

Coupling Cloth and Rigid Bodies for Dexterous Manipulation

Yunfei Bai*
Georgia Institute of Technology

C. Karen Liu†
Georgia Institute of Technology



Figure 1: *Dexterous manipulation of cloth.*

Abstract

This paper introduces a new simulation technique to enable detailed dexterous manipulation of cloth. Without reimplementation or substantial modification, existing cloth simulators can only be used to approximate limited interaction between cloth and rigid bodies due to the incorrect computation of contact forces. For example, a simple scenario of two fingers pinching a piece of cloth often results in the cloth slipping out of the hand. Our technique provides a simple solution to cloth-rigid coupling using existing cloth and rigid body simulators as-is. We develop a light-weight interface so that the rigid body and cloth simulators communicate on a demand-driven manner to achieve two main goals: allow the rigid bodies to impart friction forces to the cloth and avoid unsolvable collision situations between the rigid bodies and the cloth. We demonstrate a set of basic manipulation skills including gripping, pinching, and pressing, that are frequently seen in daily activities such as dressing and folding clothes.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: cloth simulation, physically based animation, hand animation

1 Introduction

Manipulating cloth is an important skill of daily living. From getting dressed, to folding laundry, many essential daily tasks depend on complex dynamic interaction between our hands and cloth. Unlike manipulation of rigid bodies, which at its heart is a problem of control and planning, synthesizing manipulation of cloth presents additional challenges in physics simulation. Currently there is a lack of an accurate, efficient, and unified simulator that simulates complex interaction between the hands and cloth.

Realistic cloth simulation has been extensively demonstrated in computer animation, but the interaction between cloth and rigid

bodies is usually limited to simple cases, such as a shirt lying on a mannequin or a handkerchief falling on an object. In current cloth simulators (e.g. Maya [MAY] or ARCSim [ARC]), the algorithm for coupling the cloth and the rigid body is not capable of simulating accurate interactions when rigid bodies impart forces to the cloth. As an illustrative example, current cloth simulators are unable to hold a piece of cloth, no matter how hard the hand grips the cloth (Figure 2). This example reveals an important issue of current simulators; the contact force computation only depends on kinematic information (position and velocity) of the rigid body, but dynamic information, such as inertial forces and other applied forces on the rigid body is ignored. A second important issue of current simulators is that rigid body motion is usually simulated or scripted separately from the cloth; the rigid bodies are oblivious to the existence of the cloth. In an example of two fingers pinching a piece of cloth, without the awareness of the cloth’s current state, the space between two fingers might not be sufficient for the cloth simulator to resolve collisions, leading to unstable and unappealing cloth motion. This paper introduces a new simulation technique

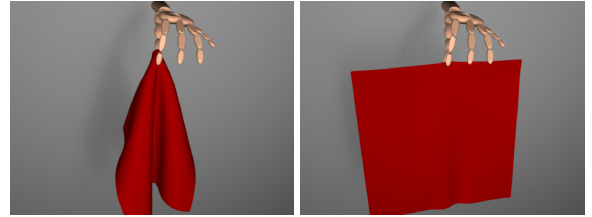


Figure 2: *Pinching grasp. Left: A piece of cloth is held by the thumb and the index finger using our contact force computation. Right: The cloth slips out of the hand in ARCSim simulation.*

to enable detailed dexterous manipulation of cloth, based on *pre-existing* cloth and rigid body simulators. Our algorithm facilitates communication between two simulators and computes crucial information for physical interaction. To illustrate how our technique improves upon current cloth simulators, we characterize different coupling methods between the cloth and rigid bodies in the following table:

The columns indicate the information passing from the rigid body to the cloth (i.e. $\mathbf{r} \rightarrow \mathbf{c}$), while the rows indicate the information from the cloth to the rigid body (i.e. $\mathbf{c} \rightarrow \mathbf{r}$). To achieve accurate two-way coupling between two dynamic systems, the dynamic information needs to be passed on in both directions (i.e. bottom-right cell). Most state-of-the-art cloth simulators belong to the first row and the second column, because they only utilize the kinematic information of the rigid body and assume that the rigid body

*e-mail:ybai30@mail.gatech.edu

†e-mail:karenliu@cc.gatech.edu

	$\mathbf{r} \rightarrow \mathbf{c}$ None	$\mathbf{r} \rightarrow \mathbf{c}$ Kinematics	$\mathbf{r} \rightarrow \mathbf{c}$ Dynamics
$\mathbf{c} \rightarrow \mathbf{r}$ None		ARCSim	
$\mathbf{c} \rightarrow \mathbf{r}$ Kinematics			Our work
$\mathbf{c} \rightarrow \mathbf{r}$ Dynamics			2-way couple

is unaware of the cloth. Our work strives to improve upon those cloth simulators. However, we argue that a true two-way coupling scenario is unnecessary for the applications of hand manipulation because the dynamic effects to the hand caused by the cloth are negligible. On the other hand, achieving correct two-way coupling involves redefining the state space and reformulating equations of motion, which requires a large amount of effort on reimplementation of existing simulators. Consequently, we believe that the “sweet-spot” for our applications is a coupling technique located at the second row and the third column. Our technique builds a lightweight rigid body interface that we call *rigid cloth patch*, which is a local representation of cloth in contact with a rigid body. Through rigid cloth patch, the rigid body and the cloth simulators communicate on a demand-driven manner in achieving two main goals: allowing rigid bodies to impart friction forces on cloth and avoiding unsolvable collision situations between the rigid bodies and the cloth. We pass contact forces computed in the rigid body simulator to the cloth simulator so that proper friction forces can be applied. Conversely, we inform the rigid body simulator of the state of cloth by using dynamic rigid cloth patches to approximate the geometry of the cloth.

