

Dexterous Manipulation of Cloth

Yunfei Bai[†]

Wenhai Yu[†]

C. Karen Liu

Georgia Institute of Technology

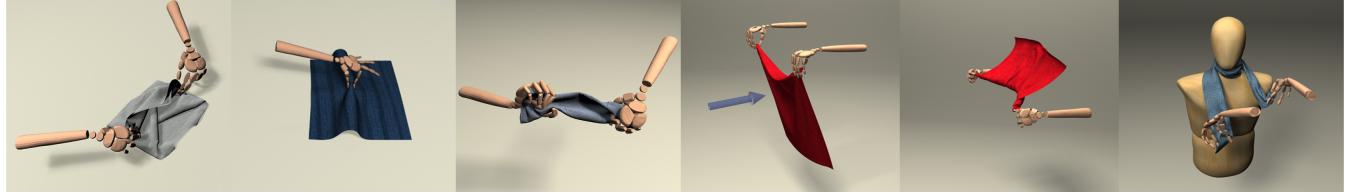


Figure 1: Given a simple description of the desired cloth motion, our algorithm computes appropriate joint torques for a physically simulated hand, such that, via contact forces, the result of cloth simulation follows the desired motion.

Abstract

This paper introduces a new technique to synthesize dexterous manipulation of cloth. Given a simple description of the desired cloth motion, our algorithm computes appropriate joint torques for physically simulated hands, such that, via contact forces, the result of cloth simulation follows the desired motion. Instead of optimizing the hand control forces directly, we formulate an optimization problem that solves for the commanding forces from the hands to the cloth, which have more direct impact on the dynamic state of the hands and that of the cloth. The solution of the optimization provides commanding forces that achieve the desired cloth motion described by the user, while respecting the kinematic constraints of the hands. These commanding forces are then used to guide the joint torques of the hands. To balance between the effectiveness of control and computational costs, we formulate a model-predictive-control problem as a quadratic program at each time step. We demonstrate our technique on a set of cloth manipulation tasks in daily activities, including folding laundry, wringing a towel, and putting on a scarf.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Dexterous manipulation of cloth is a unique human skill essential to many activities of daily living spanning from dressing to grooming to folding laundry. Synthesizing these activities automatically for computer animation requires realistic depiction of human dexterity and cloth simulation. An experienced artist with the aid of a state-of-the-art cloth simulator might be able to animate simple manipulation tasks through trial-and-error, but the complex interaction of dexterous manipulation and cloth motion demands more sophisticated automatic solution.

Dexterous manipulation and user-control of cloth simulation are two active computer animation research areas that have rarely

crossed path in the past. Researchers have demonstrated control algorithms on simulated hands to achieve impressive dexterous tasks, but the scope has been limited to manipulating rigid objects. Deformable objects, such as cloth, present new challenges because the hand has to manipulate a much larger number of uncontrollable degrees of freedom of cloth. Furthermore, the complex relation of the cloth state and contact forces increases the difficulty of dexterous controller design. On the other hand, a separate research community has developed various algorithms to control physically simulated cloth according to user-specifications. These techniques aim to produce the cloth motion alone without consideration of human hands, thereby unphysical forces are allowed to be applied anywhere on the cloth. Directly applying existing techniques from either area is insufficient to address the new challenges of dexterous manipulation of cloth.

We propose a new technique to consolidate the constraints imposed by hand manipulation and control of cloth simulation. In our

[†] Yunfei Bai and Wenhai Yu both contributed equally to this paper, and should be considered equal first authors.

**Figure 2:** Fold a T-shirt.

framework, the artist provides desired trajectories of a small set of features on the cloth (e.g. vertices at corners of the cloth) and expects the resulting cloth motion to follow the input feature trajectories through the contact forces provided by the virtual hands. Our algorithm formulates a reduced optimization problem that solves for the commanding forces from the hands to the cloth, a pivotal physical quantity that couples the state of the hands and that of the cloth. The optimization aims to find the optimal commanding forces that achieve desired cloth motion described by the user, while respecting constraints from contact dynamics and hand kinematics. To balance between the effectiveness of control and computational costs, we formulate a model-predictive-control (MPC) problem as a quadratic program (QP) at each time step.

We demonstrate our method on a variety of cloth manipulation tasks such as folding laundry, wringing a towel, or putting on a scarf (Figure 1). To validate the cloth motion planning algorithm quantitatively, we randomly generate a set of feature trajectories in a defined space and measure the accuracy of the resultant cloth motion. The evaluation shows that our cloth motion planning can accurately achieve the user-specifications on the cloth and can be successfully executed by the simulated hands.

2. Related Work

Previous work in dexterous hand manipulation in computer animation has demonstrated a variety of manipulation strategies on physically simulated hands, such as finger gaiting [YL12], rolling/sliding [Liu09, MPT12, BL14a], or grasping/regrasping [PZ05, KP06, Liu08, ZZMC13, WMZ*13]. Researchers have also developed more accurate hand models with detailed simulation of tendons and muscles [SKP08] and demonstrated their impact on control of manipulation tasks [SSB*15]. In spite of great progress made in this research area, existing techniques are limited to manipulating rigid objects with only six degrees of freedom.

In contrast, robotics researchers have explored manipulation of deformable objects extensively, with an emphasis on folding clothes [OSK07, BC10, VDBMGA11, CTSM*11, BPK11, MVDBF*12, Jim12]. A more comprehensive review can be found in the survey written by Jiménez [Jim12]. Many previous approaches improved the control and planning algorithms by using cloth simulation techniques to estimate the state of the cloth. Because the interaction between the robot hands and the cloth is relatively simple (i.e. grasping only), the simulation method can be simplified to applying position constraints to pin the cloth in the air instead of simulating the hands. In this paper, our goal is to provide more general computational tools suitable for a wide set

of manipulation tasks in addition to grasping. In addition, many existing methods [BC10, VDBMGA11, MVDBF*12] assume the cloth state to be quasi-static during manipulation while our algorithm aims to handle cloth moving in high velocity. Some existing studies also looked into manipulation of deformable linear objects such as tying a knot [MK06, SI06, SI07, RWE*09, RPCR12]. Probabilistic roadmap based methods are used to control the motion of these deformable linear objects with topological motion planner [SI06, SI07] or minimal energy curve motion planner [MK06]. Because the algorithms were designed specifically for linear objects, they cannot be applied to the problem of manipulating cloth with arbitrary representations.

