Animating Human Dressing

Alexander Clegg * Jie Tan † Greg Turk ‡ C. Karen Liu §

Georgia Institute of Technology

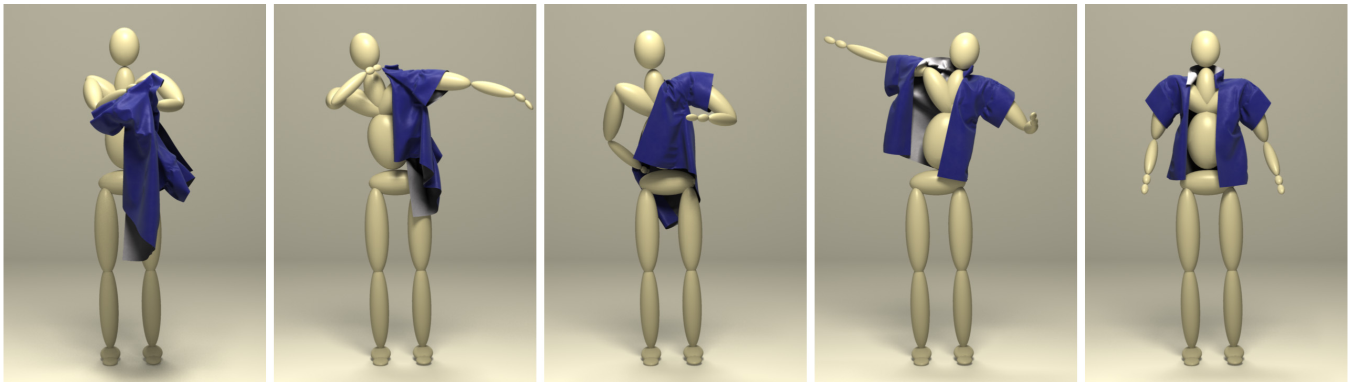

Figure 1: *A character puts on a jacket.*

Abstract

Dressing is one of the most common activities in human society. Perfecting the skill of dressing can take an average child three to four years of daily practice. The challenge is primarily due to the combined difficulty of coordinating different body parts and manipulating soft and deformable objects (clothes). We present a technique to synthesize human dressing by controlling a human character to put on an article of simulated clothing. We identify a set of *primitive actions* which account for the vast majority of motions observed in human dressing. These primitive actions can be assembled into a variety of motion sequences for dressing different garments with different styles. Exploiting both feed-forward and feedback control mechanisms, we develop a dressing controller to handle each of the primitive actions. The controller plans a path to achieve the action goal while making constant adjustments locally based on the current state of the simulated cloth when necessary. We demonstrate that our framework is versatile and able to animate dressing with different clothing types including a jacket, a pair of shorts, a robe, and a vest. Our controller is also robust to different cloth mesh resolutions which can cause the cloth simulator to generate significantly different cloth motions. In addition, we show that the same controller can be extended to assistive dressing.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation.

Keywords: Human figure animation, cloth simulation, path planning.

1 Introduction

This paper describes a system for animating the activity of putting on clothing. Dressing is one of the most common activities that each of us carries out each day. Scenes of dressing are also common in live-action movies and television. Some of these scenes are iconic, such as the "jacket on, jacket off" drill in The Karate Kid (2010 version) or Spiderman pulling his mask over his head for the first time. Such dressing scenes are noticeably absent in computer animated films. Despite the importance of dressing in our lives and in film, there is as yet no systematic approach to animating a human that is putting on clothing.

Our goal is to provide a system that will allow an animator to create motion for a human character that is dressing. We want the animator to have a high degree of control over the look of the final animation. To this end, we desire a system that allows the user to describe the dressing scene as a sequence of high-level actions. Also, we would like our system to accept approximated human motion, in either the form of keyframes or motion capture, as reference for styles or aesthetics. Thus the input from the animator for a given dressing scene consists of: a character model, a garment, a sequence of dressing actions, and reference motions for the character. In order to create animation that is physically plausible, we made the choice to use physical simulation of cloth to guide the garment motions. By using cloth simulation, the human figure, made of a collection of rigid segments, can interact with the cloth in a natural manner.

The essence of animating the act of dressing is modeling the interaction between the human character and the cloth. The human's motion must adapt to the motion of the cloth, otherwise problems occur such as the clothing slipping off or a hand getting stuck in a fold. We often take for granted the complex set of motions that are needed to put on our clothes. The seemingly simple act of putting on a jacket requires a careful coordination between the person and the jacket. Unconsciously we make constant adjustments to our hand's position when inserting it into the jacket's sleeve. We hold our body at an angle to keep a sleeve from sliding off our shoulder. After putting on the first sleeve, we may use any of several strategies to get our hand behind our back and within reach of the second sleeve. A system for animation of dressing must address

---

[*] e-mail: aclegg3@gatech.edu

[†] e-mail: jtan34@gatech.edu

[‡] e-mail: turk@cc.gatech.edu

[§] e-mail: karenliu@cc.gatech.edu

these kinds of complexities.