Our technique adds very little computational overhead and requires minimal modification of pre-existing simulators. Our implementation is built upon two open-source projects: ARCSim for cloth simulation and DART [DAR] for multibody simulation. We demonstrate a set of basic manipulation skills, including gripping, pinching, and pressing; that are frequently seen in daily activities such as dressing and folding clothes. Moreover, we compare our technique to the state-of-the-art cloth simulators and our technique shows favorable results.

2 Related Work

Dexterous manipulation is a broad research area that has a variety of applications in computer graphics and robotics. Although a precise definition is still open to interpretation, dexterous manipulation is typically defined as the use of multiple fingers to achieve a desired object configuration. In computer graphics, researchers have shown that intricate control strategies, such as finger gaiting [Ye and Liu 2012], rolling/sliding [Liu 2009; Bai and Liu 2014], or grasping/regrasping [Pollard and Zordan 2005; Kry and Pai 2006; Zhao et al. 2013; Wang et al. 2013], can be physically simulated on an anthropomorphic hand model. However, one of the most important assumptions these techniques make is that the manipulated objects are rigid bodies with only six degrees of freedom. Manipulating deformable objects is a more challenging problem due to more degrees of freedom and more complex collision phenomena. Researchers in robotics have studied the problems of manipulating fabric, cables, foam rubber, or sheet metal [Kosuge et al. 1995; Wu et al. 1995; Fahantidis et al. 1997; Osawa et al. 2007; Bersch et al. 2011; Miller et al. 2012]. Many previous approaches enhance control and planning algorithms by using simulation techniques to estimate the state of the deformable objects. For example, robotics researchers used a cloth simulator to approximate the contour of the clothes being folded by a PR2 robot [Cusumano-Towner et al.

2011]. Because the interaction between the robot hands and the cloth was relatively simple (i.e. grasping only), their simulation applied position constraints to pin the cloth in the air instead of simulating the hands. Our work aims to simulate the hands, the cloth, and their effects to each other realistically. We show that with a more accurate simulation routine, a wide variety of manipulation strategies can be achieved.

Because our work is directly built upon the open source cloth simulator, ARCSim, we will describe a subset of research in cloth simulation that led to the techniques used by ARCSim. Since the seminal work by Terzopolous [Terzopoulos et al. 1987], many techniques have been proposed for cloth simulation [Baraff and Witkin 1998; Müller et al. 2007; Kaldor et al. 2010; Liu et al. 2013]. ARCSim adapts the elastic model formulation from several different approaches [Bridson et al. 2003; Grinspun et al. 2003; Müller and Gross 2004; Wang et al. 2011] and used different algorithms for collision handling. There has been a rich body of research on contact and collision for cloth simulation [Bridson et al. 2002; Baraff et al. 2003; Harmon et al. 2008; Kaufman et al. 2008; Otaduy et al. 2009; Harmon et al. 2009; Tang et al. 2010; Miguel and Otaduy 2011; Ainsley et al. 2012]. The core algorithm for collision handling in ARCSim is based on Bridson *et al.* [Bridson et al. 2002], who uses a geometry-based repulsive impulse to handle collision robustly. In addition, ARCSim improves the quality of the simulation by adapting inelastic projection for resolving simultaneous collisions [Harmon et al. 2008]. ARCSim also takes advantage of remeshing techniques to improve computational speed and simulation quality. Narain *et al.* proposed an adaptive anisotropic remeshing technique to dynamically refine and coarsen triangle meshes in simulation [Narain et al. 2012; Narain et al. 2013]. Though ARCSim is able to produce very compelling visual results with a relatively robust simulation process, it is not capable of simulating accurate interactions when a pair of rigid bodies impart forces on both sides of a piece of cloth. This is a crucial requirement for dexterous manipulation of cloth. Our technique enhances ARCSim by coupling rigid bodies and cloth in a more accurate fashion, while making minimal modification to its source code.

Two-way coupling of dynamic systems is an important research problem in computer animation. While graphics practitioners have demonstrated cloth-rigid interaction by treating a rigid body as a very rigid cloth using commercial cloth simulators (e.g. [MAY]), researchers have proposed various methods to handle two-way coupling between rigid bodies and deformable bodies [Jansson and Vergeest 2003; Sifakis et al. 2007; Shinar et al. 2008; Otaduy et al. 2009; Miguel and Otaduy 2011]. Otaduy et al. [Otaduy et al. 2009] solved contacts between cloth and rigid bodies by implicitly solving a large mixed linear complementarity problem (LCP). This requires accessing internal dynamic states of the simulators, such as the mass matrix, force vector and its derivative, and contact constraints. Our work treats existing cloth and rigid body simulators as black boxes without altering the internal formulation of collision handling. We solve a local nonlinear complementarity problem to compute contact forces. In conjunction with the proposed cloth rigid patches, our method is able to resolve the collision between cloth and rigid body in the pinched situation. Tackling the two-way coupling problem accurately requires formulating a unified set of equations of motion for both systems, and solving for the contact forces and the next state simultaneously. Given the requirements of our application, we chose a simpler and less involved approach to the coupling problem.