Another related topic in robotics is sampling-based motion planning techniques for deformable objects/robots [GSLM05, RLA06, GRS*07, MLM08, PvdBA11, CKHL11, PGB14]. To consider the dynamics of the deformable objects, physical simulation is used to generate the samples [RLA06, GRS*07], or to follow the generated paths [GSLM05, MLM08, PvdBA11]. Other techniques efficiently model deformable objects without costly physical simulation by using voxel-based representation [PGB14]. The performance of the sampling-based methods can degrade considerably for deformable objects/robots with high dimensionality. Instead, we use gradient-based method to compute the optimal control force, and we formulate a MPC as a QP at each time step which has the benefit of fast computation speed and avoiding local minimum. Moreover, the existing techniques assume the deformable robot is fully actuated or the deformable object can be controlled by arbitrary unphysical forces. Our problem is more restricted where the control forces can only be applied at the contact points between hands and cloth through physical simulation.

Accurate two-way coupling between cloth and rigid bodies is important to realize dexterous hand manipulation of cloth. A simple task of pinching a piece of cloth between two fingers could fail without accurate contact forces and successful collision handling. Likewise, simulated hands will not be able to create budes or wrinkles on the cloth without appropriate friction forces. Researchers have proposed various methods to handle two-way coupling between rigid bodies and deformable bodies [JV03, SSIF07, SSF08, OTSG09, MO11, SVAC12, BL14b]. Recently, Bai et al. proposed a method to couple cloth and rigid body simulation using the existing simulators as black box [BL14b]. Their method allows rigid bodies to impart friction forces on cloth and avoids unsolvable collision situations between the rigid bodies and the cloth. We adopted their simple method for handling two-way coupling because we desired to use existing rigid body and cloth simulators. However, their method focused on simulation algorithms and only demonstrated a

few manually-designed grasp controllers insufficient for more complex manipulation tasks of cloth.

Controlling simulated cloth motion for computer animation and special effect applications has been investigated extensively [WMT06, BMWG07, JC07, BSG12, HSvTP12]. Many approaches formulated a large spacetime optimization and solved it efficiently using different methods, such as adjoint method [WMT06] or model reduction [BSG12, HSvTP12]. Jeon et al. used spacially localized controllers to control the deformable objects, which are fast and conceptually simple [JC07]. Bergou et al. used low-resolution cloth simulation as preview and tracked this motion with high-resolution simulation to enhance visual details and fidelity [BMWG07]. These methods can generate desired cloth motion specified by the user, but they allow for unphysical forces applied on the cloth, as if the cloth is an actively actuated object. Our cloth motion planning algorithm is similar in that it creates desired cloth motion according to user specifications, but the commanding forces applied on the cloth must be caused by the movement of hands in contact with the cloth.

Though still in its infancy, a few methods have demonstrated the potential and the importance of cloth manipulation in character animation [HK09, WK12, WSSK13, MFS14, CTTL15]. Ho and Komura introduced a technique to interact with deformable bodies using topology coordinates [HK09]. Wang et al. showed a virtual human putting on a sock and a pair of shorts by using electric flux for path planning [WSSK13]. Most recently, Clegg et al. proposed a method for animating human dressing [CTTL15]. However, the interaction between hands and cloth is achieved simply through position constraints. In this work, we are interested in animating manipulation of cloth with dexterous hands through simulated contact and friction.

3. Overview

Most cloth manipulation tasks can be broken down to a sequence of *contact phases*. In each contact phase, a *grasp/regrasp* action first establishes contact between the hand and the cloth followed by a *manipulation* action that brings the cloth to a desired state. For example, the folding task shown in Figure 2 requires three contact phases, each of which involves grasping a corner of the cloth followed by moving the corner to the desired location.

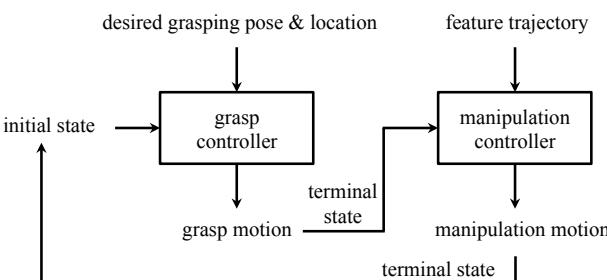


Figure 3: Our algorithm consists of two components: a grasp controller and a manipulation controller.

Based on this observation, our algorithm (Figure 3) consists of

two components: a grasp controller (Section 5) and a manipulation controller (Section 4). The grasp controller takes as input the desired grasping pose and the desired grasping location on the cloth and outputs a sequence of hand and cloth motion which terminates when the cloth is firmly gripped by the hand in the desired pose at the intended location. The terminal state of the grasp motion becomes the initial state of the manipulation controller. Starting from this initial state, the manipulation controller takes as input the desired motion of cloth described by a few feature trajectories and simulates the motion of hand manipulating the cloth. A feature can be any linear function of the vertices on the cloth. The final state of the manipulation motion becomes the initial state of the next contact phase.

4. Manipulation Controller

The manipulation controller aims to achieve the desired motion of cloth through the actions of hands. At each time step, the manipulation controller consists of three consecutive steps: cloth motion planning, rigidity rectification, and hand control. Cloth motion planning, formulated as a model-predictive-control problem, solves for required commanding forces from the hands such that the cloth tracks the user-specified feature trajectories closely. Rigidity rectification makes sure that the movement of contact points from the plan is achievable by the grasping hand. Finally, the rectified contact points are used to guide the actuation of the grasping hand.

4.1. Cloth Motion Planning

Directly solving for hand actuation to manipulate the cloth is very challenging due to complex dynamic coupling between the state of the hand ($\mathbf{q}, \dot{\mathbf{q}}$) and the state of the cloth ($\mathbf{u}, \dot{\mathbf{u}}$). Instead, our computation focuses on solving commanding forces, $\mathbf{f}_c \in \mathbb{R}^{3n}$, where n is the number of vertices of the cloth. The goal of the motion planning is to produce \mathbf{f}_c that makes the cloth track the user-specified feature trajectories, while being achievable by the hands through the contact points.

