deformable objects. Finally, we evaluate and discuss our methods in section 5.

## 2 Related Work

In this section, we provide an overview on the investigated approaches in the area of haptic rendering relevant for our work. Most of the methods rely on a virtual coupling approach where a proxy object is coupled to the input of the haptic device to increase the rendering stability [AH99]. The type of the proxy object is chosen based on the application. Some use single-points, as for instance [ZS95] or lately [BS13], [KK14] and our approach. Others use rigid bodies [WM03, OL06, BJ08] or deformable objects [DD06, DPD*13]. The rendering techniques can be furthermore distinguished by the way contacts are handled. Some apply penalty forces, for instance [WM03, BJ08], which have the disadvantage that they result in object inter-penetrations during the simulation. In order prevent these, others use mathematical bilateral [ZS95] or unilateral [DD06, OTSG09] constraints. The latter type is also employed to model frictional contact based on Coulombs law [DD06, OTSG09]. The equations describing the unilateral constraints are commonly linearized in real-time applications. Like detailed in the introduction, this creates problems in convex contact situations because the configuration space does not properly reflect the geometry. [NT10] tackled this problem for convex geometries in the context of rigid-body simulations. They model each convex object through a set of linear inequalities which are combined using an intersection operation.

The computation of a complete simulation time-step can be too time consuming to be performed on a haptic frame rate. Therefore, so-called multi-rate approaches perform computationally expensive parts asynchronously on a lower frequency [OL08, PDC11, DPD*13]. For complex scenes, one potentially expensive part is the collision detection. Therefore, [OL06] investigated an approach for the haptic rendering of a static environment in which the collision detection was performed on a lower rate. To increase the rendering quality, a linear update of the penalty-based collision response was proposed. When it comes to scenes including complex deformable objects, the setup and solving of the dynamics problem gets an additional computational burden. Some methods circumvent the problem by reducing the complexity of the objects to be able to simulate them on a high frequency [BJ08, DDCB01]. [WSL14], for instance, simulated deformable objects based on sphere-trees augmented with spring-dampers. Although, such methods allow the simulation on haptic rates, they come at the cost of physical accuracy. As this is not suitable in many applications, other methods facilitate complex deformable objects by performing their simulation on a lower rate, while the simulation of the objects controlled by the haptic device is performed on a high haptic rate. [PDC11] proposed a method where these objects are able to interact with each other via an intermediate contact representation. The latter is computed on a lower rate and shared with the high rate. Although the used intermediate representation reflects the compliance of the objects, it allowed only a quasi-static simulation. The approach was extended by [DPD*13] to handle the full dynamics of the involved objects. The used intermediate contact representations are based on linear unilateral constraints. As described in the introduction, the linearization does not allow to accurately model the geometry in convex contact situations, especially in multi-rate simulation schemes. This creates artifacts in the haptic rendering as we will show in section 5.

The main objective of our approach is to provide a contact model which resolves this problem. The proposed model facilitates a description of a contact configuration, which stays accurate for a larger range of motions than a description based on linear inequalities. The method also fits well with standard mechanical descriptions of deformable objects and constraint-based methods to model, for instance, friction or joints.

## 3 Accurate Contact Modeling for Multi-rate Haptic Rendering

In this section, we will introduce our proposed contact model. First, we will describe the fundamental dynamics equations and notations. Then, we will lay out a multi-rate simulation approach (sec.3.2), which gives the needed context for the new contact model detailed afterwards (sec.3.3). The model builds upon the method of geometrically limited constraints [KK14]. We will analyze its problems in our context and resolve them using a local convex decomposition of the contact area and special constraints for convex areas. Afterwards, we present an iterative solving scheme for the created constrained dynamics problems based on a Projected Gauss-Seidel scheme (sec.3.4). We increase the robustness of this process utilizing techniques from trust region-based optimization (sec.3.5).

### 3.1 Dynamics Fundamentals and Notations

This section introduces the notations and equations of dynamics used throughout the paper. For more details and corresponding proofs, we refer to fundamental literature like [ESHD05]. The mechanical behavior of the simulated objects is described by the usual equation of dynamics:

$$\mathbf{M}\mathbf{a} = \mathbf{f}(\mathbf{x},\mathbf{v}), \tag{1}$$

where **a** specifies the acceleration, **x** and **v** are respectively positions and velocities, and **M** is a mass matrix. The external and internal forces originating from deformation, gravity etc. are defined by **f**. This equation is used to model the haptic proxy as a mass point, the virtual coupling as a spring-damper, and the deformable objects via the finite element method as described in section 4. Contact forces are integrated using Signorini's contact model [OL08] and defined via a complementarity condition:

$$0 \leq \lambda \perp \psi(\mathbf{x},\mathbf{v}) \geq 0, \tag{2}$$

where $\lambda$ is the contact force magnitude and $\psi$ is a function, which describes the signed distance between the bodies with respect to the contact.

For a stable temporal integration, a backward Euler scheme using linear approximations of the forces is employed. The dynamics problem is formulated with a fixed time-step $h$ on a velocity level as:

$$\underbrace{(\frac{1}{h}\mathbf{M}-\mathbf{B}-h\mathbf{K})}_{\mathbf{A}}d\mathbf{v} = \underbrace{\mathbf{f}(\mathbf{x}_{t_i},\mathbf{v}_{t_i})+h\mathbf{K}\mathbf{v}_{t_i}}_{\mathbf{b}}+\mathbf{J}^T\boldsymbol{\lambda}, \tag{3}$$

where $\mathbf{x}_{t_i}$ and $\mathbf{v}_{t_i}$ are the generalized coordinates and velocities of the whole simulated system at the beginning of the time-step $t_i$. $\mathbf{B} = \partial\mathbf{f}/\partial\mathbf{v}$ and $\mathbf{K} = \partial\mathbf{f}/\partial\mathbf{x}$ are the according partial derivatives of the internal forces and $\mathbf{J} = \partial\boldsymbol{\psi}/\partial\mathbf{v}$ is used to apply the constraint forces to the objects. Finally, $d\mathbf{v}$ denotes the change in the velocity that is to be computed. This equation is augmented by complementarity conditions for the unilateral contacts:

$$\mathbf{0} \leq \boldsymbol{\lambda} \quad \perp \quad \mathbf{J}(\mathbf{v}_{t_i}+d\mathbf{v})+\frac{\boldsymbol{\psi}_{t_i}}{h} \geq \mathbf{0}, \tag{4}$$

where $\boldsymbol{\lambda}$ are the stacked contact forces of all constraints and $\boldsymbol{\psi}_{t_i}$ are the stacked values of the corresponding constraint functions at the beginning of the time-step. These conditions enforce a penetration free state at the end of the time-step with respect to the incorporated constraints. Other constraint types, e.g. modeling friction, are possible and are considered later on. The latter two equations form together a Mixed Linear Complementarity Problem (MLCP).

In single-rate simulation schemes, each time-step would include the following two steps: (i) the constraint set is determined based on a proximity query, and (ii), the according MLCP is setup and solved. This conflicts in some simulation scenarios with the real-time requirements in haptic rendering. One reason may be that the proximity queries are computationally too expensive, another, that the mechanics of the simulated objects are too complex, for instance, when deformable objects are simulated. Therefore, a common approach is to perform a multi-rate simulation, in which computational demanding parts are done on a lower rate.

### 3.2 Multi-rate Simulation

In this section, we describe a basic multi-rate simulation scheme, which provides the necessary context for our new contact model. The latter is described in the next section and integrated into this scheme. Like [DPD*13], our multi-rate scheme performs proximity queries on a low rate to setup a constraint set, which is used for multiple simulation steps on a high rate. On the low rate (*LR*), we linearize the dynamic system and setup the dynamics problem (3) including the constraints (4). On the high haptic rate (*HR*), the resulting dynamics problem is partially updated and solved to compute the new object states. For reasons of performance, the problem is reformulated using the Schur complement as a pure Linear Complementarity Problem (LCP). This can then be efficiently solved using iterative methods as the Projected Gauss-Seidel Method [ESHD05]:

$$0 \leq \boldsymbol{\lambda} \perp \underbrace{\mathbf{J}\mathbf{A}^{-1}\mathbf{J}^T}_{\mathbf{A}_\psi}\lambda+\underbrace{\boldsymbol{\psi}_{t_i}/h}_{\mathbf{b}_\psi}+\underbrace{\mathbf{J}(\mathbf{v}_{t_i}+h\mathbf{A}^{-1}\mathbf{f}_{t_i})}_{\Delta\mathbf{b}_\psi} \geq \mathbf{0}, \tag{5}$$

where $\mathbf{b}_\psi$ reflects the constraint values at the beginning of a time-step and $\Delta\mathbf{b}_\psi$ how the values would change during this step in case of an unconstrained motion. Furthermore, $\mathbf{A}_\psi$ denotes the constraint response matrix (CRM). It approximates how the values of the constraint functions $\boldsymbol{\psi}$ will change for a set of constraint forces $\boldsymbol{\lambda}$ and can be seen as a mechanical intermediate representation for the haptic rendering. We emphasize that the CRM reflects the complete dynamics of the involved objects under consideration of the temporal integration scheme. In case of deformable objects, this also includes their compliant behavior. All in all, the multi-rate simulation algorithm then encompasses the following steps:

Low rate
L1. Setup dynamics problem $\mathbf{Ax}=\mathbf{b}$
L2. Perform proximity query
L3. Define constraint set and setup Jacobi $\mathbf{J}$
L4. Compute constraint response matrix $\mathbf{A}_\psi$
L5. Share data with high rate

High rate
H1. Receive haptic device state
H2. Compute $\mathbf{b}_\psi$,$\Delta\mathbf{b}_\psi$
H3. Solve LCP($\mathbf{A}_\psi$,$\mathbf{b}_\psi$,$\Delta\mathbf{b}_\psi$) $\rightarrow \boldsymbol{\lambda}$
H4. Compute new object states using (3)
H5. Send virtual coupling forces to haptic device
H6. Share object states with low rate

Notice that this approach already allows a simulation with deformable objects. Nevertheless, it will be slow due to the full motion integration in step H4. We address this topic further in section 4.

In the simulation scheme described above, the contact constraints are setup in *LR*. Therefore, they will be used in multiple *HR*-time-steps and for a range of different user inputs. Like detailed in the introduction, problems arise in convex contact situations from the linearization of the constraints in such a multi-rate application. Here, the constraint set models the involved actual shape of the geometry unsatisfactorily. In the following, we present a novel contact model which resolves the problem.

### 3.3 Accurate Contact Modeling

The main objective of the proposed contact model is to facilitate a description of a contact configuration, which stays accurate for a larger range of motions. Thereby, it accurately models the contacts over a longer period of time, i.e., in a sequence of multiple *HR*-time-steps. To this end, we integrate more information about the underlying actual geometry into the contact description, so that it reflects all parts of the geometry, which may be involved in the contact during the sequence. Furthermore, we introduce a special treatment of the problematic convex areas (see sec.1) to handle these properly.

Our proposed approach builds upon the method of geometrically limited constraints (GLC) described by [KK14]. We furthermore modify it, to resolve the shortcomings of the method in context of our rendering framework. The basic idea behind the GLCs is that as a contact constraint corresponds to a specific geometric feature with a finite size, the according constraints should also be restricted to a corresponding region.