We have found that a small set of *primitive actions* account for the vast majority of the motions that a person goes through to put on an article of clothing. The approach that we take to dressing is to first have the animator assemble a desired dressing motion from a small number of such actions. These actions include placing a hand or foot through an opening, pulling the garment onto a limb, and stretching out a limb after it has been positioned in the clothing. Once this sequence of actions has been assembled, producing the dressing animation can proceed. The system steps forward in time, updating the cloth's position through simulation. The character's motion during each of the actions are guided by optimization and planning in order to satisfy the requirements of a given action. The system adjusts the character's pose to match the end of one action to the start of the next. Some portions of a dressing sequence do not require the character to react to the cloth, and such segments can follow the provided keyframe or motion capture data.

To a large degree, the problem that a dressing animation system must solve is a form of *path planning*. The character's body parts must move in coordination to complete the task while preventing self-intersection and the character must move in and around the cloth in such a way that the garment ends up properly on the person. However, the dressing problem has a few unique challenges which are not addressed by standard path planning algorithms. Unlike typical path planning that avoids collisions, contact between the body parts and the cloth is to be expected. In fact, *utilizing* contact to expand the opening of a folded sleeve or fixate a part of cloth on the body is crucial for successful dressing.

Using our action-based dressing system, we have produced a variety of animations of a character that is putting on various types of clothes. This includes putting on a jacket, pulling on pants while sitting, putting on pants while standing, dynamically swinging on a vest, and having one character assist another in putting on a robe.

2 Related Work

Close range interaction with surrounding objects or humans is an important research problem in character animation. This problem is challenging because it often involves potentially conflicting goals: maintaining intentional spatial constraints while avoiding unintentional contacts. Much research has been done on the challenge of handling contact and spatial constraints between body parts or objects [Gleicher 1998; Liu et al. 2006; Ho and Komura 2009; Kim et al. 2009; Ho et al. 2010]. Ho *et al.* [2010] used an "interaction mesh" to encode the spatial relationship of interacting body parts. By minimizing the local deformation of the mesh, their method preserved the desired spatial constraints while reducing unintentional contacts or interpenetrations. In additional to the contact problem, close range interaction also demands sophisticated path planning. Previous work exploited inverse kinematics and motion planning techniques to generate motion that satisfies desired manipulation tasks in complex or cluttered environments [Kallmann et al. 2003; Yamane et al. 2004]. A large body of robotics literature on the topic of motion planning for full-body manipulation is also highly relevant to the synthesis of close range interaction [Harada et al. 2003; Takubo et al. 2005; Yoshida et al. 2005; Nishiwaki et al. 2006]. In this paper, dressing is also an example of close range interaction. Unlike most problems studied previously, dressing involves interacting with a unique object, cloth, which is highly deformable with frequent self-collisions.

Researchers studying dexterous manipulation have developed control algorithms to handle different types of manipulation, such as grasping [Pollard and Zordan 2005; Kry and Pai 2006; Wang et al. 2013b; Zhao et al. 2013], finger gaiting [Ye and Liu 2012], or rolling [Bai and Liu 2014b]. These methods can successfully manipulate rigid bodies with various sizes and masses, but it is not clear whether they can be extended to manipulating deformable bodies, which typically have more degrees of freedom than rigid bodies [Wang and Komura 2012]. In contrast to computer graphics, manipulating deformable bodies has been addressed extensively in robotics. Researchers have demonstrated robots manipulating cloth, ropes, cables, foam rubber, and sheet metal [Kosuge et al. 1995; Wu et al. 1995; Fahantidis et al. 1997; Osawa et al. 2007; Cusumano-Towner et al. 2011; Bersch et al. 2011; Miller et al. 2012]. Our work is related to manipulation of cloth for folding laundry [Osawa et al. 2007; Cusumano-Towner et al. 2011; Bersch et al. 2011; Miller et al. 2012] and assisted dressing of partially dressed, static mannequins [Tamei et al. 2011]. However, due to the involvement of the human body, developing control algorithms for dressing differs substantially from robotic manipulation of cloth alone. In this paper, we do not address the problems related to grasping and re-grasping, since this constitutes distinct challenges and is actively being addressed by others in the robotics community.

A self-dressing virtual character has been previously demonstrated by a few methods. Ho and Komura [2009] introduced a technique to interact with deformable bodies using topology coordinates, in which the topological relationship of the character's body and the environment can be easily controlled. They generated keyframe animation in topology coordinates to demonstrate that a character is able to stretch her arms out of a piece of clothing that is wrapped around her. To demonstrate the effect of using electric flux for path planning, Wang *et al.* [2013a] showed a virtual human putting on a sock and a pair of shorts. In both cases, the clothes are already aligned with the body parts and the character simply needs to pull them in the direction indicated by the electric flux. While these methods hint at possible solutions to the dressing problem based on movement of the character relative to the cloth, they have only been successfully applied to isolated situations and under many assumptions. In contrast, our work designs a feedback controller such that the character can act autonomously based on the state of the cloth and effectively achieve the goal in a diverse set of dressing situations with relatively few assumptions.