At each time step i , we formulate a short-horizon optimization to solve for a small window of commanding forces, $\mathbf{f}_c^{i+1}, \dots, \mathbf{f}_c^{i+K}$, from frame i . We denote the time difference from the current frame by index k and the window size by K . After solving all the commanding forces in the current window, we use the state of the cloth integrated with the first commanding force \mathbf{f}_c^{i+1} and other forces in the system to guide the control of the hand (Section 4.3). With the coupled simulation of hand and cloth, we obtain the next state of cloth, $(\mathbf{u}^{i+1}, \dot{\mathbf{u}}^{i+1})$. At the next time step, the optimization window slides one step forward and a new plan for the next K frames (Table 1 lists values of K in our implementation) of commanding forces are optimized. The remaining of Section 4.1 describes how this short-horizon optimization can be formulated as a quadratic program and solved efficiently.

4.1.1. Physical Feasibility

While we wish to avoid solving the full state of the hand in the optimization, we still need to make sure that the grasping hand is capable of producing \mathbf{f}_c . As such, we assume that the commanding forces can only be applied at static contact points and attempt



Figure 4: A dynamic secondary motion can be controlled by our algorithm.



Figure 5: Wring a towel.

to maintain static contact during manipulation. This assumption results in a conservative control strategy that does not explicitly exploit slipping or rolling strategies, but it is still sufficient to achieve a wide range of cloth manipulation tasks in daily life as shown in the result section.

The algorithm first selects the vertices that are in static contact with fingers in the previous time step. For each finger o at every time step i , our algorithm identifies a selected set of vertices on the cloth, $\hat{\mathbf{u}}_j \in \mathcal{P}_o^i$, that are in static contact with the finger at frame $i - 1$. The finger o can intentionally apply commanding forces to manipulate the cloth only through $\hat{\mathbf{u}}_j$. The control strategy obeys the Coulomb friction cone condition for static contact by expressing \mathbf{f}_c as:

$$\mathbf{f}_c = [(\mathbf{B}_1 \mathbf{d}_1)^T (\mathbf{B}_2 \mathbf{d}_2)^T \cdots (\mathbf{B}_n \mathbf{d}_n)^T]^T. \quad (1)$$

For those cloth vertices not in any \mathcal{P}_o^i , we directly set $\mathbf{B}\mathbf{d} = \mathbf{0}$. For those cloth vertices in the union of all \mathcal{P}_o^i 's, the commanding force is expressed as the bases of the approximated friction cone \mathbf{B} multiplied by the coefficients \mathbf{d} . The cloth vertex may have multiple fingers in contact with it (e.g. one finger from each side of the cloth). Therefore, $\mathbf{B} \in \mathbb{R}^{3 \times 4l}$ is the combined friction cone, and $\mathbf{d} \in \mathbb{R}^{4l}$ is the combined coefficients, where l is the number of fingers in contact. \mathbf{B} is determined by the normal direction of contact and the friction coefficient. The friction cone condition requires

$$\mathbf{d} \geq \mathbf{0}. \quad (2)$$

The friction cone coefficients, $\mathbf{D} = [\mathbf{d}_1^T \cdots \mathbf{d}_n^T]^T$, will be the variables in the optimization described below.

We also wish that the vertices in \mathcal{P}_o^i move under the same rigid transformation as they are gripped by the same finger. However, the rigid transformation will result in nonlinear constraints unsuitable for a quadratic program. As such, we rely on rigidity rectification (Section 4.2) to enforce rigid motion and relax the rigid transformation requirement to linear transformation in the QP optimization:

$$(\mathbf{u}_j^{i+k} - \mathbf{c}^{i+k}) = \mathbf{L}^{i+k}(\hat{\mathbf{u}}_j^i - \mathbf{c}^i). \quad (3)$$

$\mathbf{c} \in \mathbb{R}^3$ is the center of mass of the vertices $\hat{\mathbf{u}}_j \in \mathcal{P}_o$. $\mathbf{L}^{i+k} \in \mathbb{R}^{3 \times 3}$ is the linear transformation of $\hat{\mathbf{u}}_j$ relative to \mathbf{c} from frame i to frame $i + k$. We solve for \mathbf{L} together with \mathbf{D} in the optimization.

The commanding forces also need to obey the laws of physics governing the cloth motion:

$$\mathbf{M}\Delta\dot{\mathbf{u}} = h\mathbf{G}(\mathbf{u}) + h\mathbf{f}_e + h\mathbf{f}_c, \quad (4)$$

where $\mathbf{M} \in \mathbb{R}^{3n \times 3n}$ is the mass matrix of cloth mesh, $\mathbf{u} \in \mathbb{R}^{3n}$ is the position vector of all the cloth vertices, \mathbf{G} includes bending and stretching forces of the cloth mesh, \mathbf{f}_e is the other external forces (e.g. gravity, cloth self-collision forces, or incidental contact forces with other objects in the scene), and h is the time step size ($h = 0.02s$ in our implementation). Note that \mathbf{G} is a nonlinear function of \mathbf{u} , but we can linearize \mathbf{G} about the current state \mathbf{u}^i to meet the requirements of quadratic programming:

$$\mathbf{M}\Delta\dot{\mathbf{u}}^{i+k} = h\mathbf{G}(\mathbf{u}^i) + h\frac{\partial\mathbf{G}}{\partial\mathbf{u}}|_{\mathbf{u}^i}(\mathbf{u}^{i+k} - \mathbf{u}^i) + h\mathbf{f}_e^{i+k} + h\mathbf{f}_c^{i+k}. \quad (5)$$

Because $\Delta\dot{\mathbf{u}}$ can be expressed as a linear function of \mathbf{f}_c via Equation 5, the cloth positions in the time window are also linear functions of \mathbf{f}_c via implicit Euler time integration:

$$\mathbf{u}^{i+k} = \mathbf{u}^i + kh\dot{\mathbf{u}}^i + h \sum_{j=0}^{k-1} (k-j)\Delta\dot{\mathbf{u}}^{i+1+j}. \quad (6)$$

4.1.2. Objective Function

In addition to physical feasibility, the optimization considers three objectives. The first objective is to closely follow the desired cloth motion described by the user-defined feature trajectories, $\bar{\phi}$, represented as b-splines:

$$E_f = \sum_{k=1}^K \|\phi(\mathbf{u}^{i+k}) - \bar{\phi}^{i+k}\|^2, \quad (7)$$

where $\phi(\mathbf{u}^{i+k})$ evaluates the features using the cloth configuration at frame $i + k$.