To model such a GLC, first, a coordinate system is defined around each contact point using tangential vectors. Second, tangential functions $\tau_1(\mathbf{x},\mathbf{v},t)$ and $\tau_2(\mathbf{x},\mathbf{v},t)$ are defined to describe the contact position with respect to these tangential directions. This is done in analogy to the signed distance function $\boldsymbol{\psi}$ for the normal direction. These are then utilized to define constraints that are limited to specific regions. Hence, for a rectangular region the constraint should only be active if:

$$-s_1 \leqslant \tau_1(\mathbf{x},\mathbf{v},t) \leqslant s_1 \quad \textit{and} \quad -s_2 \leqslant \tau_2(\mathbf{x},\mathbf{v},t) \leqslant s_2 \tag{6}$$

holds, where $s_1$ and $s_2$ specify the size of the active region (see Fig.2 left).

GLCs are created for all entities delivered by the proximity queries in step L2 and form the constraint set $C_{\text{GLC}}$. The entities can be triangles, point shell points, or edges. The latter can be used to create GLCs for sharp concave contact areas (see Fig.2 right); we will come back to this topic in section 5. In contrast to [KK14], we use the same $C_{\text{GLC}}$ in multiple iterations of *HR*. Therefore, it has to be valid over a longer period and farther movements as in a single-rate simulation. Consequently, we also need to perform the proximity queries with a higher proximity threshold (see sec.5).

This higher proximity threshold can create problems with highly concave or thin geometries. If the user interacts, for instance, with the arm of the statue in the right of Figure 6, $C_{\text{GLC}}$ contains GLCs for the bottom side too. However, in the original method proposed by [KK14] a GLC is not limited in depth. Consequently, GLCs from opposing parts of the geometry will overlap and create conflicts. To prevent this, we additionally limit the active region of a GLC (6) in the normal direction $\psi$ by a depth d (see Fig.2 left):

$$-d \leqslant \psi(\mathbf{x},\mathbf{v},t) \leqslant 0, \tag{7}$$

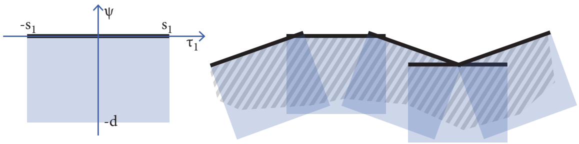

**Figure 2:** *(Left) GLC with coordinate system (Right) GLC contact set with concave and convex areas.*

Nevertheless, further issues emerge from overlaps between the active regions of the GLCs in convex contact situations. A first problem was already described by the authors of [KK14]. It arises when the borders of the constraints are not perfectly aligned at the surface (see Fig.3 left). Here, the overlaps create a small concave surface which results in a locking effect as described in the introduction. The impact of the effect on the simulation quality increases with the size of the overlap. Therefore, the simulation quality strongly depends on a good matching sampling of the geometry surface by the GLCs. This can be cumbersome in case of complex geometries, as it requires a high number of small constraints, which reduces the efficiency of the approach. Another problem occurs if constraints are combined with an angle that is too sharp (see Fig.3 right). Here, the active volumes of the constraints reach into the actually free space. As a result, the configuration space is reduced more than necessary. Nevertheless, even if the constraints do not overlap on the surface, there is an overlap in the inside (see Fig.3 middle). Therefore, it becomes ambiguous which constraint should be active in case the constraints are already violated. The latter is possible due to the overall approximative nature of simulations based on linearized dynamics. Furthermore, in case an iterative solving approach is employed, the violation can occur during the solving process as described in section 3.4.

To resolve the demonstrated problems, we introduce a method based on a convex decomposition of the active GLCs. To this end, we build upon the approach described by [NT10]. Here, a convex object involved in a contact is defined via a set of half-spaces $j = 1,..,m$ modeled via linear inequalities $\psi_j(\mathbf{x},\mathbf{v},t) \geqslant 0$. For a complete convex object there is no penetration if at least one of these inequalities is non-violated. Therefore, one can use the following condition to define a non-penetration constraint for the object:

$$\exists j \in 1..m : \psi_j(\mathbf{x},\mathbf{v},t) \geqslant 0 \Leftrightarrow \max_{j=1,..,m} \{\psi_j(\mathbf{x},\mathbf{v},t)\} \geqslant 0. \tag{8}$$

Nevertheless, to be able to apply this method, a convex decomposition of the involved geometries is needed. The approach described by [NT10] is based on a complete convex decomposition of the objects. This is not only computationally quite expensive for complex scenarios, but can also degenerate for

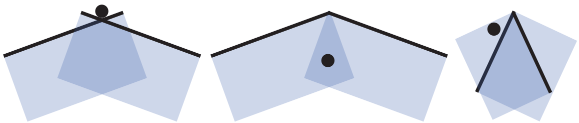

**Figure 3:** *Problems due to GLC overlaps: (left) locking effect, (middle) constraint ambiguity, (right) too sharp corner.*

geometries with a bowl-like shape [Eri05], which tend to occur when pressing on soft materials. As we target a simulation with deformable objects, the decomposition is furthermore necessary in each time-step. However, we need the convexity information only for the currently active GLCs. To this end, we first determine the subset $C^a_{\mathrm{GLC}} \subseteq C_{\mathrm{GLC}}$ of the GLCs that are currently in their active region. This is done based on (6) and (7) as described in more detail later on. Second, we perform a local convex decomposition of $C^a_{\mathrm{GLC}}$ (see Fig.6 left) based on the center points and normals of the GLCs as described by [Eri05]. As there are usually not more than six active constraints, we can employ a naive brute force approach without affecting the overall performance significantly.

