



HAL
open science

Aggregate Constraints for Virtual Manipulation with Soft Fingers

Anthony Talvas, Maud Marchal, Christian Duriez, Miguel A Otaduy

► **To cite this version:**

Anthony Talvas, Maud Marchal, Christian Duriez, Miguel A Otaduy. Aggregate Constraints for Virtual Manipulation with Soft Fingers. *IEEE Transactions on Visualization and Computer Graphics*, 2015, 21 (4), pp.9. 10.1109/TVCG.2015.2391863 . hal-04813516

HAL Id: hal-04813516

<https://inria.hal.science/hal-04813516v1>

Submitted on 2 Dec 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Aggregate Constraints for Virtual Manipulation with Soft Fingers

Anthony Talvas, Maud Marchal, Christian Duriez and Miguel A. Otaduy

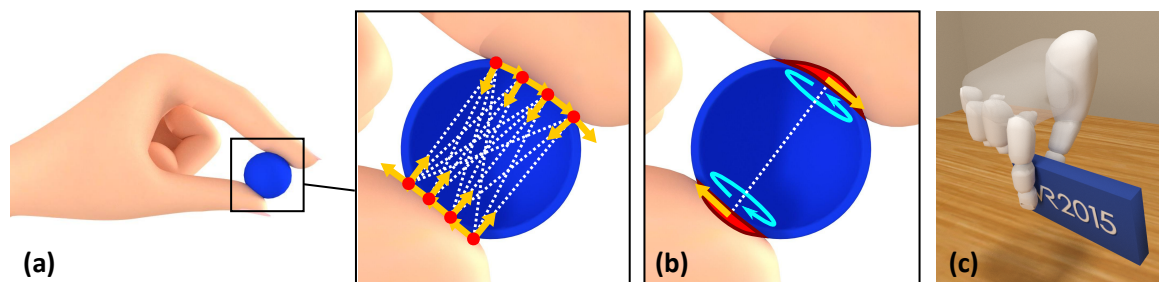


Fig. 1. (a) Interaction with deformable fingers generates many interconnected contact points which are expensive to solve with friction. (b) Our approach aggregates contact constraints per phalanx with torsional friction. (c) The subsequent increase in performance allows real time dexterous manipulation of virtual objects using soft fingers.

Abstract—Interactive dexterous manipulation of virtual objects remains a complex challenge that requires both appropriate hand models and accurate physically-based simulation of interactions. In this paper, we propose an approach based on novel aggregate constraints for simulating dexterous grasping using soft fingers. Our approach aims at improving the computation of contact mechanics when many contact points are involved, by aggregating the multiple contact constraints into a minimal set of constraints. We also introduce a method for non-uniform pressure distribution over the contact surface, to adapt the response when touching sharp edges. We use the Coulomb-Contensou friction model to efficiently simulate tangential and torsional friction. We show through different use cases that our aggregate constraint formulation is well-suited for simulating interactively dexterous manipulation of virtual objects through soft fingers, and efficiently reduces the computation time of constraint solving.

Index Terms—Physically-based simulation, grasping, dexterous manipulation, constraint computation, virtual deformable hand.

1 INTRODUCTION

In everyday life, we constantly use our hands to interact with the world around us. In the interaction with virtual environments, however, most of the existing work has focused on tool-based interaction, such as in medical training or virtual prototyping. The reason is that rigid tools and their interactions are far simpler to simulate in a physically plausible manner, making tool-based interaction a more tractable problem. Unfortunately, in many practical tasks we manipulate and explore the world directly with our hands, which calls for virtual reality solutions for direct hand interaction. The use of virtual hands that mimic the behavior of our own hands provides a more natural and immersive way to interact with virtual environments.

Grasping and dexterous virtual object manipulation are however complex tasks that require appropriate models of hand and contact mechanics to be simulated in real time. A popular and computationally efficient approach in virtual grasping is to model the hand as an articulated body made of rigid phalanges. However, the human fingerpad is naturally soft, and produces large contact areas even when applying small forces, thus increasing grasping stability. Rigid fingerpad models complicate grasping due to small contact areas, notably when picking objects from sharp edges, and also require larger forces, often leading to sticking friction artifacts.

Modeling deformable fingerpads offer a solution to better reproduce friction between a hand model and virtual objects, but simulating soft fingers has its own difficulties as well. The softness of the fingerpads leads to a fast expansion of the contact surface at initial contact, which makes the resulting grasping more stable, but also increases the number of contact points, making computation more intensive. Moreover, precise friction models need to be computed at each contact point, which further increases the complexity of the system to solve. Unfortunately, solvers for frictional contact mechanics with deformable objects scale poorly with the resolution of the objects and the number of contacts.

Allard *et al.* introduced volume contact constraints for improving the performance of contact handling with deformable objects [1]. The method used volumetric collision detection in order to formulate contact constraints as penetration *volumes* between objects instead of penetration *distances* over several contact points. In addition, the number of constraints was not dependent on the resolution of the collision meshes, but instead on a regular grid which determined how the penetration volumes were divided. Despite its benefits in terms of efficiency, this approach suffers from two important limitations for grasping simulation. The first one is that pressure is uniform at all contact points aggregated into the same volume constraint, ignoring differences between contact against a flat surface, a curved surface, or a sharp edge. The second one is that their aggregate friction constraints act only on linear velocities, not accounting for twists over the contact surface. In order to retain some pressure distribution as well as torsional friction between objects, the number of volume constraints needs to be increased, which in turn decreases the efficiency of the method.

In this paper, we introduce novel aggregate constraints for simulation of dexterous grasping. We formulate aggregate constraints for each phalanx, but we augment the volume contact constraint formulation introduced by Allard *et al.* in multiple ways, aimed at improving the computation of hand and contact mechanics during grasping,

- Anthony Talvas and Maud Marchal are with Inria Rennes and INSA Rennes, France. E-mail: {anthony.talvas,maud.marchal}@inria.fr.
- Christian Duriez is with Inria Lille, France. E-mail: christian.duriez@inria.fr.
- Miguel A. Otaduy is with Universidad Rey Juan Carlos, Spain. E-mail: miguel.otaduy@urjc.es.

Manuscript received 18 Sept. 2014; accepted 10 Jan. 2015. Date of Publication 20 Jan. 2015; date of current version 23 Mar. 2015.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org.

while maintaining efficient resolution of penetrations and realistic friction. Our first contribution is a novel formulation of volume contact constraints from point contact-based collision detection, presented in Section 3.2. Our second contribution is a method to compute a non-uniform pressure distribution within aggregate constraints, introduced in Section 3.3. It adapts to the local geometry of the objects in contact, capturing the grasping differences between flat and sharp surfaces. Our third contribution is the addition of torsional friction to aggregate volume contact constraints, detailed in Section 3.4. We use the Coulomb-Contensou friction model, developed by Contensou [9], and integrated into constraint-based contact handling methods by Leine and Glocker [28]. We incorporate this friction model to volume contact constraints, derive an aggregate formulation, and demonstrate its practical success for stable grasping through pinching with minimal added computational cost. Our fourth contribution is a deformable hand model for manipulation of virtual objects using our constraint method, described in Section 4. We evaluate the method on several interaction scenarios in Section 5, showing that aggregate constraints are well suited for grasping simulation because the soft tissue under each phalanx offers a natural domain for constraint aggregation.

2 RELATED WORK

2.1 Grasping and Hand Simulation

Dexterous manipulation of virtual objects remains a complex challenge when interacting with virtual hands. A few heuristics-based methods were proposed to allow stable grasping. Zachmann *et al.* [47] used the distribution of contacts among finger phalanges, thumb phalanges and palm to discriminate between push, precision grasp and power grasp. Grasps were then maintained by constraining the relative position and orientation of the object with respect to the palm. Moehring and Froehlich [32] used a similar approach, but instead using ray tests between distal phalanges to determine the fingers that participate in the grasping, with a hierarchy that prioritizes more stable grasps, and a simplified interaction was used during grasping as well. Grasping can also be detected through the use of *grasp pairs*, which are defined as pairs of fingers contacting with an object for which the line between both contacts is included in both friction cones [19, 33]. The motion of manipulated objects is then defined by the motion of those grasp pairs. While these methods allow stable grasps of virtual objects, they have to sacrifice physical correctness in the process, often leading to unrealistic contact between hand and object and sometimes ignoring object-object collision. Furthermore, they may have to approximate the hand motions of the user as well.