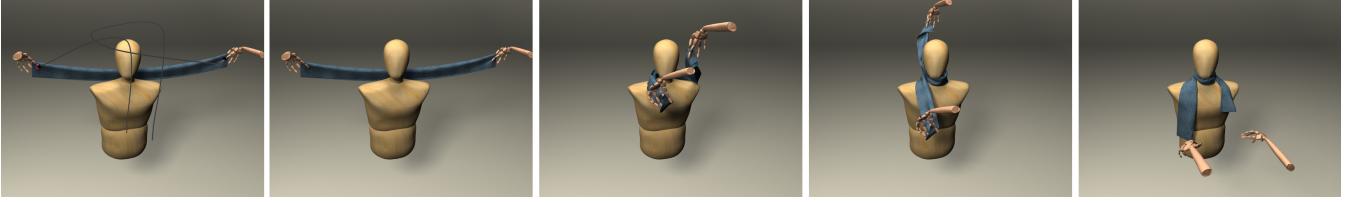


Figure 6: Put on a scarf.



Figure 7: Lift from a flat surface.

The second objective serves as a regularization term to minimize the magnitude of commanding force, preventing unnaturally large commanding forces generated by the human hand:

$$E_m = \sum_{k=1}^K \|\mathbf{f}_c^{i+k}\|^2. \quad (8)$$

We also wish to encourage the contact points in each \mathcal{P}_o to move smoothly:

$$E_s = \sum_{k=1}^K \|\dot{\mathbf{c}}^{i+k} - \dot{\mathbf{c}}^{i+k-1}\|^2 + \sum_{k=2}^K \|\mathbf{L}^{i+k} - \mathbf{L}^{i+k-1}\|^2 + \|\mathbf{L}^{i+1} - \mathbf{I}\|^2, \quad (9)$$

where \mathbf{I} is the identity matrix.

4.1.3. Formulating Quadratic Program

Finally, we formulate a quadratic program at each time step i as:

$$\begin{array}{ll} \underset{\mathbf{D}^{i+k}, \mathbf{L}^{i+k}, 1 \leq k \leq K}{\text{argmin}} & \omega_f E_f + \omega_m E_m + \omega_s E_s \\ \text{subject to} & \text{Equation 2, 3.} \end{array} \quad (10)$$

Once we solve \mathbf{D} , we can recover \mathbf{f}_c from Equation 1. Equation 5 is implicitly enforced as we express \mathbf{u}^{i+k} as a linear function of \mathbf{f}_c in the optimization. ω_f , ω_m , and ω_s are weights for each objective term, and are set to 40, 0.01, 10 respectively in our implementation.

Once we solve the commanding forces in the current window, we could simply apply the first commanding force \mathbf{f}_c^{i+1} on the cloth to advance the cloth to the next state $(\mathbf{u}^{i+1}, \dot{\mathbf{u}}^{i+1})$, and repeat this process to simulate a cloth sequence that achieves desired motion if our goal were to create cloth motion without consideration of the hands. In our case, however, instead of directly applying \mathbf{f}_c to the cloth, we need to compute the appropriate actuation of the hands to move the cloth in the same way as \mathbf{f}_c does via physical simulation with realistic contact forces.

4.2. Rigidity Rectification

Before we solve for the hand actuation, we need to make sure that the commanding forces from cloth motion planning are feasible for the hands. Equation 1, 2 and 3 improve the feasibility to some extent, but the relaxation of rigid transformation for formulating a QP can potentially cause undesired contact movements for the hands, i.e. the contact vertices $\hat{\mathbf{u}}_j \in \mathcal{P}_o$ on the same finger exhibit non-rigid motion impossible for the finger o to follow. The rigidity rectification step applies the matching algorithm by Horn et al. [Hor87] to factor out the rotation matrix $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ from \mathbf{L}^{i+1} solved in Equation 10 and uses it to rectify each $\hat{\mathbf{u}}_j^{i+1} \in \mathcal{P}_o^{i+1}$:

$$\hat{\mathbf{u}}_j^{i+1} = \mathbf{R}(\hat{\mathbf{u}}_j^i - \mathbf{c}^i) + \mathbf{c}^{i+1}. \quad (11)$$

4.3. Hand Control

With the rectified contact vertices $\hat{\mathbf{u}}^{i+1}$, we can now compute the actuation of the hands such that the corresponding contact points on the hands follow $\hat{\mathbf{u}}^{i+1}$. We first apply inverse kinematics to compute a desired hand pose $\bar{\mathbf{q}} \in \mathbb{R}^m$ to match $\hat{\mathbf{u}}^{i+1}$, where m denotes the number of degrees of freedom of the hand. Next, we solve the internal force $\boldsymbol{\tau}_{int}$ at each actuated degree of freedom on the hands using stable PD (SPD) formulation [TLT11]:

$$\ddot{\mathbf{q}} = -\mathbf{K}_p(\mathbf{q} + h\dot{\mathbf{q}} - \bar{\mathbf{q}}) - \mathbf{K}_d(\dot{\mathbf{q}} + h\ddot{\mathbf{q}}), \quad (12)$$

where $\mathbf{K}_p \in \mathbb{R}^{m \times m}$ and $\mathbf{K}_d \in \mathbb{R}^{m \times m}$ are diagonal matrices whose diagonal elements are set to 800 and 100 respectively. The acceleration of hand $\ddot{\mathbf{q}}$ can be evaluated via equations of motion:

$$\ddot{\mathbf{q}} = (\mathbf{M}_q + h\mathbf{K}_d)^{-1}(-\mathbf{C}_q - \mathbf{K}_p(\mathbf{q} + h\dot{\mathbf{q}} - \bar{\mathbf{q}}) - \mathbf{K}_d\dot{\mathbf{q}} + \boldsymbol{\tau}_{ext}), \quad (13)$$

where \mathbf{M}_q and \mathbf{C}_q denote the mass matrix, Coriolis and centrifugal force in generalized coordinates. $\boldsymbol{\tau}_{ext}$ includes all other external forces such as gravity. Please refer Tan et al. [TLT11] for details.