Finally, we need to integrate this contact model into our multi-rate simulation scheme. As described, $C_{\mathrm{GLC}}$ is setup in L3 and stays the same in multiple iterations of *HR*. However, the active constraint set $C^a_{\mathrm{GLC}}$ changes during these iterations depending on the state of the simulated objects. To test if a GLC is currently active, its according constraint values $\boldsymbol{\psi}, \boldsymbol{\tau}_1, \boldsymbol{\tau}_2$ have to be checked in *HR* regarding their bounds (6) and (7). Similar as in (5), this can be done for $\boldsymbol{\psi}$ on a velocity level by:

$$-\mathbf{d}/h \leqslant \mathbf{A}_\psi \boldsymbol{\lambda} + \mathbf{b}_\psi + \Delta\mathbf{b}_\psi \leqslant \mathbf{0}. \tag{9}$$

This also incorporates the changes in $\boldsymbol{\psi}$ through active constraint forces $\boldsymbol{\lambda}$ via the constraint response matrix $\mathbf{A}_\psi$. For the tangential functions $\tau_i$, $i = 1,2$, we proceed similarly as for $\boldsymbol{\psi}$ and create linear approximations:

$$\boldsymbol{\tau} \approx \boldsymbol{\tau}_{t_i} + h\mathbf{H}(\mathbf{v}_{t_i} + d\mathbf{v}), \tag{10}$$

where $\boldsymbol{\tau}$ denotes the resulting values of the functions in a stacked form, $\boldsymbol{\tau}_{t_i}$ are the corresponding values of the functions at time $t_i$, and $\mathbf{H}_\tau = \partial\boldsymbol{\tau}/\partial\mathbf{v}$. Furthermore, we use analog to (9):

$$-\mathbf{s}/h \leqslant \underbrace{\mathbf{H}\mathbf{A}^{-1}\mathbf{J}^T}_{\mathbf{A}_\tau}\boldsymbol{\lambda} + \underbrace{\boldsymbol{\tau}_{t_i}/h}_{\mathbf{b}_\tau} + \underbrace{\mathbf{H}(\mathbf{v}_{t_i} + h\mathbf{A}^{-1}\mathbf{f}_{t_i})}_{\Delta\mathbf{b}_\tau} \leqslant \mathbf{s}/h \tag{11}$$

to test $\boldsymbol{\tau}$ in *HR* against the bounds $\mathbf{s}$. $\mathbf{A}_\tau$ approximates how the tangential functions $\boldsymbol{\tau}$ change for a set of constraint forces $\boldsymbol{\lambda}$. In this, $\mathbf{A}_\tau$ reflects the complete dynamics of the involved objects under consideration of the temporal integration scheme. In our multi-rate simulation, $\mathbf{A}_\tau$ is additionally computed in step L4 and can be seen as part of the intermediate contact representation. $\mathbf{b}_\tau$ and $\Delta\mathbf{b}_\tau$ are calculated in step H2.

Moreover, we need to integrate the constraints for the convex areas (8) into our dynamics problem. To this end, we replace the constraint equations from (4) by one complementarity condition of the form:

$$0 \leq \lambda \perp \max_{j=1,..,m} \{\underbrace{\mathbf{J}_j(\mathbf{v}_{t_i} + d\mathbf{v}) + \boldsymbol{\psi}^{t_i}_j/h}_{\psi_j}\} \geqslant 0, \tag{12}$$

for each convex area. Here, the constraint force $\lambda$ is applied to the dynamics problem (3) using the Jacobi matrix $\mathbf{J}_j = \partial\psi_j/\partial\mathbf{v}$ of the constraint $\psi_j$ with the smallest violation. As the resulting constrained dynamics problem does not comply to standard solvers, we present in the following an adequate efficient solving scheme.

### 3.4 Iterative Solver

In this section, we introduce a solver for constrained dynamics problems, which incorporate the previously described contact model. The proposed solver uses an iterative approach and follows a Projected Gauss-Seidel (PGS) scheme. The latter is often applied to solve LCPs in real-time applications. Implementation details on the PGS method itself can be found in [ESHD05]. The solver computes penetration resolving constraint forces $\boldsymbol{\lambda}$ with respect to a given constraint set $C$. The latter can comprise a set of GLCs, $C_{\mathrm{GLC}}$, but also additional constraints, $C_{other}$, modeling friction or joints. As additional input, the solver requires the constraint response matrices $\mathbf{A}_\psi$ and $\mathbf{A}_\tau$ plus the according vectors $\mathbf{b}_\psi$, $\Delta\mathbf{b}_\psi$, $\mathbf{b}_\tau$ and $\Delta\mathbf{b}_\tau$.

At the beginning of each iteration, the active GLC subset $C^a_{\mathrm{GLC}} \subseteq C_{\mathrm{GLC}}$ is computed. To this end, we determine for each GLC whether it is in its active range. This is done under consideration of the previous results of the constraint forces $\boldsymbol{\lambda}$ using (9) and (11). The set $C^a_{\mathrm{GLC}}$ is then decomposed into its convex subsets $C^i_{cs}, i = 1,..,m$ as described in the last section. Afterwards, we resolve the constraint violations employing the PGS strategy; hence, we iterate over all sets and resolve them consecutively. To this end, we first compute, for a subset $C^i_{cs}$, the GLC with the minimal violation. The violation of the GLC is then resolved using the standard PGS method. If there was no violation, the complete subset was non-violated and we proceed to the next subset.

After the GLC subsets are handled, we proceed to the other constraints $C_{other}$. These are resolved using the standard PGS method. The whole process is repeated until an error threshold is reached and the forces converge (see sec.5). The complete solving scheme is summarized in the following and substitutes the solver in step H3 of the multi-rate simulation.

```
Iterate until convergence and error threshold is reached
    - Compute active GLC Set $C^a_{\mathrm{GLC}}$ based on $\boldsymbol{\lambda}$
    - Set $\lambda_{c_k} = 0$ for all $c_k \in C_{\mathrm{GLC}} \setminus C^a_{\mathrm{GLC}}$
    - Compute convex sub sets $C^i_{cs} \subseteq C^a_{\mathrm{GLC}}, i = 1,..,m$
    - Iterate over all $C^i_{cs}, i = 1,..,m$
        - Compute $c_j \in C^i_{cs}$ with minimal violation
        - If violated, compute resolving $\lambda_{c_j}$ for $c_j$
        - Set $\lambda_{c_k}$ with $c_k \in C^i_{cs} \setminus \{c_j\}$ to zero
    - Iterate over all $c_l \in C_{other}$
        - Compute resolving $\lambda_{c_l}$ for $c_l$
```

### 3.5 Approximation Quality Awareness & Trust Regions

Using the approach described above, we can model large contact spaces including convex areas and incorporate them into our multi-rate simulation. Nevertheless, when we solve the dynamics problem, we are not aware in which area such a contact space description is valid. Therefore, a solution to the problem might lead to movements which go out of the modeled area. In consequence, the movements might result in penetrations of the unmodeled parts of the geometry.