Although cloth simulation is a relatively mature research area, dynamic coupling between cloth and rigid body systems still presents many challenges. A variety of methods are proposed to handle two-way coupling between deformable and rigid bodies [Jansson and Vergeest 2003; Sifakis et al. 2007; Shinar et al. 2008; Otaduy et al. 2009; Miguel and Otaduy 2011; Macklin et al. 2014], which can be potentially extended to rigid-cloth coupling. Otaduy et. al. [Otaduy et al. 2009] solved contacts between cloth and rigid bodies by implicitly solving a large mixed linear complementarity problem. Bai and Liu [Bai and Liu 2014a] proposed a simpler coupling method which treats existing cloth and rigid body simulators as black boxes without altering the internal formulation of collision handling. Most recently, Macklin et. al. proposed a unified dynamic system in which all object interactions are solved as a collection of particle constraints. In our work, we directly use the open source multibody simulator, DART [Liu and Jain 2012], and the cloth simulator, ARCSim [Narain et al. 2012; Narain et al. 2013]. ARCSim treats the rigid bodies in the scene as objects with infinite mass and only considers the contact forces from the rigid bodies to the cloth. Since the type of clothes we consider in this paper are relatively massless compared to the human character, ignoring the impact of cloth on the character is a reasonable assumption. However, considering accurate two-way coupling might be critical for dressing tighter clothing or assistive dressing.

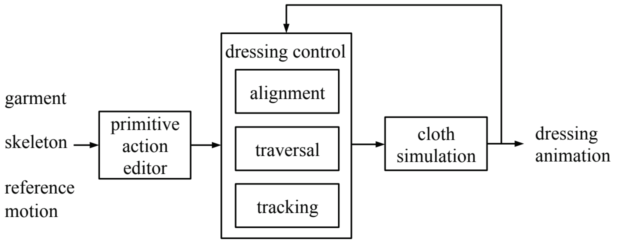

Figure 2: *The overview of our system.*

Action	Description
Grip(RH, $\mathbf{f}_1$)	Grip the collar feature $\mathbf{f}_1$ with the right hand.
Track($\hat{\mathbf{q}}(t)$, T_1)	Track the reference motion $\hat{\mathbf{q}}$ for T_1 seconds.
Align(LH, $\mathbf{f}_2$)	Align the left hand with the armhole $\mathbf{f}_2$.
Drag(RH, $\{B_i\}$)	Drag the cloth along the left hand B_1, the left arm B_2 and the left shoulder B_3.
Release(RH)	Release the cloth from the right hand.
Track($\hat{\mathbf{q}}$(t), T_2)	Track the reference motion $\hat{\mathbf{q}}$ for T_2 seconds.
Align(RH, $\mathbf{f}_3$)	Align the right hand with the right armhole $\mathbf{f}_3$.
Stretch(RH)	Stretching the right hand into the sleeve.
Track($\hat{\mathbf{q}}$(t), T_3)	Track the reference motion $\hat{\mathbf{q}}$ for T_3 seconds.
Idle(T_4)	Idle for T_4 seconds.

Table 1: *An example action queue for dressing the upper body of a character with a jacket.*

3 Overview

We have designed a system that allows a virtual human character to put on various types of garments. Our system consists of three main components: the primitive action editor, the dressing controller and the cloth simulator. The input to our system includes a garment, a character, and a reference dressing motion that approximates the desired dressing style. The reference motion can be a motion captured sequence or a sparse set of keyframes. The user first assembles a sequence of actions to describe the reference motion using our primitive action editor. For example, putting an arm into a sleeve can be described as first *aligning* the hand with the armhole and then *dragging* the cloth up the arm, from the wrist to the shoulder. These primitive actions are parameterized building blocks for creating various dressing animations. Table 1 shows a complete action queue for putting on a jacket. At each step, our system fetches an action from the queue, executes the corresponding dressing controller and simulates the physics of the cloth. Figure 2 illustrates the main components of our system.

4 Garments

Our example garments are from the Berkeley Garment Library, and have been edited to fit the size of our human character. These example garments are a jacket, a pair of shorts, a robe, and a vest. The garments are modeled as a finite element mesh and their motion is physically simulated using the ARCSim cloth simulator [Narain et al. 2012]. We use linear stretching and bending models and constitutive models derived from measurements [Wang et al. 2011]. The collisions are detected using a bounding volume hierarchy [Tang et al. 2010] and resolved with non-rigid impact zones [Harmon et al. 2008].

Garment Features. For each garment we define a set of cloth features that are important for dressing control. A feature is a set of vertices on the cloth mesh. Each feature is either a target for a hand or foot to align with, or a location for a hand to grasp. For example, we use the vertex loop of an armhole as a feature for the hand to target when putting the arm into a sleeve. Figure 3 shows all the features that we use for the jacket and the shorts.

Figure 3: *Cloth features of a jacket and a pair of shorts that are used in dressing control. The red loops and patches are the features for alignment and grip respectively.*

5 Dressing Control

The dressing problem, in its most abstract form, can be viewed as path planning with the goal of finding a path in the configuration space of the human such that the article of clothing ends up on the human's body in a desired manner. Unlike traditional path planning problems, the validity of the path depends on the evolution of another dynamic system, the cloth. The need to consider the state of the garment invalidates many planning algorithms that utilize model predictive control, because the computational cost of cloth simulation is simply too high to be involved in any optimization loop. We remedy the issue based on two core ideas. First, we find that the state of cloth is extremely crucial, but only for a few brief moments in the entire scene. We identify those "cloth-sensitive" actions and develop separate controllers for them. Second, for planning the cloth-sensitive moments, we exploit the geodesic information of cloth to make up for the lack of computation resources for online prediction.

We have designed a small set of primitive actions to control a character to put on a variety of garments using different styles. The two most important actions are *alignment* and *traversal*. We monitor the cloth state and solve an optimization at each time step only during the alignment phase. For other actions, we plan the entire path at the beginning of the action, and do not take the state of cloth into consideration.