The physics-based simulation of interactions between hands and deformable objects can also be used in game applications and haptics research [James and Pai 2001; Magnenat-Thalmann et al. 2007]. Our method couples two dynamic systems without adding significant computation time to the simulation. If the cloth and rigid

body simulators perform in real-time, the coupled system can also achieve real-time using our technique. Therefore, our technique has potential to be used in game applications.

3 Algorithm Overview

Our method is independent from the integration scheme and contact algorithm used in the simulators. We utilize the contact force from the rigid body simulator to compute the cloth contact, and we take the state of the cloth in cloth simulator to create rigid cloth patches. Our algorithm is implemented with existing rigid body simulator DART, and cloth simulator ARCSim. Although we choose to implement our technique with these two simulators, we emphasize that our algorithm also applies to other physics engine, such as ODE [ODE] or Bullet [Bul], as long as the state of the rigid body, the state of the cloth, and rigid body contact forces can be accessed by the APIs of the physics engine. The following pseudo code (Algorithm 1) illustrates the simulation procedure from the current state of the cloth ($\mathbf{x}^n, \mathbf{v}^n$) (position and velocity) and of the rigid bodies ($\mathbf{q}^n, \dot{\mathbf{q}}^n$) to the next state. We use \mathbf{q} to represent the generalized coordinates of both the hand model and the rigid cloth patches, which are used to approximate the current state of the cloth (Section 5). The contributions of this paper are highlighted in blue.

```

for every time step  $n$  do
   $\bar{\mathbf{v}}^{n+1} \leftarrow \text{ClothDynamics}(\mathbf{x}^n, \mathbf{v}^n)$ ;
   $\mathcal{V}_c, \mathcal{V}_r \leftarrow \text{ClothCollisionDetection}(\mathbf{x}^n, \bar{\mathbf{v}}^{n+1}, \mathbf{q}^n, \dot{\mathbf{q}}^n)$ ;
  for every cloth vertex  $i$  in  $\mathcal{V}_c$  or  $\mathcal{V}_r$  do
    if in  $\mathcal{V}_r$  then
       $\mathbf{f}_N \leftarrow \text{RigidContactForce}(\mathbf{f}_c^i)$ ; // Section 4
    end
    else
       $\mathbf{f}_N \leftarrow \text{ClothCollision}(\mathbf{x}^n, \bar{\mathbf{v}}^{n+1})$ ;
    end
     $\bar{\mathbf{v}}_i^{n+1} \leftarrow \text{FrictionForce}(\mathbf{x}^n, \bar{\mathbf{v}}^{n+1}, \mathbf{q}^n, \dot{\mathbf{q}}^n, \mathbf{f}_N)$ ; // Section 4
     $\bar{\mathbf{x}}_i^{n+1} = \mathbf{x}_i^n + \Delta t \bar{\mathbf{v}}_i^{n+1}$ ;
    if staticFriction = TRUE then
       $\mathbf{c}_i \leftarrow \text{PositionConstraint}(\bar{\mathbf{x}}_i^{n+1})$ ;
    end
  end
   $\hat{\mathbf{x}}^{n+1}, \hat{\mathbf{v}}^{n+1} \leftarrow \text{StrainLimit}(\bar{\mathbf{x}}^{n+1}, \mathbf{c})$ ;
   $\check{\mathbf{x}}^{n+1}, \check{\mathbf{v}}^{n+1} \leftarrow \text{InelasticProjection}(\hat{\mathbf{x}}^{n+1}, \hat{\mathbf{v}}^{n+1}, \mathbf{q}^n, \dot{\mathbf{q}}^n, \mathbf{c})$ ;
   $\mathbf{x}^{n+1}, \mathbf{v}^{n+1} \leftarrow \text{ClothRemeshing}(\check{\mathbf{x}}^{n+1}, \check{\mathbf{v}}^{n+1}, \mathbf{q}^n, \dot{\mathbf{q}}^n)$ ;
   $\bar{\mathbf{q}}^{n+1}, \dot{\bar{\mathbf{q}}}^{n+1} \leftarrow \text{UpdateRigidClothPatches}(\mathbf{x}^{n+1}, \mathbf{v}^{n+1}, \mathcal{V}_r)$ ; // Section 5
   $\tau \leftarrow \text{ControlForce}(\bar{\mathbf{q}}^{n+1}, \dot{\bar{\mathbf{q}}}^{n+1})$ ; // Section 6
   $\mathbf{q}^{n+1}, \dot{\mathbf{q}}^{n+1}, \mathbf{f}_c^{n+1} \leftarrow \text{RigidSimulation}(\bar{\mathbf{q}}^{n+1}, \dot{\bar{\mathbf{q}}}^{n+1}, \tau)$ ;
end

```

Algorithm 1: Cloth and Rigid Simulation Algorithm

The time-stepping function begins with advancing the cloth to the candidate velocity $\bar{\mathbf{v}}^{n+1}$ by computing the cloth dynamics in the standard way [Bridson et al. 2003; Grinspun et al. 2003; Müller and Gross 2004; Wang et al. 2011]. We then apply the collision detection algorithm in ARCSim to identify a set of cloth vertices \mathcal{V}_r colliding with rigid bodies and another set \mathcal{V}_c colliding with the cloth itself. The algorithm first determines the normal contact force \mathbf{f}_N for each vertex. If a cloth vertex i is in \mathcal{V}_r , we use the normal contact forces \mathbf{f}_c^i , computed by the rigid body simulator in the previous time step. Otherwise, we use the repulsive force computed by the cloth simulator to be its \mathbf{f}_N . Next, we compute the friction force based on Coulomb’s friction model (Section 4). To ensure

that the static contact points are not subject to changes made by the following refinement steps in the cloth simulator, i.e. strain limit, inelastic projection and remeshing, we enforce position constraints on those static vertices. After the cloth is advanced to its next state, we update the state of the rigid cloth patches to approximate the new state of the cloth (Section 5). Finally, we compute the control force for the hand (Section 6) and advance the rigid body simulator to the next time step.