To tackle this problem, we add an additional requirement on the allowed solutions. Following the idea of trust region-based optimization [Yua00], we include a constraint to our dynamics problem, which restricts the results to the region where the employed approximations are trusted. In our case, this region corresponds to the area that is accurately modeled by the current constraint set. To this end, we add in step L3 a box-constraint $c_{box}$ to our constraint set $C$. We use the largest possible box that fits into the area covered by the proximity queries. Thereby, it conservatively restricts the solution of the dynamics problem to the area covered by the GLCs. The constraint is defined via:

$$-\frac{r_i}{2} \leqslant \gamma_i(\mathbf{x},\mathbf{v},t) \leqslant \frac{r_i}{2}, \text{with } i=1,2,3 \tag{13}$$

where $\gamma_i$ are functions that map to the box-constraint-coordinates and $r_i$ define the extend of the box. To integrate this constraints into our simulation, we linearize $\gamma_i$ and add according constraint inequalities into the dynamics equations (3) and (4). This is done similarly to the other constraints and we therefore spare the details. As result, the solution is not allowed to leave the region modeled by the contact set.

Furthermore, another problem arises during the iterative solving process from the limited range in which an *individual* constraint should be active. To understand the problem, we take a closer look on how the proposed solver works and interpret the iterative process geometrically.

The algorithm starts with $\boldsymbol{\lambda}_0 = 0$, which means there are zero constraint forces. Considering our simulation scheme, that would lead to an unconstrained motion of the objects. This would result in state:

$$\mathbf{x}_{\text{free}} = \mathbf{x}_{\text{old}} + \Delta\mathbf{x}_{\text{free}}, \tag{14}$$

where $\mathbf{x}_{\text{old}}$ is the result of the last time-step and $\Delta\mathbf{x}_{\text{free}}$ is the motion without constraints. Then, during the successive solving process a sequence of constraint forces $\boldsymbol{\lambda}_i$ are computed. These correspond to corrective movements:

$$\Delta\mathbf{x}_{\lambda_i} = h\mathbf{A}^{-1}\mathbf{J}^T\boldsymbol{\lambda}_i \tag{15}$$

of the objects and define an according sequence of tentative states $\mathbf{x}_i = \mathbf{x}_{\text{free}} + \Delta\mathbf{x}_{\lambda_i}$. In this process, the computation of a set of constraint forces $\boldsymbol{\lambda}_i$ at step $i$ is based on the previous state $\mathbf{x}_{i-1}$; the latter is used to evaluate the constraints and test if an GLC is in its active range.

We now analyze two problematic contact scenarios. In the first one, there are two opposing surfaces and an unconstrained motion, which would tunnel from one side to the other (see Fig.4 left). Therefore, our solving process starts with a state $\mathbf{x}_{\text{free}}$, which has already passed GLC1 and GLC2. Consequently, no constraint will be active in the first iteration and the constraint forces $\boldsymbol{\lambda}$ will be zero. Therefore, no corrective motion will be applied and the result of the step will be $\mathbf{x}_{\text{free}}$ again. The next iteration will then start under the same conditions an come to the same result. In consequence, the object tunnels through the object. A first naive approach to resolve the problem would be to start the process with $\mathbf{x}_{\text{old}}$. This fails in the second problematic scenario (see Fig.4 right). Here, starting from $\mathbf{x}_{\text{old}}$, only the GLC1 will be active in the first iteration. This results in the tentative state $\mathbf{x}_{\text{GLC1}}$ which jumps over the GLC2. Hence, no constraint will be active in the next iteration, which results in a tunneling through the geometry.

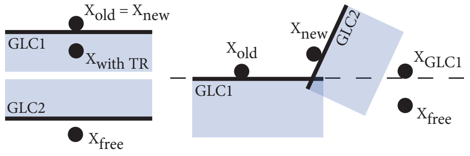

**Figure 4:** *(Left) Free motion overleaps both GLCs. (Right) GLC1 resolution results in overleap of GLC2.*

In summary, issues arise when the sequence of tentative states $\mathbf{x}_i$, computed during the solving process, comprises too large movements. More precisely, the problem occurs when the movement from one state to the next, $\Delta\mathbf{x}_i = \mathbf{x}_i - \mathbf{x}_{i-1}$, skips the active region of one of the constraints.

To resolve this problem, we again follow the idea of a trust region. Therefore, we add a second smaller trust region into our solving process that reflects the range of the *individual* constraints. It is used in each iteration to limit the change in the tentative state $d\mathbf{x}_i$. In doing so, we want to ensure that we can recover in the next state from penetrations and do not miss out any GLC. The trust region is again implemented as a box-constraint and its size is chosen dynamically with the size of the smallest GLC in the current contact set $C_{\text{GLC}}$. For the first iteration of our solver, we set the center of the box to $\mathbf{x}_{\text{old}}$. After each iteration, the constraint is applied similarly to the other constraints using the PGS method. In doing so, the final tentative state $\mathbf{x}_i$ after the iteration lies inside the currently trusted region. In the next iteration of the solver, a new active set of GLCs will be determined based on $\mathbf{x}_i$. Therefore, we can trust, that the region around $\mathbf{x}_i$ is modeled correctly by the set. This allows us to move the center of the box-constraint to $\mathbf{x}_i$ so that further movements are possible during the iteration.