5.1 Alignment

The first step to putting on a garment is to align one body part, such as an end effector, with a cloth feature. Examples of this are to align a hand with the armhole of a shirt, or to align a foot with the waistband of a pair of pants. In alignment, we typically choose a loop of vertices as the cloth feature and the goal is to control the end effector to pass through this loop. This can be challenging because the target cloth feature is often folded and occluded by other parts of the cloth. In such cases, the target feature is not directly visible or reachable from the current end effector location. This means that alignment is a process of chasing a moving feature that has nonlinear dynamics and complex deformations. It is difficult to predict the movement of the feature without simulating the entire cloth, but the computation cost of cloth simulation makes this

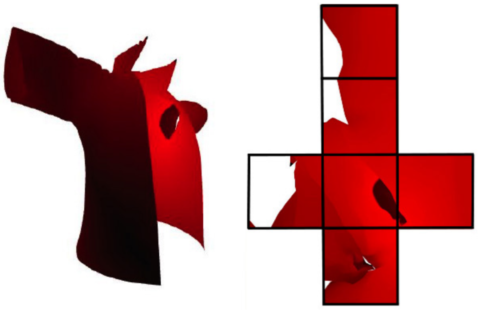

Figure 4: *Left: The precomputed geodesic distance on the jacket to the left armhole. Brighter red means smaller distance. Right: A cubic map rendering of the geodesic distance when the character aligns his hand with the first sleeve (The first column of Figure 1).*

approach infeasible. Worse yet, as the end effector approaches the feature, it is highly likely that it will collide with the cloth in the neighborhood of the feature and knock the target away. To address these challenges, we design a cloth-aware, feedback controller for the alignment action.

Our alignment controller first finds an intermediate goal towards the target feature, and then moves the end effector a small distance towards this goal in a way that minimizes the chance of knocking away the target feature. These two steps are performed iteratively until the end effector successfully reaches the feature.

We set the intermediate goal as a point on the cloth that is visible from the end effector and has the smallest geodesic distance to the target feature. In our implementation, we find this point using rasterization techniques (Figure 4 Right) and update it at every timestep. We assign the color of each vertex on the cloth mesh based on its geodesic distance to the target. We place a virtual camera at the end effector and render the cloth mesh into a cubic environment map. The brightest pixel on the map corresponds to the direction of the intermediate goal. Note that we choose to render all six directions of the cubic map to allow the end effector to move not only forward, but also sideways and backward. Our experiments show that the ability to detect the intermediate goal when it is behind the end effector drastically increases the success rate of the alignment. This is because a better intermediate goal might be behind the end effector and may only emerge when the end effector is sufficiently close to the current intermediate goal or when the end effector impacts the cloth, causing significant deformation.

In our initial work, the cloth geometry was represented as a single-layer triangular mesh, which means that there was no difference between a point at the outer and the inner surface of the cloth. We find, however, that it is important for the alignment to be able to distinguish between the inside and the outside of the garment. To achieve this, we duplicate the mesh to create two separate layers, and we connect these two layers at their boundaries. We precompute the geodesic distance on the two-layer cloth mesh using breadth first propagation starting from the feature vertices on the intended layer. For example, Figure 4 visualizes the geodesic distance from the left armhole on the inner side of jacket. Note that the vertices inside the sleeve have large geodesic distances because we restrict the direction of initial distance propagation toward the inside of the jacket by adding only the vertices on that side of the sleeve feature to the initial queue during Breadth First Search, essentially cutting graph connectivity at the inside seam. Otherwise, the best geodesic path to the feature could align the hand with the armhole by going through the cuff.

To move the end effector towards the intermediate goal, we formulate an optimization to solve a collision-free inverse kinematics (IK) problem. The solution moves the character's end effectors to the desired locations, keeps the full body motion similar to the reference, and guarantees that the body parts do not overlap with each other. We solve

$$\min_{\mathbf{q}} ||\mathbf{q}-\hat{\mathbf{q}}||^2_{\mathbf{w}} \tag{1}$$

$$\begin{aligned} &\text{subject to} \\ &\qquad \mathbf{p}(\mathbf{q}) = \hat{\mathbf{p}} \\ &\qquad \mathbf{q}_{min} \le \mathbf{q} \le \mathbf{q}_{max} \\ &\qquad ||\mathbf{c}_i(\mathbf{q}) - \mathbf{c}_j(\mathbf{q})||^2_2 - (r_i + r_j)^2 \ge 0 \end{aligned}$$

where $\mathbf{q}$ are the joint angles, $\hat{\mathbf{q}}$ is the reference pose, $\mathbf{w}$ is a diagonal matrix that specifies the weight of each joint, $\mathbf{p}(\mathbf{q})$ are the end effector positions, $\hat{\mathbf{p}}$ are the target positions, $\mathbf{q}_{min}$ and $\mathbf{q}_{max}$ are the joint limits. The last constraint prevents inter-body penetrations. We approximate the collision volume of each body with multiple spheres and enforce no penetration for each pair of spheres that belong to different bodies. $\mathbf{c}_i$ and r_i in the constraint are the center and radius of the ith sphere.