4 Contact Force

Computing correct contact forces from the hand to the cloth is crucial for many dexterous manipulation applications. Most current cloth simulators (e.g. [ARC; MAY]) compute contact forces based only on the position and the velocity of the rigid body. The consequence of an inaccurate contact force is particularly evident when the rigid body is not interpenetrating the cloth but is imparting force on the cloth (e.g. two fingers pinch-grasp a piece of cloth). We describe a new method for handling contact forces successfully in the above situations.

4.1 Normal Force

If a cloth vertex i is marked as being in contact with one or more rigid bodies (i.e. in the set \mathcal{V}_r), we compute its normal contact force using the collision handling routine in the rigid body simulator. DART solves the rigid body contact force as an implicit time-stepping, velocity-based LCP to guarantee non-penetration, directional friction, and approximated Coulomb friction cone conditions [Stewart and Trinkle 1996]. After performing collision handling in the rigid body simulation, we store a list of contact points for each rigid body.

For each rigid body that the cloth vertex \mathbf{x}_i collides with, we find the nearest contact point \mathbf{p} in the list of contacts for that rigid body and assign the normal contact force at \mathbf{p} to \mathbf{x}_i . This normal contact force will be divided by the number of cloth vertices associated with \mathbf{p} , after all the cloth vertices have been visited. We store all the normal contact forces applied on \mathbf{x}_i in a vector $\mathbf{f}_N \in \mathbf{R}^{3M}$, where M is the number of rigid bodies that \mathbf{x}_i is in contact with. For those cloth vertices that are not in contact with any rigid body, we simply obtain the normal contact forces from the repulsive forces computed in the cloth simulator.

4.2 Friction Force

For every cloth vertex \mathbf{x}_i in \mathcal{V}_c or \mathcal{V}_r , our algorithm formulates a nonlinear complementarity problem to compute the friction force on \mathbf{x}_i . Our formulation simultaneously considers all the objects in contact with each cloth vertex, leading to more accurate friction forces compared to the sequential computation in most cloth simulators.

Suppose a cloth vertex is in contact with M triangles, we denote the normal contact forces and tangential contact forces as \mathbf{f}_N and \mathbf{f}_T , where $\mathbf{f}_N, \mathbf{f}_T \in \mathbf{R}^{3M}$. \mathbf{f}_N is computed either from the rigid body simulation or from the repulsive force in the cloth simulator, but \mathbf{f}_T is an unknown vector. We formulate the following equations based on Coulomb’s friction model:

$$(\mu_i \|\mathbf{f}_N\| - \|\mathbf{f}_T\|)(\|\mathbf{v}_i^r + h \frac{\sum_{j=1}^M \mathbf{f}_{T_j}}{m}\|) = 0 \quad (1)$$

$$\mu_i \|\mathbf{f}_N\| - \|\mathbf{f}_T\| \geq 0 \quad (2)$$

where $1 \leq i \leq M$. μ is the friction coefficient, \mathbf{v}^r is the relative tangential velocity at the contact, h is the simulation time step, and

m is the mass of a cloth vertex. The subscript i or j indicates the contact with the i -th (or j -th) triangle. The first constraint describes a complementarity condition; either the relative tangential velocity at the next time step is zero (static friction), or the contact force lies on the boundary of the friction cone (dynamic friction). The second constraint enforces the contact force to lie in the friction cone.

Solving a nonlinear complementarity problem is very difficult and time consuming for a large number M . Fortunately, for cloth simulation, vertices collide with no more than two triangles, one on each side of the cloth. With $M = 2$, we can solve the problem efficiently by checking all possible cases. First, we assume that both contacts are static, leading to the following equations:

$$\|\mathbf{v}_1^r + h \frac{\mathbf{f}_{T_1} + \mathbf{f}_{T_2}}{m}\| = 0 \quad (3)$$

$$\|\mathbf{v}_2^r + h \frac{\mathbf{f}_{T_1} + \mathbf{f}_{T_2}}{m}\| = 0 \quad (4)$$

$$\mu_1 \|\mathbf{f}_{N_1}\| - \|\mathbf{f}_{T_1}\| \geq 0 \quad (5)$$

$$\mu_2 \|\mathbf{f}_{N_2}\| - \|\mathbf{f}_{T_2}\| \geq 0 \quad (6)$$

If solutions for \mathbf{f}_{T_1} and \mathbf{f}_{T_2} do not exist, we switch to the case where one contact is static and the other one is dynamic. We first choose the contact with larger $\mu_i \|\mathbf{f}_{N_i}\| - \frac{m \|\mathbf{v}_i^r\|}{h}$ to be dynamic. This heuristic is based on the static condition and the approximation of $\|\mathbf{f}_{T_i}\|$:

$$\mu_i \|\mathbf{f}_{N_i}\| - \|\mathbf{f}_{T_i}\| \approx \mu_i \|\mathbf{f}_{N_i}\| - \frac{m \|\mathbf{v}_i^r\|}{h} \quad (7)$$

Suppose the first contact is set to be dynamic. We replace Equation 3 with $\mu_1 \|\mathbf{f}_{N_1}\| - \|\mathbf{f}_{T_1}\| = 0$ and delete Equation 5. If we cannot find a solution for either contact being static, we relax both contacts and make them dynamic.