As the trust region size is chosen to be smaller than the extent of the smallest GLC, the overall movement $\Delta\mathbf{x}_i$ created by an iteration cannot skip over any constraint. In consequence, the trust region resolves the problems described above. However, the approach restricts the movements that can be done in each iteration. Therefore, it can increase the number of iterations that have to be done to find the final result. Nevertheless, this was not an issue in our experiments as the process always converged to a solution within less than 1ms (see sec.5.2).

## 4 Simulation with Complex Deformable Objects

As announced, we want to facilitate, with our contact model, also an accurate haptic rendering of deformable environments. However, the multi-rate simulation described in section 3.2 performs in step H3 a temporal integration of *all* objects in the scene, which can be too slow for complex deformable objects. To this end, we integrate our contact model into the asynchronous simulation approach described by [DPD*13].

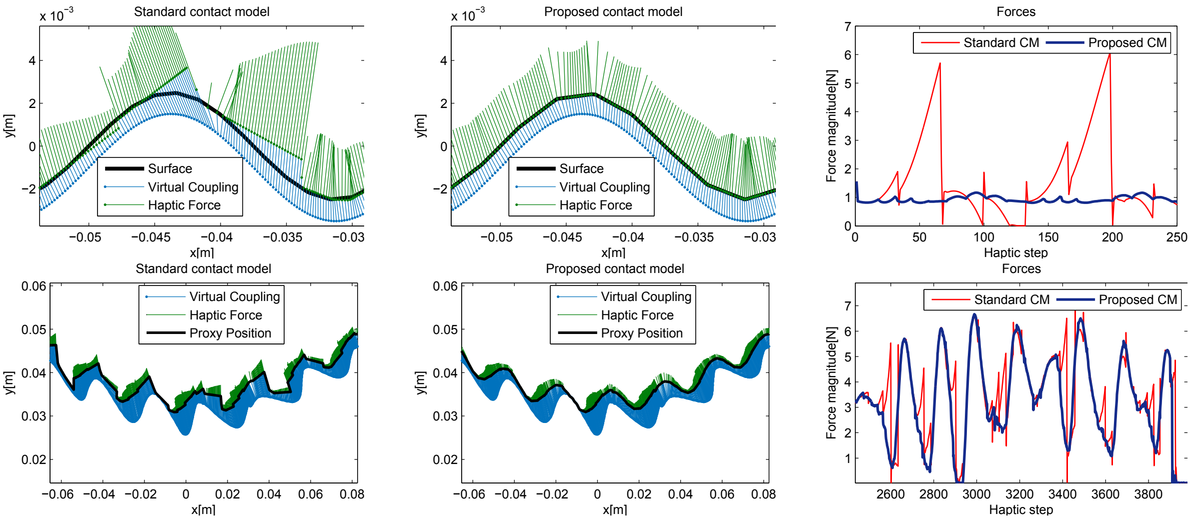

**Figure 5:** *The top row gives the results for the static scenario and the bottom row for the deformable. The left four plots show outtakes of the simulation trajectories and forces. The input positions are indicated by the blue dots and the proxy positions by green dots. Input and proxy positions from the same time-step are connected by a blue line. The green lines reflect the forces rendered for each proxy position. To this end, the x- and y-components of the forces are plotted. Although the scaling is arbitrary, the lines provide information about how continuous the forces are with respect to their magnitude and direction. The absolute magnitudes of the forces are reported in the right two plots.*

This approach also performs proximity queries on a lower rate to create a mechanical intermediate contact representation, but instead of performing a temporal integration of all objects on the haptic rate, like we do, the simulation of complex deformable objects is done *solely* on the lower rate. The collision response between the objects simulated on the different rates is calculated using the intermediate contact representation. The representation is based on the classic contact model as described in the introduction and employs a linear complementarity problem similar to (5) to reflect the dynamic behavior of the objects. Therefore, [DPD*13] also compute a constraint response matrix $\mathbf{A}_\psi$ and a corresponding $\mathbf{b}_\psi$ on the low rate.

Due to the usage of the classic contact model, the simulation suffers from the problems described in the introduction. Therefore, we replace this classic model with our proposed accurate model. To this end, we need to compute the constraint response matrix $\mathbf{A}_\tau$ additionally to $\mathbf{A}_\psi$ on the low rate and use our proposed solver for the dynamics problems including our contact model. As result, the collision response between the objects simulated on the different rates is based on our accurate contact model with all described advantages. We emphasize that no specific method for the modeling of deformable objects is required, it just has to be compatible to the backward Euler-based temporal integration. For further details on the asynchronous simulation approach, we need to refer the reader to [DPD*13].

## 5 Results and Discussion

In this section, we will evaluate the presented contact model in several scenarios. To this end, we will compare two conditions: first, the described multi-rate simulations utilizing the proposed contact model (PCM), and second, the same multi-rate simulations but using the standard contact model (SCM) based on linear inequalities (see sec.3.2). For PCM, the threshold for the proximity queries was set to a radius of 20 mm around the proxy. In case of SCM, this would create the locking effects described in the introduction and create significant artifacts in the haptic rendering. Therefore, we used only the closest delivered contact in this condition. We first demonstrate the benefits of the PCM in a simple static and a simple deformable scenario. Afterwards, we show that the advantages are still valid for more complex scenes. These are also used for performance measurements to show the real-time capabilities of the presented techniques.

Throughout the evaluation, we employed a virtual coupling approach. The mass of the simulated point-proxy was set to 1 g while the spring-damper had a stiffness of 1000 N/m and a damping constant of 0.8 Ns/m. The simulation of friction is done similar to [KK14]. Here, a standard constraint-based approach is used to model the Coulomb's friction cone via a linear approximation. The friction coefficient was set to 0.1. For the simulation of the deformable objects, we employed a hexahedral-FEM. A linear elastic constitutive law was used to create the mechanical equations, which were updated during the simulation using the co-rotational method. In all tests, we employed a multi-rate simulation approach. To this end, the haptic loop used a time-step of 1 ms for the temporal integration and was run with an update rate of 1000 Hz. For the slow loop, we applied a time-step of 33 ms and an update rate of ∼30 Hz. The latter corresponds to a typical update rate, often used in multi-rate applications for the lower rate. The complete simulation is based on in-house developed software frameworks.