One of the most straightforward methods to perform more physically-accurate hand interaction is the use of articulated hand models based on rigid bodies, where palm and phalanges are represented by interconnected rigid bodies. Borst and Indugula [7] introduced the use of such models, by using linear and torsional spring-dampers between the simulated palm and phalanges and their tracked configurations. This effectively allowed grasping of objects, though it required high friction coefficients which could result in excessive sticking. This method was shown to be sufficiently efficient to allow dexterous interaction with haptic feedback on the palm and fingers [39]. Jacobs *et al.* [23] improved on this model by building upon the 6 degree-of-freedom (DOF) god-object method [37] to build a fully constrained *God-hand* model, which solves the interdependencies between the multiple rigid bodies of the hand using Gauss's principle of least constraints. Several studies have focused on synthesizing plausible grasping motions with rigid body hands [27, 29, 46]. Overall, rigid body models are computationally efficient, but they do not account for skin deformation under contact, which is not only necessary for a more realistic-looking interaction and motion [24], but also to simulate friction at contact surfaces more accurately. A *god-finger* method was proposed to render contact surfaces that emulate skin deformation under contact with rigid body models [45], however it only approximates the behavior of deformable fingers in contact and does not produce any visual deformation.

The hand is a complex structure made of a skeleton, muscles, tendons and skin, all with their own properties, making it challenging

to accurately simulate the deformation of the hand as a whole during motion and contact. While it is possible to simulate the biomechanical behavior of the hand tendons and muscles [44], such methods do not meet the performance requirements for interactive simulation and are more suited for offline rendering. Finite element methods [48, 14, 42, 21] have been used in several works as an effective way to simulate finger deformation. Gourret *et al.* proposed a model for hand animation made of an FEM-based deformable hand with an underlying rigid skeleton, allowing more realistic contact with deformable objects [18]. The rigid skeleton was made of realistic bones, so as to better simulate the restriction of flesh deformation due to bones. Other deformable hand models use conjointly a rigid hand to simulate the back of the hand, which is the least deformable part, and soft bodies to simulate the finger pads, either using FEM [43] or lattice shape matching with adaptive stiffness [22]. Garre *et al.* proposed another way to couple an FEM-based hand with a rigid body hand, by using springs between nodes of the deformable hand and rest positions skinned from the rigid body configuration [17]. These methods effectively allow dexterous manipulation of objects with fairly realistic deformation under contact. Contact is also more realistic thanks to the larger contact surfaces compared to rigid body models. However, the number of contacts could become very large when taking into account different fingers, preventing real-time performances.

2.2 Resolution of Contacts with Soft Bodies

Constraint-based contact solving methods attempt to precisely solve all penetrations and friction between objects in contact by solving a linear complementarity problem (LCP) [4]. The resulting system can be solved either in one step using a direct solver [2, 3, 40], or by processing contacts sequentially using an iterative solver [11, 12, 38]. For friction, a pyramid discretization of the friction cone has to be used to keep the problem as an LCP [2, 31], but an exact cone can also be used by using iterative methods to solve the resulting non-linear complementarity problem (NLCP) [11]. Methods have also been proposed to more specifically handle contact constraints with articulated bodies, either rigid [10] or deformable [16].

Contact scenarios involving deformable objects tend to produce large numbers of contact points due to the molding of the objects they are colliding with. This, in turn, may cause computational problems when attempting to solve LCPs, notably when a large number of contacts affects a small number of degrees of freedom in rigid-deformable contacts [30]. This raises the need for methods to efficiently handle such complex scenarios. Separating constraint sets involving rigid-rigid, deformable-deformable and rigid-deformable contacts was shown to be an efficient way to solve large LCPs involving all these types of contacts simultaneously [30].

Another approach for speeding up the resolution of contacts is the use of contact reduction or clustering methods. They are commonly used in physics engines for video games [34], but were also shown to be suitable for haptic interaction [26]. Some preprocessing can be applied at the collision detection level to ensure that the smallest amount of redundant contacts gets generated in the first place. Then, contact clustering methods can be used to further reduce the number of contacts, for instance using the Euclidean distance between contacts, keeping a single weighted average contact point and contact normal for each cluster, with the maximum penetration depth of the contact points within the cluster [26]. Such methods are best suited for rigid body interaction however, as deformable bodies require a fine sampling over the entire contact surface.

An alternative method consists in reducing the number of constraints generated during contact by formulating contact as *volume constraints*, as opposed to *point constraints* [1]. Contact constraints are no longer defined as numerous penetration distances, but rather as a much reduced number of penetration volumes to be solved, number which is defined by the arbitrary resolution of a regular grid. This method also gathers friction frames at the different contact points into one frame per volume constraint. However, while the use of multiple friction frames allows to handle torsional friction over contact surfaces without an explicit consideration of that phenomenon, the gathering of

friction forces ignores rotational effects within each volume constraint. Our approach, instead, retains torsional friction without requiring multiple friction frames.

A few models exist for simulating torsional friction from a single contact. Barbagli *et al.* [5] derived a simple *soft finger* model from the study of human fingertips to achieve two-finger grasping of virtual objects with only one rigid contact per finger. This work was extended to account for the coupling between tangential and torsional friction forces [15] using the concept of limit curve [20]. Ciocarlie *et al.* [8] used approximations of the local geometry of the finger and object in contact for their soft finger model, also achieving grasping with torsional friction using rigid fingers. Finally, the Coulomb-Contensou friction model [9, 28] extends regular Coulomb friction by formulating tangential and torsional friction as a function of sliding and angular velocity, also taking into account the interdependencies between both types of friction. In our work, we adapt the Coulomb-Contensou model to add torsional friction to volume contact constraints.

3 AGGREGATE VOLUME CONTACT CONSTRAINTS

In the simulation of grasping using deformable fingers, fingerpads adapt smoothly to the surfaces of grasped objects. The resulting large contact areas help stabilize grasping, but also induce a large number of point contacts that affect simulation performance, because with typical constrained dynamics methods the number of constraints is proportional to the number of contacts. Instead, we build on volume contact constraints [1] as a more efficient way to handle soft-finger contacts. With our approach, the number of constraints scales with the number of phalanges in contact and is independent from the discretization of collision meshes.

We overcome limitations of previous volume constraint methods by integrating torsional friction within the individual constraints, using the Coulomb-Contensou friction model. We also integrate a non-uniform distribution of forces over the contact surface within the formulation of the constraints, based on the penetration between objects during unconstrained motion. By doing so, we obtain results similar to those obtained with a multi-volume contact constraint approach, while removing the need to divide the contact volumes using a regular grid, thus further reducing the number of constraints involved to a total of 4 per phalanx. This section describes the formulation of these constraints.

3.1 Integration of Constrained Dynamics

Before describing our approach to volume contact constraints, we first formulate the dynamics of a physical system with generic constraints. According to Newton's second law, the dynamics of a body within the simulation follow:

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{f}(\mathbf{q}, \mathbf{v}) + \mathbf{f}_{\text{ex}}, \quad (1)$$

where \mathbf{q} is the vector of degrees of freedom, $\mathbf{v} = \dot{\mathbf{q}}$ is the vector of velocities and \mathbf{M} is the mass matrix. \mathbf{f} provides the internal forces and \mathbf{f}_{ex} the external forces (due to contact and other constraints). Using implicit Euler integration with linearized forces, and given a time step h and current state $(\mathbf{q}_0, \mathbf{v}_0)$, the velocity increment on a time step is obtained by solving the linear system:

$$\underbrace{\left(\mathbf{M} + h \frac{\partial \mathbf{f}}{\partial \mathbf{v}} + h^2 \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \right)}_{\mathbf{A}} d\mathbf{v} = \underbrace{h \mathbf{f}(\mathbf{q}_0, \mathbf{v}_0) + h^2 \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \mathbf{v}_0 + h \mathbf{f}_{\text{ex}}}_{\mathbf{b}}. \quad (2)$$

The solution $d\mathbf{v}$ is then used to update the velocity: $\mathbf{v}_{t+h} = \mathbf{v}_t + d\mathbf{v}$, and then the position implicitly: $\mathbf{q}_{t+h} = \mathbf{q}_t + h \mathbf{v}_{t+h}$.