Once the friction force \mathbf{f}_T is solved, we can update the velocity of the vertex i by:

$$\bar{\mathbf{v}}_i^{n+1} = \bar{\mathbf{v}}_i^{n+1} + h \frac{\sum_{j=1}^M \mathbf{f}_{T_j}}{m} \quad (8)$$

Note that the friction forces are computed implicitly considering the velocity of the vertex at the next time step. Directly using $\bar{\mathbf{v}}_i^{n+1}$ (i.e. the velocity without the effect of collision) to compute friction force leads to unstable simulation in our experiments.

5 Rigid Cloth Patch

Our algorithm approximates the state of cloth using a collection of thin rigid bodies, which we call rigid cloth patches. Each rigid cloth patch is a local representation of a small piece of cloth currently in contact with the rigid bodies. At each time step, we dynamically create rigid cloth patches to best fit the geometry of the cloth and assign them appropriate velocities. Once initialized, the rigid cloth patches participate in the rigid body simulation in the same way as other rigid bodies in the scene.

We group all the cloth vertices in \mathcal{V}_r into rigid cloth patch clusters. A vertex i and a vertex j belong to the same cluster if they satisfy the following conditions: their distance is within a threshold in the world space (Equation 9), their normals are in the similar direction (Equation 10), and the displacement in the normal direction is sufficiently small (Equation 11).

$$\|\mathbf{x}_i - \mathbf{x}_j\| \leq \delta_w \quad (9)$$

$$\|\mathbf{n}_i - \mathbf{n}_j\| \leq \delta_n \quad (10)$$

$$\|(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{n}_i\| + \|(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{n}_j\| \leq \delta_d \quad (11)$$

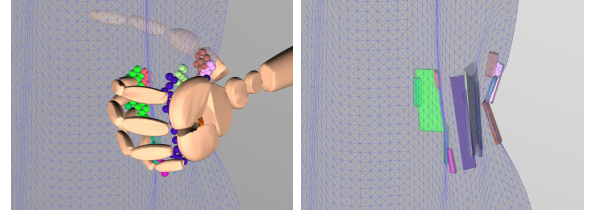


Figure 3: Illustration of rigid cloth patches. Left: Clustered vertices in the cloth triangle mesh. Right: Rigid cloth patches built from the clustered vertices.

We set the threshold from Equation 9 to Equation 11 to be $40mm$, 20° , and $3mm$ respectively.

For each cluster, we determine the orientation of the corresponding rigid cloth patch by fitting a plane to all the vertices in the cluster. The area of the rigid cloth patch is the minimum rectangle that covers all the vertices projected on the plane. The width of the rigid cloth patch is determined by the thickness of the cloth used in cloth simulator. In addition to geometry properties, we also need to assign the mass and the velocity to each rigid cloth patch. The mass is determined by the total mass of the cloth vertices in the cluster. For the velocity, we approximate the linear velocity of the rigid cloth patch using the average velocity of its corresponding cloth vertices. The angular velocity of the rigid cloth patch is approximated based on the total angular momentum of the corresponding cloth vertices with respect to the center of mass of the rigid cloth patch.

6 Hand Control

We designed a few simple hand controllers to perform grasping, pinching, and pressing the cloth with one or both hands. The controllers generate appropriate torques, τ_{int} which are added to the equations of motion by the rigid body simulator:

$$\mathbf{M}(\mathbf{q}_h) \ddot{\mathbf{q}}_h + \mathbf{C}(\mathbf{q}_h, \dot{\mathbf{q}}_h) = \tau_{ext} + \tau_{int} \quad (12)$$

where \mathbf{q}_h , a subset of \mathbf{q} , are the degrees of freedom of the hand model. \mathbf{M} and \mathbf{C} are the mass matrix and the Coriolis and centrifugal forces expressed in generalized coordinates. τ_{ext} are the sum of all external forces, including the collision forces with rigid cloth patches.

All the controllers are developed based on proportional-derivative (PD) actuators at joints to track a desired hand pose created either manually or by inverse kinematics. For some manipulations, such as pressing a cloth on the table, we use Jacobian transpose control scheme to compute τ_{int} such that the desired Cartesian force can be generated at the end-effector:

$$\tau_{int} = \mathbf{J}_p^T \mathbf{F} \quad (13)$$

Here \mathbf{J}_p is the Jacobian matrix at a point p on the hand and \mathbf{F} is the desired Cartesian force.

7 Results

We applied our method to a variety of cloth manipulation scenarios shown in Figure 1 and in the accompanying video. Our hand model is designed based on an anthropomorphic hand structure with 33 degrees of freedom. The breakdown of computation time and the number of triangles in the input mesh is shown in Table 1. On average, the computation due to our coupling interface is about 1% of the total computation time. The simulation time step is 5 ms. We

set the repulsion thickness in ARCSim to 5mm and the damping coefficient to 0.005.