Given the direction to the goal $\mathbf{d}$ and the current end effector location $\mathbf{p}$, we set the desired end effector position $\hat{\mathbf{p}}$ at the next time step to be

$$\hat{\mathbf{p}} = \mathbf{p}^n + \alpha\mathbf{d} \tag{2}$$

where α is the user-specified step size. We choose the initial character pose when the alignment starts as the reference $\hat{\mathbf{q}}$ throughout the whole alignment phase. This reference pose has very little effect on the degrees of freedom active in the alignment task (e.g. wrist, elbows), but stabilizes the other degrees of freedom that are not involved in other constraints or objective terms in Equation 1 (e.g. knees in the jacket example).

In addition to the above IK formulation, we find that the orientation of the end effector also plays an important role in the alignment action. Since the end effector may need to weave through a tight and winding space between folds of the cloth, its orientation should be aligned with its direction of motion. This way of moving reduces the space swept by the end effector, lowering its chance of colliding with the nearby cloth and minimizing the normal impacts if collisions happen. We add the following objective to the optimization to regulate the orientation of the end effector.

$$E_{orientation} = 1 - \mathbf{d}^T\mathbf{r}(\mathbf{q}) \tag{3}$$

where $\mathbf{r}(\mathbf{q})$ is the direction from the center to the tip of the end effector.

We also limit the joint speed within a certain threshold to ensure the smoothness of the motion.

$$-\dot{\mathbf{q}}_{max} \le \frac{\mathbf{q}-\mathbf{q}^n}{\Delta t} \le \dot{\mathbf{q}}_{max} \tag{4}$$

where $\mathbf{q}^n$ is the current pose, Δt is the time step, and $\dot{\mathbf{q}}_{max}$ is the maximum allowed speed.

Finally, to determine whether the alignment action has succeeded, we first find a plane that best fits the cloth feature. We then project the feature vertices onto this plane. If the line segment linking the parent joint of the end effector to its center intersects the plane within the polygon formed by the projected feature, the alignment has succeeded and we move on to the next action in the queue.

5.2 Traversal

After alignment, the center of the end effector has passed the desired cloth feature. However, at this point, the feature can still easily fall out of control due to gravity or inappropriate end effector motions. We design a traversal action to further secure the alignment and bring the feature to its final destination. Examples of traversal include stretching an arm into a sleeve until the feature reaches the shoulder or dragging the waistband of the pants up to the waist. We find that unlike alignment, the feedback from the state of cloth does not play an important role during traversal. We therefore use a feed-forward controller that plans the joint trajectory for the entire action at the beginning of the traversal. We first compute a series of desired targets of the end effector relative to the other limb. We then solve the collision-free IK (Equation 1) for their corresponding full body poses, which are used as keyframes for the traversal motion. Although these keyframes are free of self collision among body parts, directly interpolating them can lead to inter-body penetrations. For this reason, we apply bi-directional Rapidly Expanding Random Tree (RRT) [LaValle and Kuffner 2001] to find a collision free trajectory between adjacent keyframes. RRT is a stochastic search method that finds collision-free paths in a given configuration space. In our case, the configuration is the set of joint angles for the relevant limbs. Because RRT takes random steps to explore the configuration space, the path it returns is typically jagged and indirect. As such, we shorten and smooth the resulting trajectory to remove any unsightly jittering.

We observed that in daily dressing activities, the traversal action can be categorized into two types. In the first type, the limb to be dressed remains relatively passive while another limb *drags* the cloth along it. For example, the character uses its hands to drag the pants up along the legs. In the second type, the limb *stretches* itself to pass through the tubular part of the cloth without assistance from other limbs. This situation is often seen when putting on the second sleeve of a jacket. To accommodate both types of traversal, we set up different objectives or constraints in the IK formulation.

Dragging. In the first case, a user can specify one end effector to drag the cloth and a set of body parts $\{B_1, ..., B_n\}$ that the cloth should be dragged upon. We want the end effector to pass just over the surface of the body parts and to avoid intersection. Therefore, we use an offset of the positions of the parent joints of those bodies as path nodes. The offset is based on the relative dimensions of the limbs being traced, such that the begin and end state of RRT will be collision free. For example, if the character is using his or her right hand to dress the left arm, the path nodes are points just off the surface of the left wrist, the left elbow and the left shoulder. For each path node $\mathbf{p}_i$, we set the target end effector location $\hat{\mathbf{p}} = \mathbf{p}_i$ in Equation 1, and solve the collision-free IK for one keyframe of the dragging motion.

Stretching. In the second case of traversal, one limb straightens into the cloth tube without assistance from other end effectors. The key to stretching a limb is to specify constraints that will move the limb's end effector away from the body. Just specifying a new end effector position is not sufficient because the whole body can lean to accomplish this. Instead, we need to guarantee that the limb's end effector moves relative to the rest of the body. We do this by identifying a part of the body (usually on the torso) that is our *anchor node*, and requiring that this node's position stays fixed. For example, when stretching the right arm into a sleeve as shown in the third column of Figure 1, the anchor node is the left shoulder. If the shoulder was not fixed, the IK could compute a pose in which the torso moves with the stretching arm. This would eliminate the required relative movement between the arm and sleeve for the stretching action. We implemented the anchor node as an additional constraint in the optimization (Equation 1).

$$\mathbf{p}(\mathbf{q}) = \mathbf{p}^n \tag{5}$$

where $\mathbf{p}^n$ and $\mathbf{p}(\mathbf{q})$ are the center of the anchor node at the current and the next time step.