We now consider a deformable phalanx with a triangular surface mesh whose vertices collide with triangles of another body. In order to prevent their interpenetration, as well as to simulate friction at contacts, we solve the previous system for both bodies under contact constraints. These constraints provide the external forces \mathbf{f}_{ex} , which are defined as the product of \mathbb{H} , a matrix of constraint directions, and λ , a vector of constraint force intensities. This leads to the following

linear system for two bodies in contact:

$$\begin{aligned} \mathbf{A}_1 d\mathbf{v}_1 &= \mathbf{b}_1 + h \mathbb{H}_1^T \lambda, \\ \mathbf{A}_2 d\mathbf{v}_2 &= \mathbf{b}_2 + h \mathbb{H}_2^T \lambda. \end{aligned} \quad (3)$$

Let us define as δ a vector of the relative velocities at the contact points expressed in the frames defined by the constraint directions \mathbb{H} . By separating the unconstrained velocities of the bodies, $\mathbf{v}_1^{\text{free}}$ and $\mathbf{v}_2^{\text{free}}$ (i.e. with $\lambda = \mathbf{0}$), from the velocity update due to contact forces, we can express the relative velocities at contact points as:

$$\delta = \underbrace{\mathbb{H}_1 \mathbf{v}_1^{\text{free}} - \mathbb{H}_2 \mathbf{v}_2^{\text{free}}}_{\delta^{\text{free}}} + h \left[\mathbb{H}_1 \mathbf{A}_1^{-1} \mathbb{H}_1^T + \mathbb{H}_2 \mathbf{A}_2^{-1} \mathbb{H}_2^T \right] \lambda. \quad (4)$$

To solve for the velocity update $d\mathbf{v}$ subject to non-penetration constraints of the form $\delta \geq 0$ corresponds to solving a mixed linear complementarity problem (MLCP). In our implementation, we do this by first computing the constraint compliance matrix $\mathbb{H} \mathbf{A}^{-1} \mathbb{H}^T$, and then solving the resulting LCP using Projected Gauss-Seidel (PGS) relaxation. More details on this formulation can be found in [11, 12, 38, 6].

The non-penetration constraints defined above act only on relative velocities, and the final position integration may introduce drift over time. We avoid this through a post-stabilization step where non-penetration constraints are solved on positions after the velocity solve. Given a vector Λ of constraint force intensities for the position solve, the constrained update of body positions can be expressed as the linear system:

$$\begin{aligned} \mathbf{A}_1 d\mathbf{q}_1 &= h \mathbb{H}_1^T \Lambda, \\ \mathbf{A}_2 d\mathbf{q}_2 &= h \mathbb{H}_2^T \Lambda. \end{aligned} \quad (5)$$

Then, non-penetration constraints on relative positions at contact points, δ , can be expressed by summing up the relative positions after the velocity update, δ^{vel} , plus the position update:

$$\delta = \delta^{\text{vel}} + h^2 \left[\mathbb{H}_1 \mathbf{A}_1^{-1} \mathbb{H}_1^T + \mathbb{H}_2 \mathbf{A}_2^{-1} \mathbb{H}_2^T \right] \Lambda \geq 0. \quad (6)$$

We solve the resulting MLCP using the same approach as for the velocity update.

To summarize, the global animation loop for obtaining the velocities \mathbf{v}_{t+h} and positions \mathbf{q}_{t+h} at the next iteration from the current velocities \mathbf{v}_t and positions \mathbf{q}_t thus follows Algorithm 1.

Algorithm 1 Animation loop for velocity and position update with constraints

Input: Current velocities \mathbf{v}_t and positions \mathbf{q}_t

Output: Velocities \mathbf{v}_{t+h} and positions \mathbf{q}_{t+h} at next iteration

```

d $\mathbf{v}^{\text{free}} \leftarrow$  solving (3) with  $\lambda = \mathbf{0}$ 
 $\mathbf{v}_{t+h}^{\text{free}} \leftarrow \mathbf{v}_t + d\mathbf{v}^{\text{free}}$ 
 $\mathbf{q}_{t+h}^{\text{free}} \leftarrow \mathbf{q}_t + h \mathbf{v}_{t+h}^{\text{free}}$ 
Compute collision detection
 $\mathbb{H} \leftarrow$  constraint directions
 $\delta^{\text{free}} \leftarrow$  constraint violations in velocity
 $\lambda \leftarrow$  MLCP solve in velocity using PGS
 $\mathbf{v}_{t+h} \leftarrow \mathbf{v}_{t+h}^{\text{free}} + h \mathbf{A}^{-1} \mathbb{H}^T \lambda$ 
 $\mathbf{q}_{t+h}^{\text{vel}} \leftarrow \mathbf{q}_t + h \mathbf{v}_{t+h}$ 
 $\delta^{\text{vel}} \leftarrow$  constraint violations in position
 $\Lambda \leftarrow$  MLCP solve in position using PGS
 $\mathbf{q}_{t+h} \leftarrow \mathbf{q}_{t+h}^{\text{vel}} + h^2 \mathbf{A}^{-1} \mathbb{H}^T \Lambda$ 

```

3.2 Volume-based Separation Constraints

The constrained dynamics algorithm described in the previous section is valid for generic formulations of non-penetration constraints. Typically, such constraints are formulated individually for each contact

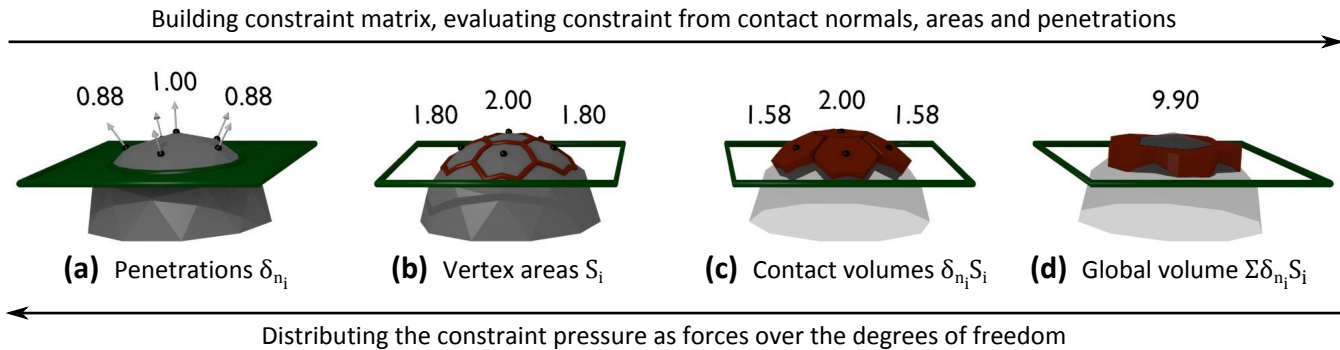


Fig. 2. Formulation of separation contact constraints as an aggregate volume constraint. (a) Two bodies in contact with penetrations evaluated *w.r.t.* contact normals. (b) Areas computed for every contact point from the local geometry. (c) Formulation of contacts as penetration volumes. (d) Summing these volumes to formulate an aggregate constraint, which results in a corrective pressure upon solution.

point, using the separation gap as a metric for non-penetration. In this section, we describe volume constraints instead. In addition, we formulate a single constraint for an entire contact surface, thus dramatically reducing the cost of solving constrained dynamics. Figure 2 shows in a schematic way the process for our constraint formulation.

3.2.1 Constraint matrix

We denote as \mathbb{H}_n the rows of the constraint matrix corresponding to normal directions at contacts, in contrast to tangential directions to be used later in Section 3.4. And we denote as λ_n the normal forces applied by the constraints on the bodies in contact.