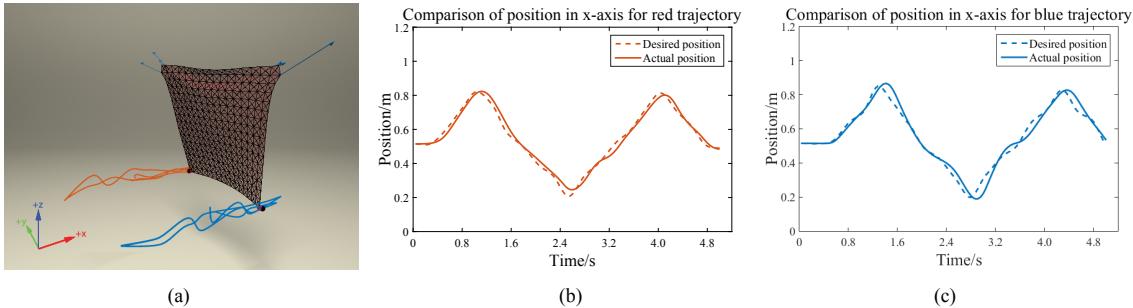


Figure 8: Evaluation of cloth motion planning. (a) We define two feature trajectories on the lower corners of a handkerchief while using two upper corners as controllable points. The optimal commanding forces are shown as blue arrows. (b,c) We plot the desired position (dashed line) and the actual position (solid line) in x-axis of the two bottom corners.

5. Grasp Controller

The grasp controller brings the hand from its initial state to the state in which the cloth is firmly grasped. The user provides the desired contact points on the cloth and two hand configurations: one for pre-shaping and one for gripping. Using these two input configurations, the algorithm solves inverse kinematics problems to generate two poses, \mathbf{q}_p and \mathbf{q}_g , that reach the desired contact points while maintaining the pre-shaping and gripping configurations respectively as close as possible. During simulation, the grasp controller computes the appropriate torques to move the hand from its arbitrary initial pose to \mathbf{q}_g through \mathbf{q}_p , using SPD tracking scheme described in Section 4.3.

The path from the initial pose to \mathbf{q}_p can potentially have collisions. The problem rarely occurs in our experiments because we focus on manipulating a single piece of cloth in an uncluttered environment. In the occasion of collision, additional keyframes of hand are added by the user to avoid undesired collisions. We use Catmull-Rom splines to interpolate hand keyframes to achieve smooth motion. If the collision issue occurs more frequently, an automatic path planning algorithm, such as Rapidly-exploring Random Tree (RRT) [LK01], can be implemented.

6. Evaluation

Our hand model is designed based on an anthropomorphic hand structure with 34 degrees of freedom. The simulator is built upon two open-source projects: ARCSim [NSO12, NPO13, PNdJO14] for cloth simulation and DART [LJ12] for multibody simulation. We couple the simulation of cloth and hands using the contact model proposed by Bai et al. [BL14b].

The window size of the optimization in Equation 10 is selected empirically. We start with the minimum window size and gradually increase it until the resulting motion is above the accuracy threshold. In theory, the linearized model will deteriorate as the window size becomes too large, but in practice the upper bound of the window size is often limited by the computation budget. A rough guideline to window size selection is based on how energetic the desired cloth motion is and the distance between controllable and feature vertices. As an example, we use the same input trajectories shown in Figure 8 to evaluate the accuracy of control with

different window sizes. As shown in Figure 9, the error between the desired position and the actual position decreases as the window size increases. When the window size grows beyond 12 frames, the improvement in accuracy becomes negligible.

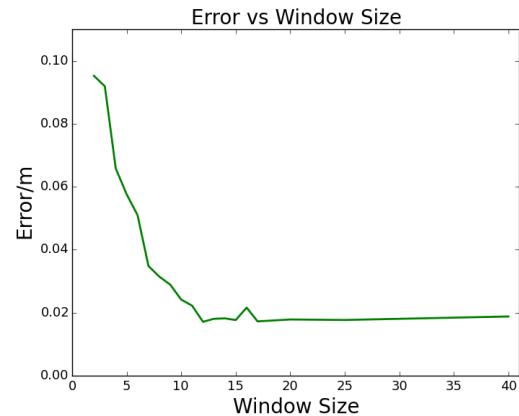


Figure 9: Evaluation of window size. We plot of the relation between the window size and the error for the input trajectories shown in Figure 8.

We evaluated our method by demonstrating a variety of manipulation tasks for real-world activities. In addition, we quantitatively tested the accuracy of the cloth motion planning algorithm on a random set of input feature trajectories. We also evaluated the generality of the cloth motion planning algorithm on a rope simulated with a different physical model from cloth.

6.1. Dexterous Manipulation Tasks

We applied our method to a variety of cloth manipulation scenarios shown in Figure 1 and in the accompanying video. The breakdown of computation time, the complexity of the input mesh, and the parameters of the method are listed in Table 1.

Shaking a handkerchief. This example demonstrates the ability of cloth motion planner to control dynamic secondary-motion

of cloth. Because the controllable points (upper two corners) and the feature points (bottom two corners) are far apart, the action of hands does not have direct impact on the motion of the features. Consequently, a seemingly simple feature trajectory can be very difficult to control using the commanding forces from the hands. We demonstrate that our algorithm is able to control two feature trajectories that move asynchronously. To validate our algorithm on a more dynamic motion, we synthesized a sequence where the bottom corners follow a circle around the hands (Figure 4). This motion requires the commanding force to be large enough to overcome gravity but not too large to outpace the timing specified by the trajectories. Finally, we demonstrate that our algorithm is able to handle unexpected external forces robustly as one of the advantages of model-predictive-control. In this example, we added a wind force to the simulation, causing the commanding forces from the hands to compensate.

Wringing a towel. Wringing a towel presents more complex contacts between the fingers/palm and the cloth. Nevertheless, the feature specification can be two simple circular trajectories moving in the opposite directions in the middle part of the towel (Figure 5). Because grasping on a bunched up cloth configuration results in a large number of contact points scattered in a wide range on the cloth, the linear transformation and rigidity rectification are crucial to ensure that the contact points are achievable by the hand.