Besides specifying the position of the anchor node, a correct stretching direction is also critical. We use the direction from the center of the anchor node to the current end effector location as the stretching direction. Along this direction, the friction force caused by the stretching limb is canceled by the tension force of the cloth pulling from the anchor node. This prevents the cloth feature from moving with the end effector so that the limb can further pass through the feature. We add an objective term to Equation 1 to specify the desired stretching direction.

$$E_{stretch} = \sum_i 1 - \mathbf{d}_{stretch}^T \mathbf{r_i}(\mathbf{q}) \tag{6}$$

where $\mathbf{d}_{stretch}$ is the desired stretching direction and $\mathbf{r_i}(\mathbf{q})$ is the longest principal axis of the ith body of the limb.

Together with the stretching direction objective (Equation 6) and the anchor node constraint (Equation 5), the collision-free IK (Equation 1) solves for the keyframe pose at the end of the stretching action.

5.3 Other Actions

Tracking. To preserve the dressing style specified by the user, we use the tracking action to follow the reference motion $\hat{\mathbf{q}}(t)$. In most cases, this action simply uses the next pose in the reference motion.

$$\mathbf{q} = \hat{\mathbf{q}}^{n+1}$$

However, after alignment and traversal, the joint trajectory of the character may deviate from the reference motion. For this reason, interpolation from the current pose to a pose in the reference is necessary for a smooth animation. To prevent inter-body collisions during the interpolation, similar to the traversal action, we apply RRT for a collision free path and then follow this path to the target pose.

Grip and Release. The grip action models the grasping of the character's hand. The cloth feature moves together with the character's hand if it is gripped. This action constrains the vertices in the cloth feature to the local frame of the end effector.

$$\mathbf{p}_w = \mathbf{R}\mathbf{p} + \mathbf{t}$$

where $\mathbf{p}_w$ is the world coordinate of a vertex in the cloth feature, $\mathbf{p}$ is the local coordinate of this vertex at the end effector's frame, $\mathbf{R}$ and $\mathbf{t}$ are the rotation and translation of the end effector. The release action simply removes the above constraints and the cloth feature no longer moves with the end effector once it is released.

Idling. This action freezes the character's motion for a user-specified time period. The main purpose of this action is to wait for the clothes to settle before proceeding to the next dressing action.