In previous work [1], penetration volumes between each pair of bodies were provided directly by specific collision detection methods. Here, we resort to more generic collision detection methods, which provide a set of contact points \mathbf{q}_i (in penetration with the other body) with penetration distances $\delta_{n,i}^{free}$ and penetration distance gradients (*i.e.* transpose of contact normals): $\mathbf{n}_i^T = \partial \delta_{n,i}^{free} / \partial \mathbf{q}_i$. To define a penetration volume from these contact points, we first estimate for each vertex \mathbf{q}_i on the surface of a phalanx a corresponding area S_i . Given a closed surface mesh made of polygons with shared vertices, the area of each polygon P is distributed equally between its vertices. This leads to an estimation of the area of each contact point $S_i = \sum_j 1/n_{P_j} S_{P_j}$, where S_{P_j} denote the areas of the polygons P_j incident on the vertex i , made of n_{P_j} vertices. As a result, we can obtain, for each contact i , the contact volumes and contact volume gradients:

$$\begin{aligned} V_i &= S_i \delta_{n,i}^{free}, \\ J_{V_i} &= \partial V_i / \partial \mathbf{q}_i = S_i \mathbf{n}_i^T. \end{aligned} \quad (7)$$

These individual volumes and volume gradients are then summed to formulate an aggregate volume constraint for both contacting bodies: $V = \sum V_i$ and $J_V = \sum J_{V_i}$ (Figure 2). Given \mathbb{H}_n the row of the constraint matrix \mathbb{H} corresponding to the separation constraint, the contribution of each contact i to the constraint matrix is:

$$\mathbb{H}_{n,i} = S_i \mathbf{n}_i^T. \quad (8)$$

3.2.2 Constraint evaluation

Signorini's law [11] is formulated as a complementarity relation (noted \perp) between the constraint force and the penetration volume. On the velocity solve this means that the constraint force must be repulsive or null ($\lambda_n \geq 0$), the volume must not increase ($\dot{V} \leq 0$), and the constraint applies pressure only if the penetration volume is null.

$$\lambda_n \geq 0 \perp J_V(\dot{\mathbf{q}}_0 + \Delta \dot{\mathbf{q}}) \leq 0. \quad (9)$$

As discussed in Section 3.1, the addition of this condition to Eq. 3 leads to an MLCP. For post-stabilization, the complementarity condition can be reformulated such that penetration volume is removed:

$$\Lambda_n \geq 0 \perp V(\mathbf{q}_{t+h}^{vel}) + J_V d\mathbf{q} \leq 0. \quad (10)$$

3.3 Non-uniform Pressure Distribution

With point contact approaches, each one of the contact constraints that sample a contact area undergoes a different contact pressure. Contact points with a larger penetration depth are expected to receive a higher pressure to resolve non-penetration. With the standard volume constraints formulated by Allard et al. [1], on the other hand, all point contacts aggregated into the same constraint undergo exactly the same pressure. With our basic formulation above, contact pressure is modulated by the area associated to each contact point, but it ignores the amount of penetration too (Figure 3). Next, we propose a method to support non-uniform pressure distribution within aggregate constraints.

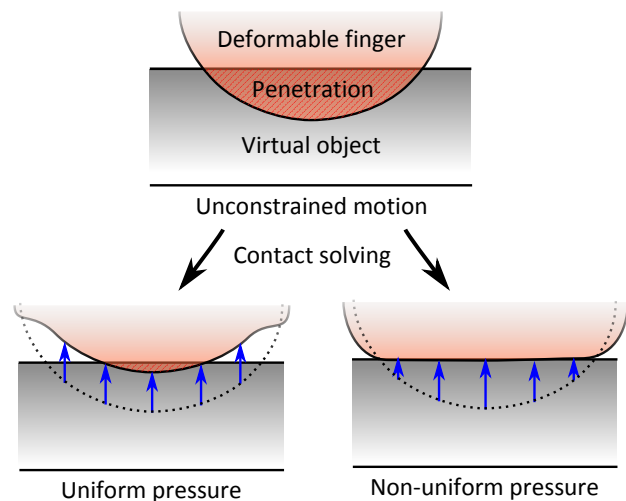


Fig. 3. Differences in contact solving with volume contact constraints using uniform and non-uniform pressure distribution.

Given multiple contact points on the skin of a phalanx, the amount of pressure force on each contact point indicates the force needed to enforce non-penetration for that particular contact. Assuming that the tissue in the contact area is roughly homogeneous, and that the object under contact is also homogeneous around the contact area, removing the contact forces would produce larger penetrations on points with higher contact pressure. Based on this observation, we have designed a method that applies a non-uniform distribution of pressure within each aggregate volume constraint as a function of the interpenetration during constraint-free motion.

Specifically, for an aggregate constraint with n_j contact points, we compute for each contact point i a weight w_i proportional to its pene-

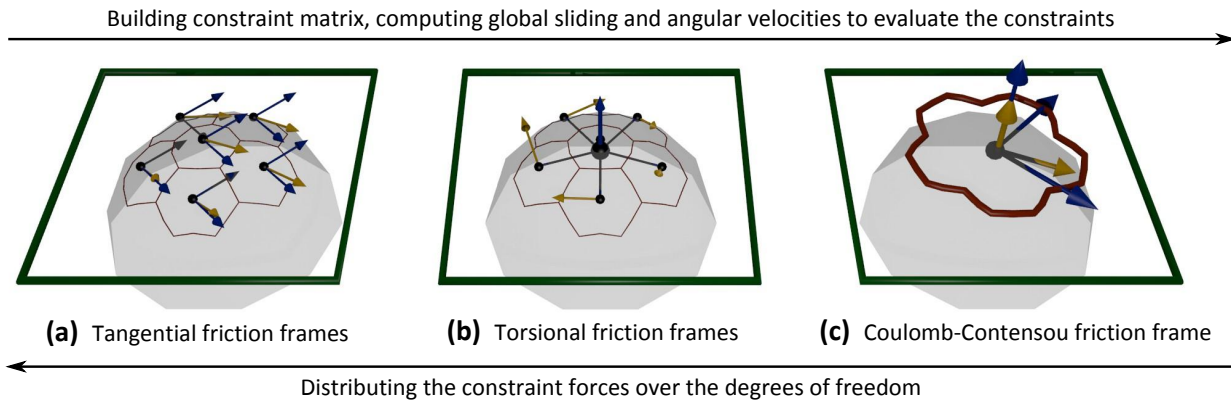


Fig. 4. Formulation of aggregate torsional friction constraints. (a) Individual Coulomb friction frames with sliding velocities at each contact point. (b) Torsional velocities evaluated from vectors orthogonal to the contact surface normal and lever arms between the contact points and their barycenter. (c) Summing these individual frames to evaluate linear and angular velocities for the contact surface.

tration distance:

$$w_i = n_j \delta_{n,i}^{free} / \sum_{k=1}^{n_j} \delta_{n,k}^{free}. \quad (11)$$

We use these weights to rescale the computation of penetration volumes in Eq. 7 as:

$$\begin{aligned} V_i &= w_i S_i \delta_{n,i}^{free}, \\ J_{V_i} &= w_i S_i \mathbf{n}_i. \end{aligned} \quad (12)$$

And as a result the entries of the constraint matrix from Eq. 8 are also rescaled as:

$$\mathbb{H}_{n,i} = w_i S_i \mathbf{n}_i^T. \quad (13)$$

3.4 Aggregate Torsional Friction Constraints

When an object is grasped with two fingers, torsional friction between the fingerpads and the object plays a key role in maintaining grasping stability. Using point contact constraints is sufficient to model regular tangential friction, and torsional friction is naturally obtained by accumulating all forces. With volume constraints, on the other hand, a naïve application of point-like friction constraints ignores all torsional friction.

We propose to incorporate an aggregate torsional friction constraint to each aggregate volume constraint, in way similar to regular tangential friction constraints, using the Coulomb-Contensou friction model [9, 28]. For simple deformable tissue such as finger phalanges, this allows to apply both tangential and torsional friction using a single set of 3 constraints per phalanx. The aggregation of torsional friction constraints is depicted in an example in Figure 4.

3.4.1 Constraint matrix

Tangential friction is expressed as two constraints spanning the tangent plane of the contact surface, represented by two unit vectors \mathbf{t} and \mathbf{s} orthogonal to the average volume contact normal \mathbf{n} . For each contact point, two terms are added to the constraint matrix as follows:

$$\begin{aligned} \mathbb{H}_{t,i} &= \frac{S_i}{S} \mathbf{t}^T, \\ \mathbb{H}_{s,i} &= \frac{S_i}{S} \mathbf{s}^T. \end{aligned} \quad (14)$$

where S is the area of the entire contact surface, defined as the sum of vertex areas: $S = \sum S_i$.