	ARCSim	DART	Interface	Time	#Triangle
Dressing	85 %	12 %	3 %	8.52s	8441
Curtain	97 %	2 %	1 %	5.30s	16384
Moving	98 %	1.7 %	0.3%	5.33s	8194

Table 1: Computational performance breakdown: ARCSim (cloth), DART (rigid), and the Interface (our method). Time: the average of total simulation time (in second) per simulation time step. #Triangle: the number of triangles in the input mesh.

Pinching, moving, and folding cloth. We used the thumb and the index finger to pinch the cloth firmly and move it around. The fingers were controlled by using PD controllers to track a pose. External forces and torques were applied at the wrist to translate and rotate the hand. We also used two hands to grab the corners of a piece of cloth and fold it in half. These examples demonstrate that our method is able to generate appropriate friction to grasp the cloth without slipping.

Putting on a sleeve. Dressing requires some of the most intricate dexterous manipulation skills. To demonstrate the possibility of applying our method to this interesting problem, we designed a simple controller that uses one hand to put a sleeve on the other arm. The hand grabbed the sleeve with the thumb, index finger, and middle finger. This example involves a large number of contact points between the cloth and the finger, and between the cloth and the forearm.

Pulling a curtain. Pulling a shower curtain presents a scenario of interaction between cloth and both the palm and fingers. We controlled the entire hand to grab the curtain and slide it sideways. This example illustrates the importance of rigid cloth patch simulation because the collision state between the hand and the cloth is complicated and rapidly changing over time.

Removing and generating wrinkles. By commanding two hands to press against a piece of cloth on the table, we can generate and remove wrinkles in the cloth. We used Jacobian transpose controllers to generate desired forces from the bottom of the palm. In the first example, the cloth presents asymmetric wrinkles in its initial configuration. When the hands slide across the cloth, the wrinkles are removed. Similarly, we showed that the wrinkles could be generated by hand motion on a cloth that was initially smooth and flat.

Picking up cloth. We demonstrated two examples of picking up the cloth from different initial configurations. The first example attempts to pick up a t-shirt in an arbitrary initial configuration. We use the entire hand to create an envelope grasp when it reaches the cloth. In the second example, the hand picks up the cloth lying flat on the table. A common strategy to pick up cloth in this situation is to first use fingers to generate opposing friction forces to create a bunch, and then grab the bunch to lift the cloth. Our method successfully simulated this scenario due to our friction force computation.

7.1 Evaluations

Controlling contact forces. An important control strategy in dexterous manipulation is to control contact forces applied on the object. One benefit of our algorithm is that the simulated cloth reacts differently when the hand applies different amounts of force to it. In Figure 4, we compared the simulation results with different push forces. When the finger exerts a small pressing force on the cloth, the finger can smoothly slide on the cloth. When the force is large,

wrinkles are generated due to the large friction. However, when the same scenarios are simulated using ARCSim, shown in the bottom row, the cloth motion shows no difference between two forces. This is because the contact force is computed based only on the relative position and velocity between the hand and the cloth. In this case, no matter how large the contact force between the hand and the cloth is, the relative position and velocity remain the same.

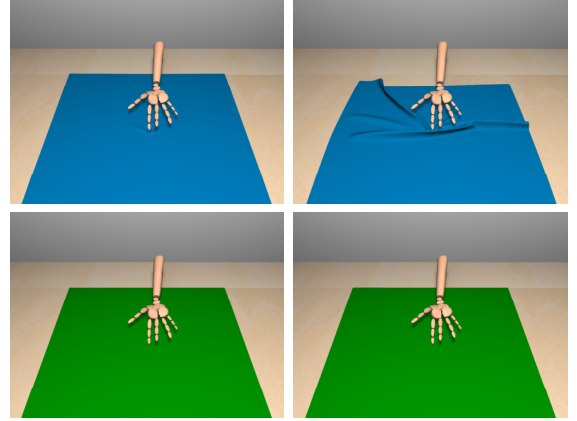


Figure 4: Controlling contact forces. Top row: Simulated results with our contact force computation in the situations of small (left) and large (right) pushing forces. Bottom row: Simulated results with ARCSim contact force computation in the situations of small (left) and large (right) pushing forces.

Manipulating multiple layers of cloth. Many common manipulation scenarios involve grasping multiple layers of cloth. Figure 5 demonstrates a scenario with three pieces of cloth held together by the hand. ARCSim typically has a difficult time handling such a scenario. When the fingers are far apart, the cloth cannot be held, and when the fingers are too close, the complex collision issues cannot be resolved. We showed that, due to the accurate approximation of cloth using rigid cloth patch simulation, our algorithm was able to hold three pieces of cloth together without any collision issues.



Figure 5: Collision with multiple cloth. Left: Simulated results of holding three pieces of cloth with (left) and without (right) rigid cloth patches.

7.2 Limitations

Although we made the decision not to implement accurate two-way coupling for the reason of having a simple implementation and efficient computation, we recognized that some examples of dexterous manipulation require that the rigid bodies react to the dynamics of the cloth accurately. In particular, when the manipulator uses the cloth in hand to interact with other rigid objects, the forces from the cloth to the rigid objects might not be negligible. For example, putting on a sock probably does not require two-way coupling between the hands and the sock, but the forces exchange between the

sock and the foot cannot be ignored.

In the cloth simulator, the cloth refinement steps, i.e. remeshing, inelastic projection and strain limit correction sometimes interfere with the contact force computed by our algorithm. The issue is more evident for static contact points. Without these refinement steps, the contact forces computed by our algorithm were able to enforce a static contact. However, when we activated the refinement procedures, the static contacts could be moved to different locations at the end of the simulation step. Therefore, we used position constraints only during the refinement steps to ensure that these vertices are not altered as a side effect of the refinement steps. Note that the position constraints do not add any forces or change the dynamic state of the system. If we skip the refinement steps, the position constraints will have no effect on the algorithm.