Folding a T-shirt. Folding a T-shirt in our example requires three contact phases (Figure 2). We show that tasks with regrasping can be achieved by executing the same algorithm multiple times. Comparing to shaking a handkerchief, folding a T-shirt is an easier example because the features are close in distance to the controlling contact points. For this case, we used small time window for optimization ($K = 3$), and our algorithm can track the feature trajectories very accurately (mean error is 0.003m).

Putting on a scarf. Putting a scarf on a mannequin causes additional collision forces on the cloth. Because our algorithm considers all other external forces in cloth motion planning (Equation 5), the hands are able to overcome the collision and track the desired trajectories closely (Figure 6).

Lifting from a flat surface. A common strategy to pick up a thin layer of cloth lying flat on a surface is to first use fingers to generate opposing friction forces to create a bulge on the cloth, and grab the bulge to lift the cloth. The feature trajectories include two points on the cloth moving towards middle and four points in the middle of the cloth moving upward (Figure 7). To achieve this task, the thumb and the index finger make two disjoint contact areas with the cloth and move asynchronously towards each other. This example demonstrates that every finger can have its own contact area with the cloth moving in a different direction from other fingers.

6.2. Evaluation of Cloth Motion Planning

To validate our algorithm of cloth motion planning, we tested our algorithm on a set of randomly sampled feature trajectories shown in the accompanying video. Each trajectory is represented as a b-spline with 22 control points. The space for each trajectory is parameterized by the position and the velocity of the b-spline control points. In our experiments, we defined two feature trajectories

on the lower corners of a handkerchief while using two upper corners as controllable points. Starting from two asynchronous straight lines as the “mean trajectories”, we drew samples from a normal distribution with different standard deviations (SD) in the space of b-spline control points.

Because the evaluation only focuses on cloth motion planning and requires a large amount of testing sequences, we made two simplifications in the experiments. First, we applied the optimal commanding force \mathbf{f}_c solved from Equation 10 directly to update the state of the cloth, excluding the involvement of hands in this evaluation. Second, we used a piece of cloth with lower resolution to speed up the simulation time. However, the challenge of controlling cloth remains because the number of uncontrolled degrees of freedom is still much more than the controlled ones (shown as blue arrows in Figure 8 (a)). Furthermore, we chose an example where the feature points are very far from controllable points.

As shown in the accompanying video, the feature points (shown as red dots) follow the desired trajectories closely. To evaluate the accuracy more precisely, we measured the Euclidean distance between the actual position and the desired position of a feature point. Figure 8 shows the comparison of positions in x-axis for one testing sequence. For an input feature trajectory that has a known control solution (i.e. the feature trajectory is generated by physical simulation), our motion planning algorithm can achieve mean error less than 0.01m over time. For randomly sampled feature trajectories, the mean error of the distance is 0.016m. The dimension of the cloth is a 0.5m by 0.5m square. For completeness, we also applied the entire algorithm including hand control on a selected subset of testing sequences. Please see the accompanying video for visual evaluation.

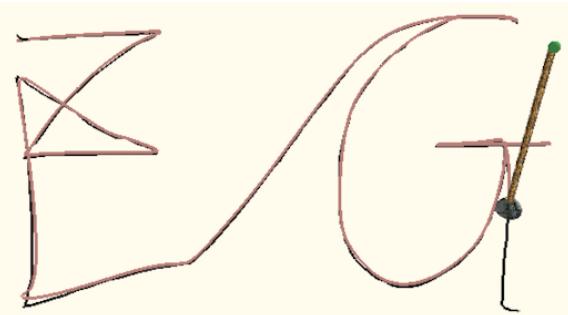


Figure 10: Our cloth motion planning algorithm is applied on a rope to control the actual trajectory of the ball at the bottom (red curve) to track a desired trajectory drawn by the user (black curve). The commanding force is applied at the top of the rope (green dot).

Our cloth motion planning algorithm is independent of the choice of simulators (e.g. ARCSim). To validate the generality of our algorithm, we tested it on a volumetric mesh simulated as a mass-spring system. Figure 10 shows a rope with a ball hung at the bottom to track a desired trajectory provided by the user. The desired commanding force is solved for the top of the rope (green dot) with our cloth motion planning algorithm, so that the actual motion trajectory of the ball (red curve) can follow the desired trajectory (black curve).

Example	#Triangle	#Vertex	Window size	Animation time	Optimization time	Control time	Simulation time
Shaking a handkerchief (asynchronous trajectories)	8192	4225	20	5s	1h56m	37s	21m
Shaking a handkerchief (circle trajectories)	8192	4225	20	5s	3h2m	40s	33m
Shaking a handkerchief (with wind force)	8192	4225	10	6s	27m	45s	24m
Wringing a towel	4078	2071	10	4s	1h18m	1m2s	18m
Folding a T-shirt	26252	13256	3	13s	1h	1m40s	5h14m
Putting on a scarf	11831	6077	3	3.5s	9m	25s	18m
Lifting from a flat surface	8525	4392	10	1.3s	9m	6s	5m

Table 1: Parameters and performance of the examples. #Triangle: the number of triangles in the input mesh. #Vertex: the number of vertices in the input mesh. Window size: window size K used in the optimization. Animation time: wall clock time of the animation. Optimization time: total time for formulating and solving the optimization in cloth motion planning. Control time: total time for solving hand control. Simulation time: total time for cloth and multibody simulation.

7. Limitation

The decision on decoupling the hand and the cloth in the optimization is necessary for performance efficiency, but it also introduces some limitations. In particular, the kinematic constraints of the hand can be approximated in different ways. It could be more conservative than our current method by limiting the whole hand to move as one rigid body. On the other hand, it could be more aggressive by assuming each finger segment can move independently. We choose a compromised approximation by which each finger can move independently. This approximation is reasonable for the examples demonstrated in this paper, but it can occasionally result in cloth motion plans unachievable by hands (e.g. two fingers need to move too far apart).

Our controller is conservative in that it tries to maintain static contact and not exploit slipping or rolling manipulation strategies. We show in our result that this control strategy is sufficient to achieve a wide range of cloth manipulation tasks in daily life. However, for more general cloth manipulation tasks, such as using hands to slide across the cloth to remove wrinkles, the slipping manipulation strategy is important.