To model torsional friction, for each aggregate constraint we compute an area-weighted centroid of the contact points, $\mathbf{p}_c = \sum S_i \mathbf{p}_i / S$, and an area-weighted contact normal $\mathbf{n}_c = \sum S_i \mathbf{n}_i / S$ (followed by normalization). For each contact point in the aggregate constraint, we define a lever arm $\mathbf{r}_i = \mathbf{p}_i - \mathbf{p}_c$. Torsional friction terms are then added to the constraint matrix as follows:

$$\mathbb{H}_{\omega,i} = \mathbf{n}_c \times \mathbf{r}_i. \quad (15)$$

3.4.2 Constraint evaluation

Regular Coulomb friction aims to maximize the dissipation of relative tangential velocities subject to a constraint on the required tangential force, which should be smaller than a friction coefficient μ times the normal force. The Coulomb-Contensou model extends this principle while taking into account the relationship between tangential and torsional friction: the faster an object slides along a surface, the less inclined it will be to rotate around the contact normal, and vice versa.

The application of the Coulomb-Contensou model to our aggregate constraint setting requires the evaluation of relative angular velocities between the objects in contact. First, for each contact point we compute an angular velocity w.r.t. the torsional friction frame defined in Equation 15, using its relative angular velocity $\delta_{\omega,i}^{free}$ and its lever arm \mathbf{r}_i (Figure 4). We estimate the relative angular velocity of the aggregate contact as the area-weighted sum of individual velocities:

$$\omega^{free} = \left(\sum_i \frac{S_i}{S} \frac{\mathbf{r}_i \times \delta_{\omega,i}^{free}}{\|\mathbf{r}_i\|^2} \right) \cdot \mathbf{n}. \quad (16)$$

Leine and Glocker [28] define the constraints of the Coulomb-Contensou model through a velocity potential. Given the sliding velocity at each contact point i , computed as $\mathbf{v}_s = \delta_t^{free} \mathbf{t} + \delta_s^{free} \mathbf{s} + \omega^{free} \mathbf{n} \times \mathbf{r}_i$, the velocity potential is:

$$\Phi = \sum_i \frac{S_i}{S} \mu \lambda_n \|\mathbf{v}_s\|. \quad (17)$$

The admissible values for the tangential and torsional friction forces are then computed from derivatives of this velocity potential w.r.t. the tangential and angular velocity respectively:

$$\begin{aligned} \nabla_{\delta_t^f} \Phi &= \sum_i \frac{S_i}{S} \mu \lambda_n \frac{\mathbf{v}_s}{\|\mathbf{v}_s\|}, \\ \nabla_{\omega^f} \Phi &= \sum_i \frac{S_i}{S} \mu \lambda_n \mathbf{r}_i^* \frac{\mathbf{v}_s}{\|\mathbf{v}_s\|}, \end{aligned} \quad (18)$$

where \mathbf{r}_i^* is the cross-product matrix of \mathbf{r}_i .

Coulomb-Contensou friction constraints are defined as a complementarity condition, similarly to separation constraints. For tangential friction, either the friction force is strictly included inside the friction cone defined by the velocity potential derivative, in which case objects stick together, otherwise the objects slip tangentially and the force is on the border of the cone, along the direction of motion. The condition for the friction force $\lambda_{\mathcal{T}} = \sqrt{\lambda_t^2 + \lambda_s^2}$ is thus added to the resulting NLCP following:

$$\begin{aligned} \delta_{\mathcal{T}} = \vec{0} &\Rightarrow \|\lambda_{\mathcal{T}}\| < \left\| \frac{\partial \delta_{\mathcal{T}}^{free} \Phi}{\partial \lambda_{\mathcal{T}}} \right\| \text{ (stick)} \\ \delta_{\mathcal{T}} \neq \vec{0} &\Rightarrow \|\lambda_{\mathcal{T}}\| = - \left\| \frac{\partial \delta_{\mathcal{T}}^{free} \Phi}{\partial \lambda_{\mathcal{T}}} \right\| \frac{\delta_{\mathcal{T}}}{\|\delta_{\mathcal{T}}\|} \text{ (slip)}. \end{aligned} \quad (19)$$

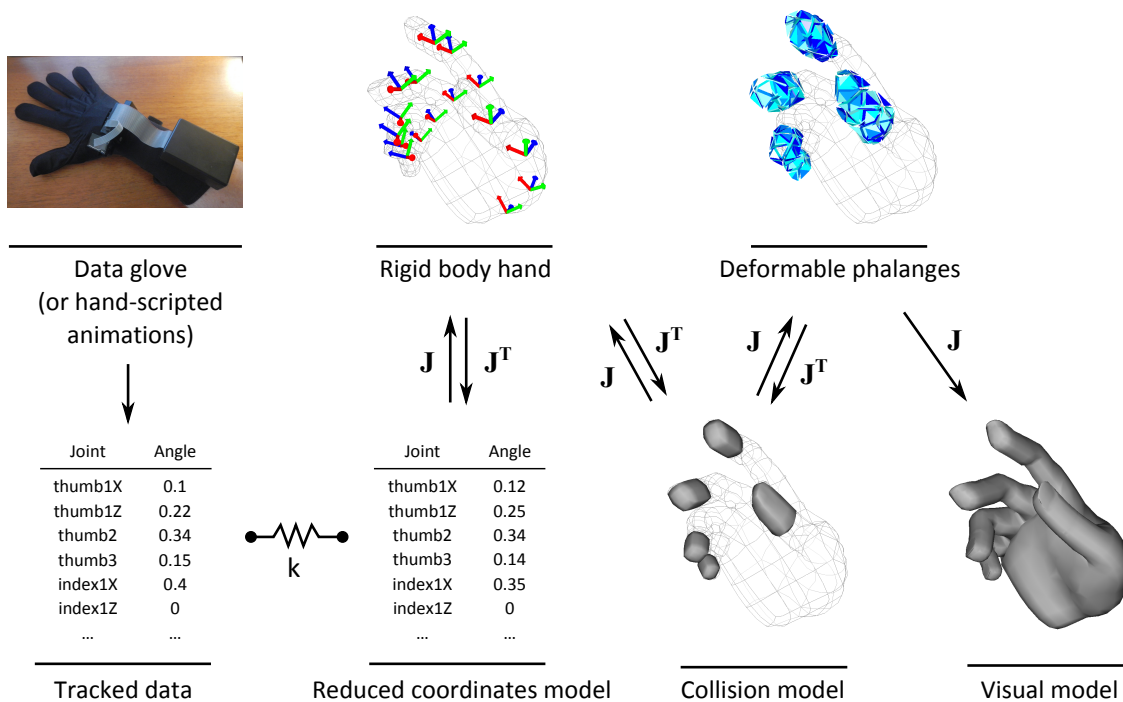


Fig. 5. Different layers of our proposed hand model. The reduced coordinates model follows the tracked data with springs, while the rigid body hand is mapped to it. The deformable phalanges are simulated with FEM. The collision models are mapped to both the rigid body model and the deformable phalanges using a corotational approach. The visual model is skinned from the deformable phalanges.

For the friction torque λ_ω , the condition is formulated in a similar manner, with either torsional sticking when the friction torque is within admissible values, or slipping otherwise. This is added to the MLCP following:

$$\begin{aligned} \delta_{\vec{T}} = \vec{0} &\Rightarrow \|\lambda_\omega\| < \|\partial_{\omega^{free}} \Phi\| \text{ (stick)} \\ \delta_{\vec{T}} \neq \vec{0} &\Rightarrow \|\lambda_\omega\| = -\|\partial_{\omega^{free}} \Phi\| \frac{\delta_{\vec{T}}}{\|\delta_{\vec{T}}\|} \text{ (slip)}. \end{aligned} \quad (20)$$

4 HAND MODEL

We propose a deformable hand model that leverages our aggregate constraint method for interactive grasping and dextrous manipulation of virtual objects. The model consists of 5 interconnected layers: tracked hand data, reduced coordinates model, mapped rigid body skeleton, deformable phalanges, and surface collision/visual model (Figure 5).