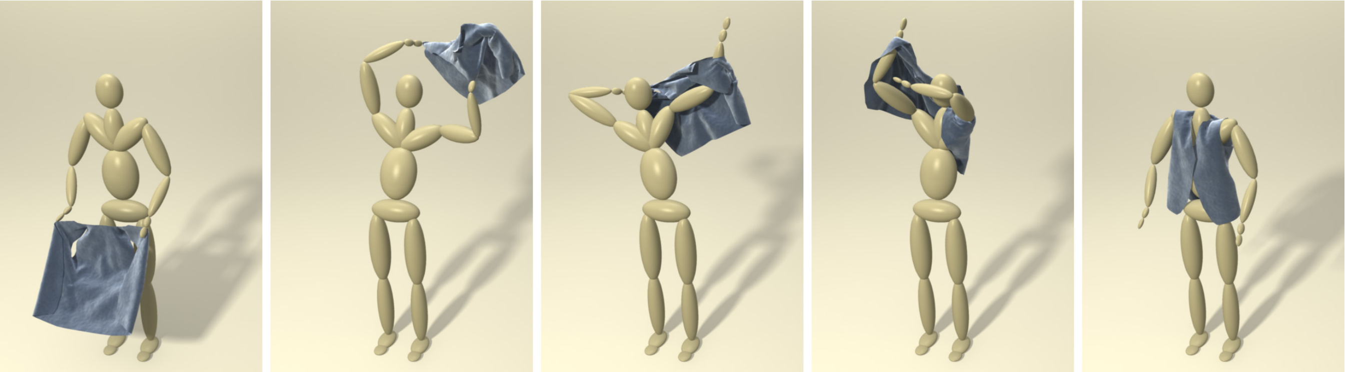

Figure 5: *A character puts on a vest by swinging it around his neck.*

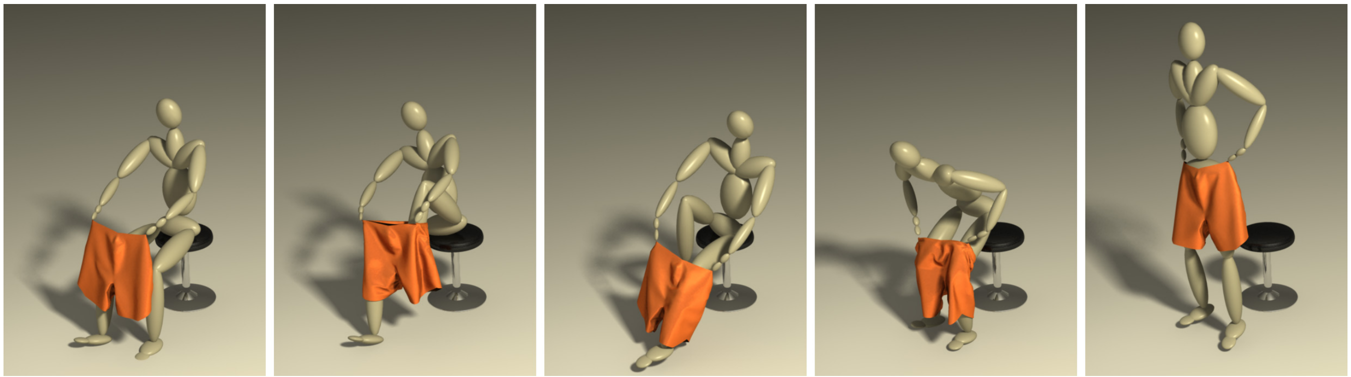

Figure 6: *A character puts on a pair of shorts in a sitting pose.*

6 Results

In this section we describe the results of our system. We chose four different types of garments from the Berkeley Garment Library, edited them to meet our needs, and put them on the character using different styles. The desired styles are specified by a sparse set of keyframes or motion capture data. All motion capture sequences used in the examples were acted out from memory, without real cloth garments. Please watch the accompanying video for the dressing animations. Our system was implemented in C++. We used DART [Liu and Jain 2012] for human character modeling and ARCSim [Narain et al. 2012] for cloth simulation. We solve the Inverse Kinematics optimization with SNOPT, which uses a sequential quadratic program solver. We used a denim material profile with 3-4x thickening to simulate layered clothing articles. The examples were run on a desktop workstation with a 4-core 3.7 GHz CPU and 8 GB of memory. The typical running time of a dressing example is several hours to a day. The cloth simulation is the most time-consuming part and takes significantly longer than control. The parameters and performance data of our examples are summarized in Table 2.

Jacket. Figure 1 shows a character putting on a jacket using a common style: Put the left arm in its sleeve, swing the right arm to the back, find the hanging sleeve, and stretch the arm into it. The reference human motion for this style is made up of six keyframes. As shown in the video, dressing by directly playing back the reference motion without any feedback control fails to put on even the first sleeve. After gripping the collar of the cloth, our system first tracks the reference motion to a particular keyframe and then performs an alignment action. The character aligns his left hand with the corresponding armhole. Once the alignment is successful, the traversal action is executed. The character uses his right hand to drag the cloth up the length of the left arm. At the end of traversal, the right hand reaches the shoulder and releases the cloth. The character then swings his right arm to the back by tracking the reference motion. The second alignment phase begins when the character's right hand starts to search for the opening of the sleeve. This alignment phase is more challenging because the target armhole is completely occluded by multiple layers of cloth. The alignment action gradually finds better intermediate goals that are closer to the target feature and guides the hand to wiggle through the cloth folds towards the goal.

In the jacket example we observed interesting emergent behavior, such as a natural exploratory gesture that is often used by humans to sort out the tangled cloth in dressing. Furthermore, the hand operates within a tight space between the pelvis and the hanging cloth during alignment. Without the collision-free IK, the hand or arm would penetrate the torso before aligning with the armhole. After the alignment, the character uses the traversal action to stretch the right arm into the sleeve and then tracks the reference motion until the jacket is completely on the body.

A common challenge in cloth simulation is that the simulation may produce drastically different motions if the resolution of the cloth changes. To test the robustness of our system, we used the same actions to put on a low resolution jacket with approximately 10x fewer triangles. Our system was able to automatically adapt to the change in resolution and the resulting differences in cloth motion without any manual tuning. Note that the more coarse resolution cloth has fewer wrinkles and folds. A consequence of this is that the lower resolution jacket has a less occluded armhole, so the right hand alignment phase is shorter in this sequence.

Vest. We simulated putting on a vest in a more dynamic style (Figure 5). We chose this example to demonstrate that using a small set of primitive actions, our system is able to put on a garment in a different style by using a different reference motion. Tracking the motion captured reference, the character swings the vest around his neck, then aligns the left hand with the corresponding armhole while the vest is swinging in mid-air. This alignment shows that our feedback control is robust and able to align not only with a target feature that is occluded by layers of cloth, but also with one that is moving quickly. Once the first arm is dressed, the right hand is aligned with the second armhole and stretches into it.

Shorts. Figure 6 demonstrates a character that is putting on a pair of shorts in a sitting position. We used a motion capture sequence as the reference motion in this example. The character grips the waistband and leans forward by tracking the reference motion. He first aligns his left foot with the waistband and then aligns it with the bottom of the shorts' left leg. Similar alignment actions are applied to the right foot. Once both feet are aligned with the desired features, the character follows the reference motion to stand up and pull up the shorts. The accompanying video shows that without the feedback control, the character fails to put the feet into the shorts and ends up not wearing the pants.

We also tested the generality of our system by using a different reference motion, in which we mocaped a person that is putting on a pair of shorts in a standing position. Despite the different style, we were able to reuse the action queue from the sitting shorts and successfully dress the lower body of the character (Figure 7).

Robe. To show that our system can be used outside of the realm of self-dressing, we applied our method to an assisted dressing scene in which one character aids another in putting on a robe (Figure 8). The reference motion for this scene consists of five keyframes. First, the dressing character tracks the reference motion, twisting to the left and aligning his left hand with the armhole. After he straightens his arm into the sleeve, the assistant releases the cloth from his left hand. Dressing the second arm is similarly performed with both characters tracking to a pre-alignment position whereupon the dressing character aligns and straightens his arm and the assistant releases the sleeve. Note that in this example, the dressing control is only performed on the dressing character while the motion of the assistant is pre-scripted. It would be interesting future work to simulate and control both characters in an assisted dressing task.