There are some limitations of using rigid cloth patches. First, the flexibility of cloth can be limited by the rigid cloth patch approximation. In our implementation, however, the rigid cloth patches are reconstructed at every time step and can be as small as the triangles on the cloth. By using small-scale rigid cloth patches and frequent reconstruction, we found the rigid cloth patch approximation to be sufficient for the hand manipulation tasks we demonstrated in this work. Second, the rigid cloth patches are not sufficiently accurate to provide the friction forces between the rigid body and the cloth. This was the reason why we only used rigid cloth patches to approximate the geometric state of the cloth and computed friction forces by solving the nonlinear complementarity problem.

8 Conclusion

We introduce a simulation technique that couples cloth and rigid body simulation to synthesize dexterous manipulation of cloth. Our technique is built upon existing cloth and rigid body simulators. We focus on two main issues in current cloth simulators. First, the contact force computation does not take into account the dynamic information of the rigid body. Second, rigid body motion is unaware of the state of cloth, which frequently causes unsolvable collision situations. We develop a light-weight interface through which the rigid body and the cloth simulators communicate on a demand-driven manner to resolve these two issues. We demonstrate a set of basic manipulation skills, such as gripping, pinching, and pressing.

This work is intended to be a simple and practical solution to couple existing rigid body simulators and cloth simulators for dexterous manipulation with deformable objects. We believe that it is an important missing piece, which will enable many interesting future research problems in control and simulation. For example, developing controllers for tying shoelaces or folding laundry can benefit robotic manipulators in numerous ways. We also envision that this technique can be integrated with full-body character simulation and control. Developing a virtual character capable of dressing itself may also be a fruitful future research direction.

9 Acknowledgements

We thank the anonymous reviewers for their helpful comments. We want to thank all the members of Gatech Graphics Lab for their help on this work. This work was funded by NIH R01 NS069655-05 and NSF G&V 1064983.

References

AINSLEY, S., VOUGA, E., GRINSPUN, E., AND TAMSTORF, R. 2012. Speculative parallel asynchronous contact mechanics. *ACM Trans. Graph. (SIGGRAPH Asia 2012)* 31, 6, 151.

ARCSim, <http://graphics.berkeley.edu/resources/arcsim/>.

BAI, Y., AND LIU, C. 2014. Dexterous manipulation using both palm and fingers. In *Proc. 2014 IEEE International Conference on Robotics and Automation*, IEEE.

BARAFF, D., AND WITKIN, A. 1998. Large steps in cloth simulation. In *Proc. SIGGRAPH 98*, ACM, 43–54.

BARAFF, D., WITKIN, A., AND KASS, M. 2003. Untangling cloth. *ACM Trans. Graph. (SIGGRAPH 2003)* 22, 3, 862–870.

BERSCH, C., PITZER, B., AND KAMMEL, S. 2011. Bimanual robotic cloth manipulation for laundry folding. In *Proc. 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 1413–1419.

BRIDSON, R., FEDKIW, R., AND ANDERSON, J. 2002. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph. (SIGGRAPH 2002)* 21, 3, 594–603.

BRIDSON, R., MARINO, S., AND FEDKIW, R. 2003. Simulation of clothing with folds and wrinkles. In *Proc. 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 28–36.

Bullet Physics Library, <http://bulletphysics.org/wordpress/>.

CUSUMANO-TOWNER, M., SINGH, A., MILLER, S., O'BRIEN, J. F., AND ABBEEL, P. 2011. Bringing clothing into desired configurations with limited perception. In *Proc. 2011 IEEE International Conference on Robotics and Automation*, IEEE, 3893–3900.

DART: Dynamic Animation and Robotics Toolkit", <https://github.com/dartsim/dart/>.

FAHANTIDIS, N., PARASCHIDIS, K., PETRIDIS, V., DOULGERI, Z., PETROU, L., AND HASAPIS, G. 1997. Robot handling of flat textile materials. *Robotics & Automation Magazine, IEEE* 4, 1, 34–41.

GRINSPUN, E., HIRANI, A. N., DESBRUN, M., AND SCHRÖDER, P. 2003. Discrete shells. In *Proc. 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 62–67.

HARMON, D., VOUGA, E., TAMSTORF, R., AND GRINSPUN, E. 2008. Robust treatment of simultaneous collisions. *ACM Trans. Graph. (SIGGRAPH 2008)* 27, 3, 23.

HARMON, D., VOUGA, E., SMITH, B., TAMSTORF, R., AND GRINSPUN, E. 2009. Asynchronous contact mechanics. In *ACM Trans. Graph. (SIGGRAPH 2009)*, vol. 28, ACM, 87.

JAMES, D. L., AND PAI, D. K. 2001. A unified treatment of elastostatic contact simulation for real time haptics. *Haptics-e, The Electronic Journal of Haptics Research* 2, 1.

JANSSON, J., AND VERGEEST, J. S. 2003. Combining deformable-and rigid-body mechanics simulation. *The Visual Computer* 19, 5, 280–290.

KALDOR, J. M., JAMES, D. L., AND MARSCHNER, S. 2010. Efficient yarn-based cloth with adaptive contact linearization. *ACM Trans. Graph. (SIGGRAPH 2010)* 29, 4, 105.