The tracked data is first stored as angular values at all joints of the hand, considering two degrees of freedom at the base of each finger (flexion and abduction) and one degree of freedom for both joints of each finger. The position and orientation of the palm are stored as well, to account for motions of the hand as a whole. The simulated hand is also first modeled in reduced coordinates, with 20 finger joint values and a 6DOF base value. Stiff springs are used on each joint to make this hand model follow the tracked hand as closely as possible. Unilateral springs are used in absence of haptic feedback, but bilateral springs can also be used as a virtual coupling scheme if haptic feedback is provided to the hand and/or fingers. Joints are also assigned minimum and maximum angular limits, which are enforced using stiff springs once either of these values are exceeded.

A mapping function is then used to determine the position of the palm and phalanges of the hand skeleton from the reduced coordinates model. Forces and constraints can be mapped both ways using jacobians and transpose jacobians, as described in [10]. This model thus serves as a proxy between the reduced-coordinate model and 3D space. The deformable phalanges are modeled as coarse tetrahedral meshes (1513 tetrahedra for all 15 phalanges), and simulated using linear cor-

rotational FEM [35]. Nodes whose positions match the anatomical position of bones are fixed using stiff springs.

For handling collisions using our approach, we assign one collision model per phalanx, hence all point contacts acting on the same phalanx are aggregated into a single volume constraint. Each collision model is a triangular surface mesh which is mapped to its respective deformable phalanx. Collision detection is performed using Local Minimum Distances (LMDs) [25], points being considered in penetration when their LMDs are under a threshold of approximately 2mm. A global corotational approach [36, 11] is used to map the collision models to both their corresponding rigid bodies and deformable phalanges. The motion of a collision model is thus the result of the global motion of its rigid body and the local relative displacement of its deformable phalanx. Inversely, transpose jacobians allow to map contacts at skin level to both the outer nodes of their deformable model, and to their rigid model.

The visual model uses a mapping function for the phalanges, and additionally uses regular skinning with linear blending from the rigid model for parts of the hand which are not given a collision model (e.g. the palm).

5 RESULTS

Our approach was implemented into the SOFA framework for physical simulation of both rigid and deformable bodies [13]. Simulations were run on a Intel Core i7-2720QM CPU, with a Nvidia Quadro 4000M GPU (used only for visual rendering). For real-time hand tracking, we used a Data Glove 5 (5DT), which provides 1DOF per finger. In addition, a combination of a GameTrak position tracker (In2Games) and a Colibri inertial motion tracker (TRIVISIO Prototyping GmbH) was used to track the 6DOF position of the hand. The limited tracking capability of the data glove allows only simple interactions, such as enclosing and opening motions of the hand (see Figure 6c). To demonstrate our methods on more dextrous interactions, we also designed benchmarks with scripted animations, executing the simulations in real time and with the scripted hand configurations substituting the input from the data glove.

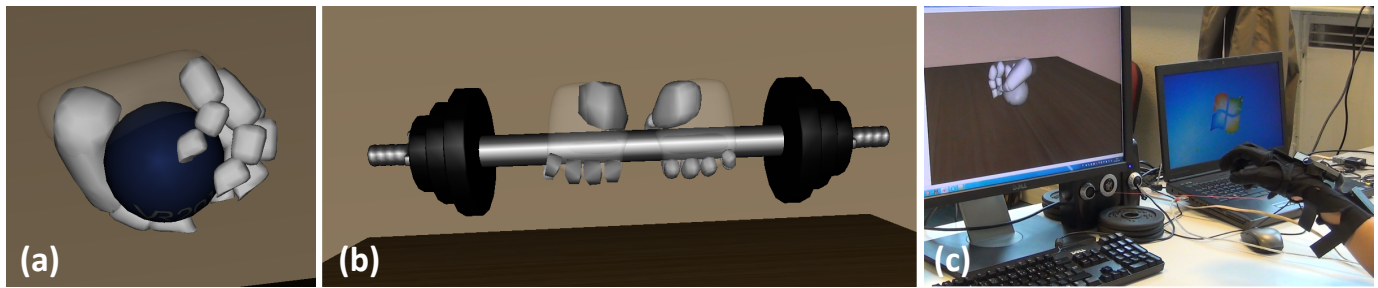


Fig. 6. Some object manipulation scenarios with our method. (a) Grasping a rigid ball with all fingers deformable. (b) Lifting a dumbbell with two fully deformable hands. (c) Manipulating a deformable ball using a data glove.

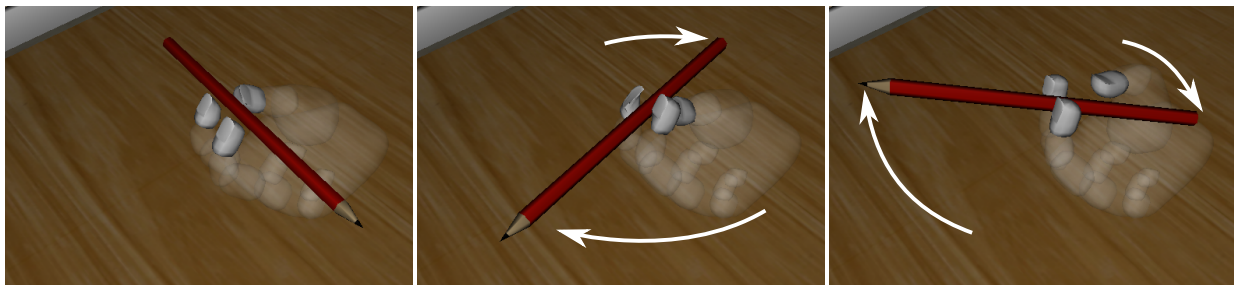


Fig. 7. Dexterous manipulation of a pen using three soft fingers with our approach. The pen is picked up and spun between the fingertips.

5.1 Illustrative Use Cases

Grasping a cube from its edges is a typical scenario showing the need for deformable fingers, as rigid fingers are unable to produce a sufficient contact area for stable grasping. This scenario also showcases the importance of pressure distribution in our contact model (Figure 8). With uniform pressure over the contact surface, grasping is not successful, similar to the result with rigid fingers. However, our weighting of contact points within the aggregate constraint formulation shown in Section 3.3 restores the pressure distribution that would be expected with point contact constraints, and allows the cube to be lifted.

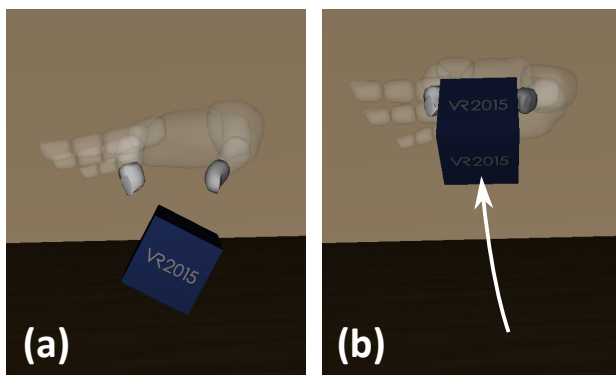


Fig. 8. Grasping a cube from its edges. (a) Uniform pressure distribution is not sufficient to lift the object. (b) Our non-uniform pressure distribution allows grasping.

The Coulomb-Contensou friction model accurately represents both the friction forces and torques at fingers in contact, as shown when grasping a long object from one of its ends with two fingers (Figure 9). The object stays in a horizontal position as long as the fingers apply a sufficient force, and good control of rotations around the grasping axis is provided as well.

Our method is suitable for simulating interaction scenarios that would be computationally challenging with point contact forces, such as enclosing and grasping a rigid ball with all fingers deformable (Fig-

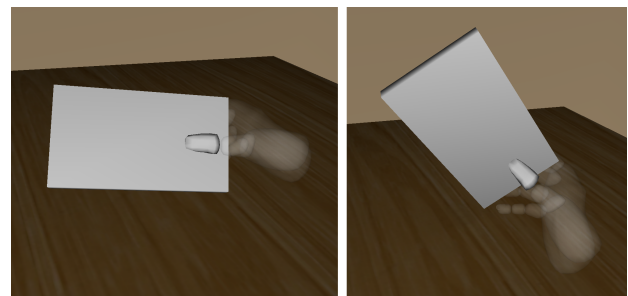


Fig. 9. Example of Coulomb-Contensou friction with our approach based on volume contact constraints. Two deformable fingers resist moments around the grasping axis, and allow rotating objects around that axis without unintentional slipping.

ure 6a), or lifting a dumbbell with two hands (Figure 6b). Both scenes present a large number of degrees of freedom and contact constraints. Highly dexterous manipulation is also possible, as demonstrated by a pencil spinning scene (Figure 7). Finally, our method is suitable for real-time interaction with data gloves, as shown by a full-hand interaction scenario with a deformable ball (Figure 6c).