We take a statistic approach to evaluate the accuracy of cloth motion planning because analytically defining the successful range of feature trajectories is very difficult if at all possible. This statistical evaluation gives us some confidence that our algorithm is able to produce plausible results for a wide range of input feature trajectories. However, we still can neither guarantee nor predict whether an arbitrary feature trajectory will result in successful dexterous manipulation.

8. Conclusion

We introduce a method to enable detailed dexterous manipulation of cloth for physics-based computer animation. Our technique provides a solution to consolidate the control of hand manipulation and the control of cloth simulation. We formulate an optimization problem that solves the commanding forces with respect to the desired cloth motion described by the user and the constraints of the

hands. The evaluation shows that our cloth motion planning can accurately achieve the user-specifications on the cloth and can be successfully executed by the simulated hands. We also provide visual demonstration of our technique on a set of cloth manipulation tasks including folding laundry, wringing a towel, and putting on a scarf. We envision that this technique can be integrated with physically simulated full-body characters.

References

- [BC10] BALAGUER B., CARPIN S.: Motion planning for cooperative manipulators folding flexible planar objects. In *IROS* (2010), pp. 3842–3847. [2](#)
- [BL14a] BAI Y., LIU C.: Dexterous manipulation using both palm and fingers. In *Proc. 2014 IEEE International Conference on Robotics and Automation* (2014), IEEE. [2](#)
- [BL14b] BAI Y., LIU C. K.: Coupling cloth and rigid bodies for dexterous manipulation. In *Proceedings of the Seventh International Conference on Motion in Games* (2014), ACM, pp. 139–145. [2, 6](#)
- [BMWG07] BERGOU M., MATHUR S., WARDETZKY M., GRINSPUN E.: Tracks: toward directable thin shells. In *ACM Trans. Graph. (SIGGRAPH 2007)* (2007), vol. 26, ACM, p. 50. [3](#)
- [BPK11] BERSCH C., PITZER B., KAMMEL S.: Bimanual robotic cloth manipulation for laundry folding. In *Proc. 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2011), IEEE, pp. 1413–1419. [2](#)
- [BSG12] BARBIĆ J., SIN F., GRINSPUN E.: Interactive editing of deformable simulations. In *ACM Trans. Graph. (SIGGRAPH 2012)* (2012), vol. 31, ACM, p. 70. [3](#)
- [CKHL11] CHOI M. G., KIM M., HYUN K. L., LEE J.: Deformable motion: Squeezing into cluttered environments. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 445–453. [2](#)
- [CTSM*11] CUSUMANO-TOWNER M., SINGH A., MILLER S., O'BRIEN J. F., ABBEEL P.: Bringing clothing into desired configurations with limited perception. In *Proc. 2011 IEEE International Conference on Robotics and Automation* (2011), IEEE, pp. 3893–3900. [2](#)
- [CTTL15] CLEGG A., TAN J., TURK G., LIU C. K.: Animating human dressing. In *ACM Trans. Graph. (SIGGRAPH 2015)* (2015), vol. 34, ACM. [3](#)
- [GRS*07] GAYLE R., REDON S., SUD A., LIN M. C., MANOCHA D.: [3](#)