KAUFMAN, D. M., SUEDA, S., JAMES, D. L., AND PAI, D. K. 2008. Staggered projections for frictional contact in multi-body systems. In *ACM Trans. Graph. (SIGGRAPH Asia 2008)*, vol. 27, ACM, 164.

- KOSUGE, K., YOSHIDA, H., FUKUDA, T., SAKAI, M., AND KANITANI, K. 1995. Manipulation of a flexible object by dual manipulators. In *Proc. 1995 IEEE International Conference on Robotics and Automation*, vol. 1, IEEE, 318–323.
- KRY, P. G., AND PAI, D. K. 2006. Interaction capture and synthesis. *ACM Trans. Graph. (SIGGRAPH 2006)* 25, 3, 872–880.
- LIU, T., BARGTEIL, A. W., O'BRIEN, J. F., AND KAVAN, L. 2013. Fast simulation of mass-spring systems. *ACM Trans. Graph. (SIGGRAPH Asia 2013)* 32, 6, 214.
- LIU, C. K. 2009. Dextrous manipulation from a grasping pose. *ACM Trans. Graph. (SIGGRAPH 2009)* 28, 3.
- MAGENAT-THALMANN, N., VOLINO, P., BONANNI, U., SUMMERS, I. R., BERGAMASCO, M., SALSEDO, F., AND WOLTER, F.-E. 2007. From physics-based simulation to the touching of textiles: The haptex project. *IJVR* 6, 3, 35–44.
- Maya, <http://www.autodesk.com/products/autodesk-maya/overview>.
- MIGUEL, E., AND OTADUY, M. A. 2011. Efficient simulation of contact between rigid and deformable objects. *Multibody Dynamics*.
- MILLER, S., VAN DEN BERG, J., FRITZ, M., DARRELL, T., GOLDBERG, K., AND ABBEEL, P. 2012. A geometric approach to robotic laundry folding. *The International Journal of Robotics Research* 31, 2, 249–267.
- MÜLLER, M., AND GROSS, M. 2004. Interactive virtual materials. In *Proc. Graphics Interface 2004*, Canadian Human-Computer Communications Society, 239–246.
- MÜLLER, M., HEIDELBERGER, B., HENNIX, M., AND RATCLIFF, J. 2007. Position based dynamics. *Journal of Visual Communication and Image Representation* 18, 2, 109–118.
- NARAIN, R., SAMII, A., AND O'BRIEN, J. F. 2012. Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. Graph. (SIGGRAPH Asia 2012)* 31, 6, 152.
- NARAIN, R., PFAFF, T., AND O'BRIEN, J. F. 2013. Folding and crumpling adaptive sheets. *ACM Trans. Graph. (SIGGRAPH 2013)* 32, 4, 51.
- Open Dynamics Engine, <http://www.ode.org>.
- OSAWA, F., SEKI, H., AND KAMIYA, Y. 2007. Unfolding of massive laundry and classification types by dual manipulator. *JACIII* 11, 5, 457–463.
- OTADUY, M. A., TAMSTORF, R., STEINEMANN, D., AND GROSS, M. 2009. Implicit contact handling for deformable objects. *Computer Graphics Forum (Eurographics 2009)* 28, 2, 559–568.
- POLLARD, N. S., AND ZORDAN, V. B. 2005. Physically based grasping control from example. In *Proc. 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, 311–318.
- SHINAR, T., SCHROEDER, C., AND FEDKIW, R. 2008. Two-way coupling of rigid and deformable bodies. In *Proc. 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 95–103.
- SIFAKIS, E., SHINAR, T., IRVING, G., AND FEDKIW, R. 2007. Hybrid simulation of deformable solids. In *Proc. 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 81–90.
- STEWART, D. E., AND TRINKLE, J. C. 1996. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal for Numerical Methods in Engineering* 39, 15, 2673–2691.
- TANG, M., MANOCHA, D., AND TONG, R. 2010. Fast continuous collision detection using deforming non-penetration filters. In *Proc. 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, ACM, 7–13.
- TERZOPOULOS, D., PLATT, J., BARR, A., AND FLEISCHER, K. 1987. Elastically deformable models. In *Proc. SIGGRAPH 87*, vol. 21, ACM, 205–214.
- WANG, H., RAMAMOORTHY, R., AND O'BRIEN, J. F. 2011. Data-driven elastic models for cloth: Modeling and measurement. *ACM Trans. Graph. (SIGGRAPH 2011)* 30, 4 (July), 71:1–11.
- WANG, Y., MIN, J., ZHANG, J., LIU, Y., XU, F., DAI, Q., AND CHAI, J. 2013. Video-based hand manipulation capture through composite motion control. *ACM Transactions on Graphics (TOG)* 32, 4, 43.
- WU, J., LUO, Z., YAMAKITA, M., AND ITO, K. 1995. Adaptive hybrid control of manipulators on uncertain flexible objects. *Advanced Robotics* 10, 5, 469–485.
- YE, Y., AND LIU, C. K. 2012. Synthesis of detailed hand manipulations using contact sampling. *ACM Trans. Graph. (SIGGRAPH 2012)* 31, 4, 1–10.
- ZHAO, W., ZHANG, J., MIN, J., AND CHAI, J. 2013. Robust realtime physics-based motion control for human grasping. *ACM Trans. Graph. (SIGGRAPH 2013)* 32, 6, 207.