5.2 Computation Times

Performance for the aforementioned simulations was measured in terms of computation times for the constraint solve and the entire simulation step w.r.t. the complexity of the scene, measured in terms of the number of deformable phalanges and contact points. No parallelism was used for the simulation, all steps being performed sequentially. The results shown in Table 1 have been averaged over 100 iterations.

In the edge grasping scene, only the last phalanges of the thumb and index were simulated. Two slightly different grasps were used to compare performance under a varying number of contact points but the same number of phalanges in contact. With 19 contacts on average, constraint solving times were reduced by 69%, a reduction that increased to 77% with 23 contacts. The overall simulation time was similarly reduced by 43% and 55% respectively.

For the rigid ball grasping simulation, the hand had all phalanges

Table 1. Performance of classical point contact constraints and our volume contact constraint approach on different manipulation scenarios.

Scenario	Figure	Phalanges	Contacts	Number of constraints		Constraint solving (ms)		Total iteration time (ms)	
				Point	Our approach	Point	Our approach	Point	Our approach
Pencil spinning	7	3	33	52	12	14.91	4.85	25.21	15.3
Edge grasping	8	2	19	49	8	14.01	4.28	22.76	13.05
			23	62	8	22.52	5.26	32.45	14.53
Rigid ball	6a	15	37	86	22	37.78	10.8	59.13	32.58
			84	233	32	229.74	15.33	254.05	39.94
Bimanual dumbbell	6b	30	93	283	70	350.91	53.42	408.26	116.53

simulated. Two cases were considered: a first one where the ball is lifted with the fingertips, generating 37 contacts in average during grasping, and a second one where the hand grasps the ball more closely, with more phalanges involved and 84 contacts in average. In both cases, the simulation result is the same as with point contact constraints. Similarly to the edge grasping scene, increasing the number of contacts also increases the gain of our method for constraint solving performance, from 71% to 93%. Moreover, in the more complex scene, constraint solving was a clear bottleneck with point contact constraints, taking 90% of the simulation step. With our aggregate constraints, it took only 38%. In this scene, our method provides a dramatic improvement of the frame rate, increasing from 3.94 fps to 25.03 fps.

For simulations with a low number of phalanges in contact, and hence less contact points, such as the pen spinning scene, our method still improves greatly constraint solving times, which are almost halved in this case. However, the impact is then smaller on the overall performance, as the simulation cost is shared with other components, such as FEM simulation.

The dumbbell lifting scenario is a more challenging scenario for our method, as it contains 30 deformable phalanges during two-handed interaction. This scenario also leads to many collisions between several phalanges and the bar, pushing the number of contacts to an average of 93 during lifting. Constraint solving thus becomes a critical part in the simulation step, taking 86% of the time with point contact constraints. Even on such a difficult case for our method, it reduces constraint solving time by 85%, and in turn reduces the overall computation time by 71%, for the same visual result.

5.3 Discussion

Our aggregate contact constraint method effectively reduces the number of constraints induced by contact with deformable phalanges during grasping and dexterous manipulation of virtual objects. This leads to a significant reduction of constraint solving times as well as overall simulation times on scenarios with moderate to high complexity. The method is especially efficient when the number of contacts per phalanx is high, and its cost grows with the number of phalanges in contact, but even in the latter case it still leads to significant improvement in computation times.

Simulations with our method are otherwise visually very similar to simulations obtained with point contact constraints, with pressure distribution and torsional friction being accurately represented. Since our method is constraint-based, we maintain a collision-free state at all frames. As a result, deep penetration simply does not occur. Shallow penetrations are produced by the constraint-free velocities on each frame, these penetrations are used to compute non-uniform pressure distribution, and then they are resolved in the constraint solve.

Our method allows to attain the interactive frame rates suitable for real-time interaction with fully deformable fingers using a data glove (over 60 fps for some scenarios). We could expect even better global performance by using parallel processing to compute computationally heavy parts such as FEM. It can then be used to build more complex interaction scenarios for applications such as surgical training, virtual prototyping or real-time physically-based computer animation. Addi-

tionally, our method proves interesting for enhancing such interactive scenarios with haptic feedback, which requires high simulation rates. The aggregate constraints provide a global force and average direction for an entire phalanx, which could be used to display kinesthetic feedback to the fingers of the user. The computation of pressure distribution, on the other hand, could be used for tactile feedback on the finger pads.

5.4 Limitations

The only missing element for whole-hand grasping in our examples is the palm, which is a more complex structure that cannot simply be mapped to a single rigid body, unlike the phalanges, due to its more complex underlying skeletal structure. Another limitation is that currently, constraint aggregation is done per phalanx, with no adaptivity. This could be alleviated by doing the aggregation in an adaptive manner, *e.g.* using contact clustering methods.

The complete hand model exhibits compliance in both the deformable skin and the coupling between the bones and the tracker. There is no compliance at contact, due to the use of constraints, hence there is no interpenetration (after the constraint solve). If high pressure is exerted during grasping, the most compliant component will yield and suffer the largest deformation. To avoid excessive finger squishing, the coupling between tracker and bones needs to be more compliant than the skin. We have used linear corotational FEM to simulate the deformation of fingers, which sets a trade-off between the softness of the skin and the compliance of the coupling with the tracker. Our method could be extended by using a strain-limiting deformation model [41]. With this model the skin would be more compliant at initial contact, while the coupling with the tracker would be more compliant under high grasping pressure. However, even with linear FEM, our method still allows interactive grasping with accurate friction.

6 CONCLUSION

We have presented a novel contact constraint method dedicated to the simulation of interaction between deformable hand models and virtual objects. Our constraint-based formulation aggregates all contacts on each phalanx into a single set of separation and friction constraints. Our approach greatly reduces the number of constraints for interaction scenarios with deformable finger pads which would otherwise generate numerous constraints during grasping. Specially in complex scenarios, our method leads to a significant improvement in terms of solving time for constraints, as well as global simulation times.

We have introduced a weighting method for simulating the non-uniform pressure distribution over the contact surface of finger pads. This approach better simulates the contact response when grasping objects by sharp edges. We have also proposed an aggregate formulation of the Coulomb-Contensou friction model within our method, in order to simulate torsional friction at contact surfaces without requiring multiple friction frames. We thus have a minimal set of constraints for accurately simulating contact and friction for an entire contact surface, which proves suitable for all sorts of dexterous manipulation scenarios. The visual results with our approach prove to be similar to those obtained with common point contact constraints, and future work could

involve its validation by comparing the resulting forces and deformations against real-world measurements.

Our method can therefore be exploited to simulate more complex real-time scenarios involving dexterous manipulation of physically-based virtual objects, for instance in virtual prototyping and assembly, or to simulate the fine manipulation of surgical tools. A user study could be performed to assess the usability of our method for more complex manipulation scenarios. Furthermore, the performance gain of the method can be used to attain the high simulation rates required in the context of haptics, so as to provide kinesthetic and tactile feedback to the fingers.

ACKNOWLEDGMENTS

This research is supported in part by ANR (project Mandarin - ANR-12-CORD-0011), the Spanish Ministry of Economy (TIN2012-35840), and the EU FP7 project WEARHAP (601165). The authors also wish to thank Gabriel Cirio.