- Efficient motion planning of highly articulated chains using physics-based sampling. In *Robotics and Automation, 2007 IEEE International Conference on* (2007), IEEE, pp. 3319–3326. [2](#)
- [GSLM05] GAYLE R., SEGARS P., LIN M. C., MANOCHA D.: Path planning for deformable robots in complex environments. In *Robotics: science and systems* (2005), vol. 2005, Citeseer, pp. 225–232. [2](#)
- [HK09] HO E. S., KOMURA T.: Character motion synthesis by topology coordinates. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 299–308. [3](#)
- [Hor87] HORN B. K.: Closed-form solution of absolute orientation using unit quaternions. *JOSA A* 4, 4 (1987), 629–642. [5](#)
- [HSvTP12] HILDEBRANDT K., SCHULZ C., VON TYCOWICZ C., POLTHIER K.: Interactive spacetime control of deformable objects. *ACM Trans. Graph. (SIGGRAPH) 2012* 31, 4 (2012), 71. [3](#)
- [JC07] JEON H., CHOI M.-H.: Interactive motion control of deformable objects using localized optimal control. In *Robotics and Automation, 2007 IEEE International Conference on* (2007), IEEE, pp. 2582–2587. [3](#)
- [Jim12] JIMÉNEZ P.: Survey on model-based manipulation planning of deformable objects. *Robotics and computer-integrated manufacturing* 28, 2 (2012), 154–163. [2](#)
- [JV03] JANSSON J., VERGEEST J. S.: Combining deformable-and rigid-body mechanics simulation. *The Visual Computer* 19, 5 (2003), 280–290. [2](#)
- [KP06] KRY P. G., PAI D. K.: Interaction capture and synthesis. In *ACM Trans. Graph. (SIGGRAPH 2006)* (2006), vol. 25, ACM, pp. 872–880. [2](#)
- [Liu08] LIU C. K.: Synthesis of interactive hand manipulation. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2008), Eurographics Association, pp. 163–171. [2](#)
- [Liu09] LIU C. K.: Dextrous manipulation from a grasping pose. In *ACM Trans. Graph. (SIGGRAPH 2009)* (2009), vol. 28. [2](#)
- [LJ12] LIU C. K., JAIN S.: *A Short Tutorial on Multibody Dynamics*. Tech. Rep. GIT-GVU-15-01-1, Georgia Institute of Technology, School of Interactive Computing, 08 2012. [6](#)
- [LK01] LAVALLE S. M., KUFFNER J. J.: Randomized kinodynamic planning. *The International Journal of Robotics Research* 20, 5 (2001), 378–400. [6](#)
- [MFS14] MIGUEL E., FENG A., SHAPIRO A.: Towards cloth-manipulating characters. In *Computer Animation and Social Agents 2014* (2014). [3](#)
- [MK06] MOLL M., KAVRAKI L. E.: Path planning for deformable linear objects. *Robotics, IEEE Transactions on* 22, 4 (2006), 625–636. [2](#)
- [MLM08] MOSS W., LIN M. C., MANOCHA D.: Constraint-based motion synthesis for deformable models. *Computer Animation and Virtual Worlds* 19, 3-4 (2008), 421–431. [2](#)
- [MO11] MIGUEL E., OTADUY M. A.: Efficient simulation of contact between rigid and deformable objects. *Multibody Dynamics* (2011). [2](#)
- [MPT12] MORDATCH I., POPOVIĆ Z., TODOROV E.: Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation* (2012), Eurographics Association, pp. 137–144. [2](#)
- [MVDBF*12] MILLER S., VAN DEN BERG J., FRITZ M., DARRELL T., GOLDBERG K., ABBEEL P.: A geometric approach to robotic laundry folding. *The International Journal of Robotics Research* 31, 2 (2012), 249–267. [2](#)
- [NPO13] NARAIN R., PFAFF T., O'BRIEN J. F.: Folding and crumpling adaptive sheets. *ACM Trans. Graph. (SIGGRAPH 2013)* 32, 4 (2013), 51. [6](#)
- [NSO12] NARAIN R., SAMII A., O'BRIEN J. F.: Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. Graph. (SIGGRAPH Asia 2012)* 31, 6 (2012), 152. [6](#)
- [OSK07] OSAWA F., SEKI H., KAMIYA Y.: Unfolding of massive laundry and classification types by dual manipulator. *JACIII* 11, 5 (2007), 457–463. [2](#)
- [OTSG09] OTADUY M. A., TAMSTORF R., STEINEMANN D., GROSS M.: Implicit contact handling for deformable objects. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 559–568. [2](#)
- [PGB14] PHILLIPS-GRAFFLIN C., BERENSON D.: Path planning and execution for deformable objects using a voxel-based representation. In *European Workshop on Deformable Object Manipulation* (2014). [2](#)
- [PNdJO14] PFAFF T., NARAIN R., DE JOYA J. M., O'BRIEN J. F.: Adaptive tearing and cracking of thin sheets. In *ACM Trans. Graph. (SIGGRAPH 2014)* (2014), vol. 33, ACM, p. 110. [6](#)
- [PvdBA11] PATIL S., VAN DEN BERG J., ALTEROVITZ R.: Motion planning under uncertainty in highly deformable environments. In *Proceedings of Robotics: Science and Systems* (Los Angeles, CA, USA, June 2011). [2](#)
- [PZ05] POLLARD N. S., ZORDAN V. B.: Physically based grasping control from example. In *Proc. 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2005), ACM, pp. 311–318. [2](#)
- [RLA06] RODRIGUEZ S., LIEN J.-M., AMATO N. M.: Planning motion in completely deformable environments. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on* (2006), IEEE, pp. 2466–2471. [2](#)
- [RPCR12] ROSELL J., PÉREZ A., CABRAS P., ROSELL A.: Motion planning for the virtual bronchoscopy. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on* (2012), IEEE, pp. 2932–2937. [2](#)
- [RWE*09] RILK M., WAHL F. M., EICHORN K. W., WAGNER I., BOOTZ F.: Path planning for robot-guided endoscopes in deformable environments. In *Advances in Robotics Research*. Springer, 2009, pp. 263–274. [2](#)
- [SI06] SAHA M., ISTO P.: Motion planning for robotic manipulation of deformable linear objects. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on* (2006), IEEE, pp. 2478–2484. [2](#)
- [SI07] SAHA M., ISTO P.: Manipulation planning for deformable linear objects. *Robotics, IEEE Transactions on* 23, 6 (2007), 1141–1150. [2](#)
- [SKP08] SUEDA S., KAUFMAN A., PAI D. K.: Musculotendon simulation for hand animation. In *ACM Trans. Graph. (SIGGRAPH 2008)* (2008), vol. 27, ACM, p. 83. [2](#)
- [SSB*15] SACHDEVA P., SUEDA S., BRADLEY S., FAIN M., PAI D. K.: Biomechanical simulation and control of hands and tendinous systems. In *ACM Trans. Graph. (SIGGRAPH 2015)* (2015), vol. 34, ACM. [2](#)
- [SSF08] SHINAR T., SCHROEDER C., FEDKIW R.: Two-way coupling of rigid and deformable bodies. In *Proc. 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2008), Eurographics Association, pp. 95–103. [2](#)
- [SIF07] SIFAKIS E., SHINAR T., IRVING G., FEDKIW R.: Hybrid simulation of deformable solids. In *Proc. 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), Eurographics Association, pp. 81–90. [2](#)
- [SVAC12] SAN-VICENTE G., AGUINAGA I., CELIGÜETA J. T.: Cubical mass-spring model design based on a tensile deformation test and nonlinear material model. *Visualization and Computer Graphics, IEEE Transactions on* 18, 2 (2012), 228–241. [2](#)
- [TLT11] TAN J., LIU K., TURK G.: Stable proportional-derivative controllers. *Computer Graphics and Applications, IEEE* 31, 4 (2011), 34–44. [5](#)
- [VDBMGA11] VAN DEN BERG J., MILLER S., GOLDBERG K., ABBEEL P.: Gravity-based robotic cloth folding. In *Algorithmic Foundations of Robotics IX*. Springer, 2011, pp. 409–424. [2](#)

[WK12] WANG H., KOMURA T.: Manipulation of flexible objects by geodesic control. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 499–508. [3](#)

[WMT06] WOJTCZAK C., MUCHA P. J., TURK G.: Keyframe control of complex particle systems using the adjoint method. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2006), Eurographics Association, pp. 15–23. [3](#)

[WMZ*13] WANG Y., MIN J., ZHANG J., LIU Y., XU F., DAI Q., CHAI J.: Video-based hand manipulation capture through composite motion control. In *ACM Trans. Graph. (SIGGRAPH 2013)* (2013), vol. 32, ACM, p. 43. [2](#)

[WSSK13] WANG H., SIDOROV K. A., SANDILANDS P., KOMURA T.: Harmonic parameterization by electrostatics. *ACM Transactions on Graphics (TOG)* 32, 5 (2013), 155. [3](#)

[YL12] YE Y., LIU C. K.: Synthesis of detailed hand manipulations using contact sampling. In *ACM Trans. Graph. (SIGGRAPH 2012)* (2012), vol. 31, ACM, pp. 1–10. [2](#)

[ZZMC13] ZHAO W., ZHANG J., MIN J., CHAI J.: Robust realtime physics-based motion control for human grasping. In *ACM Trans. Graph. (SIGGRAPH ASIA 2013)* (2013), vol. 32, ACM, p. 207. [2](#)