Except for the first test case, we recorded trajectories and replayed them for the measurements. For better comparability, we used the same trajectory for both conditions, PCM and SCM. The recordings were made with PCM using a Phantom Premium 1.5 connected to a common workstation with a 4 core 2.53 GHz Intel Xeon CPU with 12 GB RAM.

### 5.1 Haptic Rendering Quality

In this section, we demonstrate the increased haptic rendering quality of PCM compared to SCM for geometries with convex and concave areas. Our first test case uses a static surface created via a sampling of a sine function with an amplitude of 5 mm and a wavelength of 25 mm. For an easier analysis, the haptic trajectory was created using a similar sinus function as for the surface but with an offset of 1 mm. Therefore, the haptic input is quickly swiped from left to right always slightly below the surface. Figure 5 shows outtakes of the side view on the surface in the z-direction. It furthermore depicts the resulting input and proxy trajectories including their coupling. Additionally, the x- an y-components of the haptic forces are indicated for each proxy position. In the top-left plot, the results of the SCM show that the proxy trajectory regularly moves away from the surface. This happens due to the usage of constraints which are actually already out-dated. Then, the proxy snaps back when an updated constraint set arrives from the low rate. In consequence, the according haptic forces increase unnaturally and are non-smooth in direction and magnitude. The plots for the PCM show a proxy trajectory that always stays on the surface. Therefore, the forces are smoother with respect to their magnitude and direction. The slight bumps reflect the sampling of the rendered geometry and get smaller with a higher resolution of the surface geometry.

To evaluate our proposed contact model in combination with deformable objects, we embedded the above described surface into a deformable mesh (see Fig.7). The latter was fixed on the left and the right side. Here, we recorded a haptic trajectory of a fast swipe over the surface from left to right. The results are given in the bottom row of Figure 5. The plots show that in case of SCM the proxy trajectory does not reflect the sinus shape of the surface. Furthermore, the forces are discontinuous and contain large jumps with respect to magnitude and direction. In case of PCM, the proxy trajectory follows the sinus shape of the surface. This results in rather smooth forces without abrupt changes.

We also performed tests with two less synthetic use cases. The first one employs a "rocker arm" and the second a "Fertility statue" geometry (see Fig.6). The scenarios are more challenging as more complex contact configurations occur. The transition from the statue to its base, for instance, contains areas which are convex in one direction and concave in another. Such a contact situation is also shown on the left of Figure 6; it also depicts the GLC set, which is active at the end of the haptic simulation step. The correct convex decomposition of the set by our approach is illustrated via a color coding.

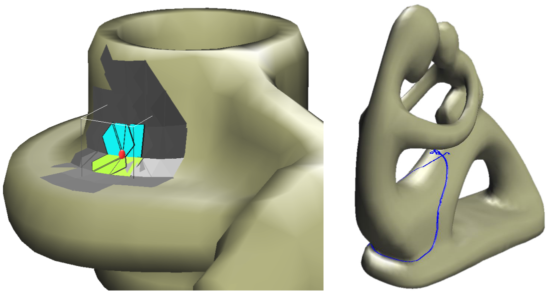

**Figure 6:** *(Left) The haptic proxy, shown as a red dot, in contact with a convex-concave surface. The GLCs in the current constraint set are visualized in gray and the ones which are in their active region are colorized. GLCs with the same color belong to the same convex set. (Right) Fertility statue geometry including the tested haptic trajectory in blue.*

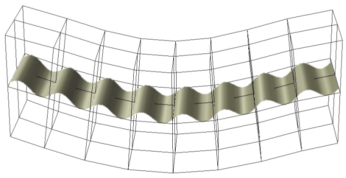

**Figure 7:** *Surface generated from sine function embedded in a deformable mesh, which is fixed on the left and right side.*

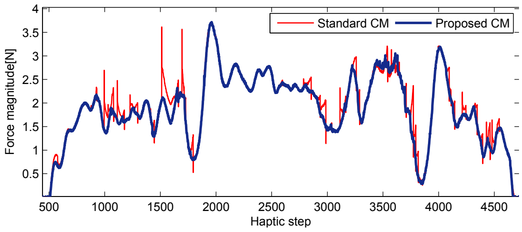

**Figure 8:** *Haptic forces for complex rigid scenario.*

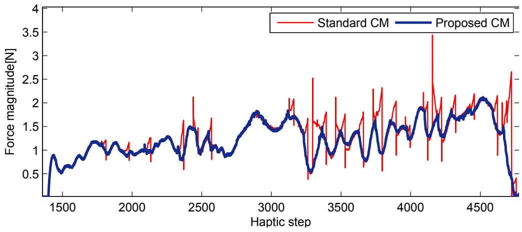

**Figure 9:** *Haptic forces for complex deformable scenario.*

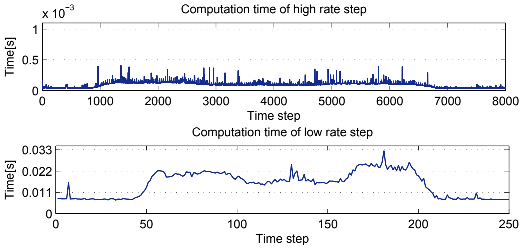

**Figure 10:** *Computation times for complex deformable scene.*

We evaluated the quality of the haptic forces created by PCM and SCM with a rigid and a deformable version of the "Fertility statue" geometry. The used trajectory is shown as a blue line on the right of Figure 6. The haptic forces for the rigid version are reported in Figure 8 and the ones for the deformable version are given in Figure 9. In both cases, rigid and deformable, the PCM generated rather smooth forces while the ones generated by SCM contain jumps and hard kinks.