REFERENCES

- [1] J. Allard, F. Faure, H. Courtecuisse, F. Falipou, C. Duriez, and P. G. Kry. Volume contact constraints at arbitrary resolution. *ACM Transactions on Graphics*, 29(4):82:1–82:10, 2010.
- [2] D. Baraff. Coping with friction for non-penetrating rigid body simulation. *Computer Graphics*, 25(4):31–40, 1991.
- [3] D. Baraff. Fast contact force computation for nonpenetrating rigid bodies. In *Proc. of 21st Annual Conference on Computer Graphics and Interactive Techniques*, pages 23–34, 1994.
- [4] D. Baraff. Linear-time dynamics using lagrange multipliers. In *Proc. of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 137–146, 1996.
- [5] F. Barbagli, A. Frisoli, K. Salisbury, and M. Bergamasco. Simulating human fingers: a soft finger proxy model and algorithm. In *Proc. of 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 9 – 17, 2004.
- [6] J. Bender, K. Erleben, and J. Trinkle. Interactive simulation of rigid body dynamics in computer graphics. *Computer Graphics Forum*, 33(1):246–270, 2014.
- [7] C. Borst and A. Indugula. Realistic virtual grasping. In *Proc. of IEEE Virtual Reality Conference*, pages 91–98, 2005.
- [8] M. Ciocarlie, C. Lackner, and P. Allen. Soft finger model with adaptive contact geometry for grasping and manipulation tasks. In *Proc. of World Haptics Conference*, pages 219–224, 2007.
- [9] P. Contensou. Couplage entre frottement de glissement et frottement de pivotement dans la théorie de la toupie. In *Kreiselp Probleme / Gyrodynamics*, pages 201–216. Springer Berlin Heidelberg, 1963.
- [10] C. Duriez, H. Courtecuisse, J. Alcalde, and P. Bensoussan. Contact skinning. In *Proc. of Eurographics*, volume 27, pages 313–320, 2008.
- [11] C. Duriez, F. Dubois, A. Kheddar, and C. Andriot. Realistic haptic rendering of interacting deformable objects in virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 12:36–47, 2006.
- [12] K. Erleben. Velocity-based shock propagation for multibody dynamics animation. *ACM Transactions on Graphics*, 26(2), 2007.
- [13] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, and S. Cotin. SOFA: A multi-model framework for interactive physical simulation. In *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, volume 11, pages 283–321. Springer Berlin Heidelberg, 2012.
- [14] C. A. Felippa. A systematic approach to the element-independent corotational dynamics of finite element. Technical report, Center for Aerospace Structures, 2000.
- [15] A. Frisoli, F. Barbagli, E. Ruffaldi, K. Salisbury, and M. Bergamasco. A limit-curve based soft finger god-object algorithm. In *14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 217–223, 2006.
- [16] N. Galoppo, M. A. Otaduy, S. Tekin, M. Gross, and M. C. Lin. Soft articulated characters with fast contact handling. *Computer Graphics Forum*, 26(3):243–253, 2007.
- [17] C. Garre, F. Hernandez, A. Gracia, and M. Otaduy. Interactive simulation of a deformable hand for haptic rendering. In *Proc. of World Haptics Conference*, pages 239–244, 2011.
- [18] J.-P. Gourret, N. M. Thalmann, and D. Thalmann. Simulation of object and human skin formations in a grasping task. *SIGGRAPH Computer Graphics*, 23(3):21–30, 1989.
- [19] D. Holz, S. Ullrich, M. Wolter, and T. Kuhlen. Multi-contact grasp interaction for virtual environments. *Journal of Virtual Reality and Broadcasting*, 5(7), 2008.
- [20] R. D. Howe and M. R. Cutkosky. Practical force-motion models for sliding manipulation. *International Journal of Robotics Research*, 15(6):557–572, 1996.
- [21] G. Irving, J. Teran, and R. Fedkiw. Invertible finite elements for robust simulation of large deformation. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 131–140, 2004.
- [22] J. Jacobs and B. Froehlich. A soft hand model for physically-based manipulation of virtual objects. In *Proc. of IEEE Virtual Reality Conference*, pages 11 –18, 2011.
- [23] J. Jacobs, M. Stengel, and B. Froehlich. A generalized god-object method for plausible finger-based interactions in virtual environments. In *Proc. of IEEE Symposium on 3D User Interfaces*, pages 43–51, 2012.
- [24] S. Jain and C. K. Liu. Controlling physics-based characters using soft contacts. In *Proc. of SIGGRAPH Asia Conference*, pages 163:1–163:10, 2011.
- [25] D. Johnson and P. Willemsen. Six degree-of-freedom haptic rendering of complex polygonal models. In *Proc. of 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 229–235, 2003.
- [26] Y. J. Kim, M. A. Otaduy, M. C. Lin, and D. Manocha. Six-degree-of-freedom haptic rendering using incremental and localized computations. *Presence: Teleoperators and Virtual Environments*, 12(3):277–295, 2003.
- [27] P. G. Kry and D. K. Pai. Interaction capture and synthesis. *ACM Transactions on Graphics*, 25(3):872–880, 2006.
- [28] R. I. Leine and C. Glocker. A set-valued force law for spatial coulomb-contensou friction. *European Journal of Mechanics - A/Solids*, 22(2):193 – 216, 2003.
- [29] C. K. Liu. Synthesis of interactive hand manipulation. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 163–171, 2008.
- [30] E. Miguel and M. A. Otaduy. Efficient simulation of contact between rigid and deformable objects. In *ECCOMAS - Multibody Dynamics*, 2011.
- [31] V. J. Milenkovic and H. Schmidl. Optimization-based animation. In *Proc. of 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 37–46, 2001.
- [32] M. Moehring and B. Froehlich. Pseudo-physical interaction with a virtual car interior in immersive environments. In *Proc. of Eurographics Conference on Virtual Environments*, pages 181–189, 2005.
- [33] M. Moehring and B. Froehlich. Enabling functional validation of virtual cars through natural interaction metaphors. In *Proc. of IEEE Virtual Reality Conference*, pages 27–34, 2010.
- [34] A. Moravanszky and P. Terdiman. Fast contact reduction for dynamics simulation. In A. Kirmse, editor, *Game Programming Gems 4*, pages 253–263. Charles River Media, 2004.
- [35] M. Müller and M. Gross. Interactive virtual materials. In *Proc. of Graphics Interface*, pages 239–246, 2004.
- [36] M. Nesme, M. Marchal, E. Promayon, M. Chabanas, Y. Payan, and F. Faure. Physically realistic interactive simulation for biological soft tissues. *Recent Research Developments in Biomechanics*, 2:1–22, 2005.
- [37] M. Ortega, S. Redon, and S. Coquillart. A six degree-of-freedom god-object method for haptic display of rigid bodies with surface properties. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):458–469, 2007.
- [38] M. A. Otaduy, R. Tamstorf, D. Steinemann, and M. Gross. Implicit contact handling for deformable objects. *Computer Graphics Forum*, 28(2):559–568, 2009.
- [39] R. Ott, V. De Perrot, D. Thalmann, and F. Vexo. MHaptic: a haptic manipulation library for generic virtual environments. In *Proc. of International Conference on Cyberworlds*, pages 338–345, 2007.
- [40] M. Pauly, D. K. Pai, and L. J. Guibas. Quasi-rigid objects in contact. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 109–119, 2004.
- [41] A. Perez, G. Cirio, F. Hernandez, C. Garre, and M. Otaduy. Strain limiting for soft finger contact simulation. In *Proc. of World Haptics Conference*, pages 79–84, 2013.

- [42] G. Picinbono, H. Delingette, and N. Ayache. Non-linear anisotropic elasticity for real-time surgery simulation. *Graphical Models*, 65(5):305–321, 2003.
- [43] M. Pouliquen, C. Duriez, C. Andriot, A. Bernard, L. Chodorge, and F. Gosselin. Real-time finite element finger pinch grasp simulation. In *Proc. of World Haptics Conference*, pages 323–328, 2005.
- [44] S. Sueda, A. Kaufman, and D. K. Pai. Musculotendon simulation for hand animation. *ACM Transactions on Graphics*, 27(3), 2008.
- [45] A. Talvas, M. Marchal, and A. Lécuyer. The God-Finger Method for Improving 3D Interaction with Virtual Objects through Simulation of Contact Area. In *Proc. of IEEE Symposium on 3D User Interfaces*, pages 111–114, 2013.
- [46] Y. Ye and C. K. Liu. Synthesis of detailed hand manipulations using contact sampling. *ACM Transactions on Graphics*, 31(4):41:1–41:10, 2012.
- [47] G. Zachmann and A. Rettig. Natural and robust interaction in virtual assembly simulation. In *Proc. of ISPE International Conference on Concurrent Engineering: Research and Applications*, 2001.
- [48] Y. Zhuang and J. Canny. Real-time simulation of physically realistic global deformation. In *Proc. of IEEE Visualization Conference*, 1999.