### 5.2 Performance Measurements

In the following, we demonstrate the real-time capabilities of our approach. To this end, we measured the key performance data for our most complex test case, the deformable "Fertility statue". During the test, the number of GLCs in the constraint set $C_{\text{GLC}}$ usually ranged between 50 and 100. Furthermore, the constraint set included two additional constraints for each GLC to simulate friction. However, the solver always converged with a constraint error threshold of $0.1 \cdot 10^{-10}$ within less than 30 iterations and 0.5 ms. The computation time for a *LR* step had a max value of 32.3 ms and a mean of 15.2 ms. Therefore, the aimed update rate of 30 Hz could be maintained. The computation time for a *HR* step had a max value of 0.41 ms and a mean of 0.09 ms. Consequently, the critical haptic update rate of 1 kHz could be easily achieved. Figure 10 shows the computation times measured for both rates.

## 6 Conclusion

In this paper, we presented a novel contact model for multi-rate single-point haptic rendering of static and deformable environments. The main contribution is in proposing a contact model which facilitates the creation of accurate intermediate contact representations. These allow a correct treatment of concave and convex contact scenarios on a haptic simulation rate, while reflecting the full dynamics of the involved objects including compliant behavior of deformable objects. Other constraints modeling, for instance, friction, can be easily integrated into the model. The resulting constrained dynamics problems are robustly solved using a novel efficient solving scheme utilizing techniques from trust region-based optimization. The quality of the haptic rendering is demonstrated for static and deformable scenarios with different complexities. Furthermore, we prove that the presented approach is fast enough for the haptic rendering of complex scenes.

The proposed methods allow an accurate haptic rendering with a single-point haptic proxy. While we believe that this already has a wide application field, extending it to more complex proxies should be one of the next steps. However, scaling the approach to multiple contact points would increase the size of the intermediate representation and its computation time. Therefore, future work will aim at providing efficient methods to deal with this problem.

### Acknowledgments

This project has received funding from the European Union's 7th Framework Programme for research, technological development and demonstration under grant agreement 610425. Furthermore, we thank AIM@SHAPE for providing the data sets used in the evaluation.

### References

[AH99] Adams R., Hannaford B.: Stable haptic interaction with virtual environments. *IEEE Trans. on Robotics and Automation 15*, 3 (June 1999), 465–474. 3

[AP97] Anitescu M., Potra F. A.: Formulating Dynamic Multi-Rigid-Body Contact Problems with Friction as Solvable Linear Complementarity Problems. *Nonlinear Dynamics* (1997). 2

[BJ08] Barbic J., James D.: Six-DoF Haptic Rendering of Contact Between Geometrically Complex Reduced Deformable Models. *Trans. on Haptics 1*, 1 (2008). 3

[BS13] Blevins N. H., Salisbury K.: Deformable Haptic Rendering for Volumetric Medical Image Data. *Interfaces* (2013). 3

[DD06] Duriez C., Dubois F.: Realistic haptic rendering of interacting deformable objects in virtual environments. *IEEE TVCG 12* (2006), 36–47. 2, 3

[DDCB01] Debunne G., Desbrun M., Cani M.-P., Barr A. H.: Dynamic real-time deformations using space & time adaptive sampling. *ACM SIGGRAPH* (2001), 31–36. 3

[DPD*13] Dervaux F., Peterlik I., Dequidt J., Cotin S., Duriez C.: Haptic rendering of interacting dynamic deformable objects simulated in real-time at different frequencies. In *IEEE Int. Conf. Intelli. Rob. and Syst.* (2013), pp. 2010–2016. 2, 3, 4, 7, 8

[Eri05] Ericson C.: Dealing with "Nondecomposable" Concave Geometry. In *Real-time collision detection*. 2005, pp. 506–507. 6

[ESHD05] Erleben K., Sporring J., Henriksen K., Dohlman K.: *Physics-based Animation*. Charles River Media, Inc., June 2005. 1, 2, 3, 4, 6

[KK14] Knott T., Kuhlen T.: Geometrically limited constraints for physics-based haptic rendering. In *Haptics: Neuroscience, Devices, Modeling, and Applications*, vol. 8619 of *LNCS*. 2014, pp. 343–351. 2, 3, 5, 8

[NT10] Nguyen B., Trinkle J.: Modeling non-convex configuration space using linear complementarity problems. *Int. Conf. on Robotics and Automation* (2010). 1, 2, 3, 5

[OL06] Otaduy M., Lin M.: A modular haptic rendering algorithm for stable and transparent 6-DOF manipulation. *IEEE Trans. on Robotics 22*, 4 (Aug. 2006), 751–762. 2, 3

[OL08] Otaduy M., Lin M.: *Haptic Rendering: Foundations, Algorithms, and Applications*. AK Peters, 2008. 3, 4

[OTSG09] Otaduy M. A., Tamstorf R., Steinemann D., Gross M.: Implicit Contact Handling for Deformable Objects. *Comp. Graphics Forum 28* (Apr. 2009), 559–568. 3

[PDC11] Peterlik I., Duriez C., Cotin S.: Asynchronous haptic simulation of contacting deformable objects with variable stiffness. *Intelligent Robots and Systems* (2011), 2608–2613. 3

[WM03] Wan M., McNeely W.: Quasi-static approximation for 6 degrees-of-freedom haptic rendering. In *IEEE Vis. 2003* (Oct. 2003), IEEE Computer Society, p. 34. 3

[WSL14] Wang D., Shi Y., Liu S.: Haptic simulation of organ deformation and hybrid contacts in dental operations. *IEEE Trans. on Haptics 7*, 1 (2014), 48–60. 3

[Yua00] Yuan Y.-x.: A review of trust region algorithms for optimization. *ICIAM* (2000), 271–282. 7

[ZS95] Zilles C. B., Salisbury J. K.: A constraint-based god-object method for haptic display. *International Conference on Intelligent Robots and Systems* (1995). 3