7 Limitations

Even though our system has produced a number of successful dressing animations, our current approach has some limitations. One such limitation is due to the nature of the feedback from the cloth position. In our system, this feedback is performed using visibility calculations. When the control system is guiding a hand to enter a sleeve, this is done by finding the closest point on the cloth to the sleeve entry that is visible to the tip of the hand. It is probable that when real people dress themselves, much of their knowledge about the cloth state is due to tactile feedback, instead of from visual information.

examples	cloth triangles	actions	anim time	sim time	control time
jacket	23.2k	10	18s	30h23m	14m
jacket (low res)	2.7k	10	18s	2h58m	1m
shorts (sitting)	14.9k	10	16s	8h41m	4m
shorts (standing)	14.9k	10	15s	7h52m	3m
vest	6.57k	14	13s	3h53m	1m
robe	31.7k	11	14s	20h40m	19m

Table 2: *Parameters and performance of the examples. cloth triangles: the number of elements in cloth simulation. Actions: number of actions. Anim time: wall clock time (in seconds) of the dressing animation. Sim and control times are the total times for the cloth simulation and our control functions respectively.*

Another limitation of our dressing controller is that it uses kinematic motion instead of calculating the dynamics of the human body. This has two repercussions. First, our system is one-way coupled, so that the cloth does not exert forces on the human. Certain character motions can occasionally cause the cloth to exceed its strain limit, producing unrealistic cloth behavior. This problem could be eliminated using feedback forces from the cloth. A second consequence is that the kinematically controlled human has no notion of balance, and thus may carry out physically impossible motions. This is especially important in dressing tasks such as putting on a pair of pants while standing up. The lack of balance control could also have more subtle effects on the stance of the person during upper body dressing.

A subset of human dressing motions involves long-term planning with approximate knowledge of the future state of the garment. This subset includes motions such as re-gripping, navigating the end effector around a moving garment and catching a thrown or dropped garment. Our current algorithm is shortsighted and therefore relies on user input to handle this class of dressing motions. It is reasonable to imagine incorporating faster, more approximate cloth simulation models to increase planning accuracy in these situations.

In all of our examples, the initial configuration of the cloth has been set in a way that is favorable to dressing. If the initial garment were tangled, our dressing actions would most likely fail. A possible avenue for addressing this may be found in the robotics research that has investigated picking up and folding cloth [Cusumano-Towner et al. 2011].

In our current implementation, actions are executed sequentially with the assumption that user input is sufficiently plausible for each action to be completed successfully. As such, our current system does not automatically respond to failure scenarios. Similarly, our system requires the user to specify the set of actions for a given dressing task. When given a new garment, the system has no way of determining a sequence of actions that will successfully place it on a body. We can imagine a more sophisticated system that would address both of these limitations by analyzing a garment, forming a plan of actions to properly dress a character and executing this plan with backtracking, re-planning, and other corrective procedures in place to respond to failures or adjust to environmental perturbation.

8 Conclusion

We have presented a system that allows an animator to create motions of people that are dressing. By providing reference motion and an action sequence, an animator has a fine degree of control over the method and the style that a character uses to put on a garment. We have demonstrated the use of our system in creating a variety of dressing animations, including putting on a jacket, a vest, pants while sitting, pants while standing, and assistance in putting on a robe. The key to our dressing system is path planning: visibility feedback for end effector alignment with cloth features, and limb motion planning to avoid body self-collisions.

There are several avenues for future work on animated dressing. One possibility is to incorporate dexterous manipulation of the cloth with our current system. Such an augmented system would allow a hand to properly grip a sleeve, instead of "gluing" the hand to a

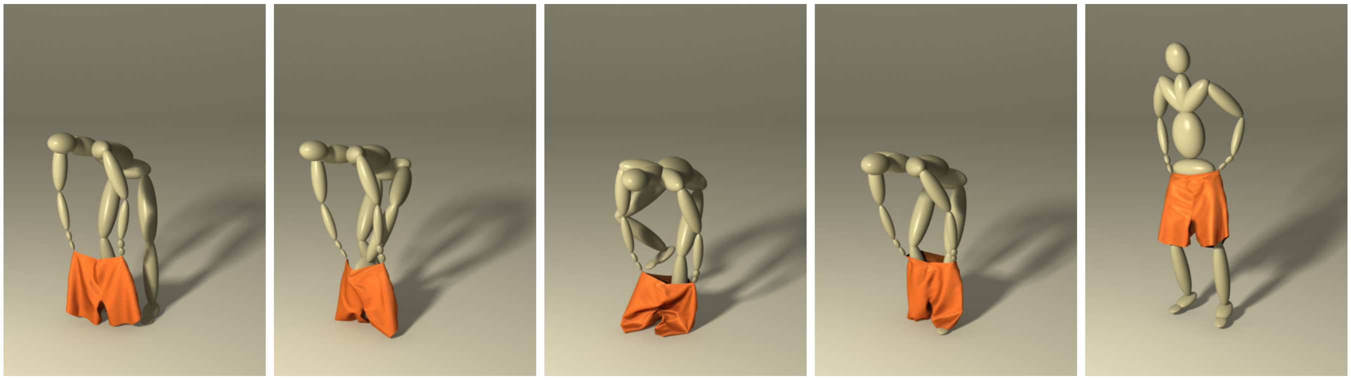

Figure 7: *A character puts on a pair of shorts in a standing pose.*

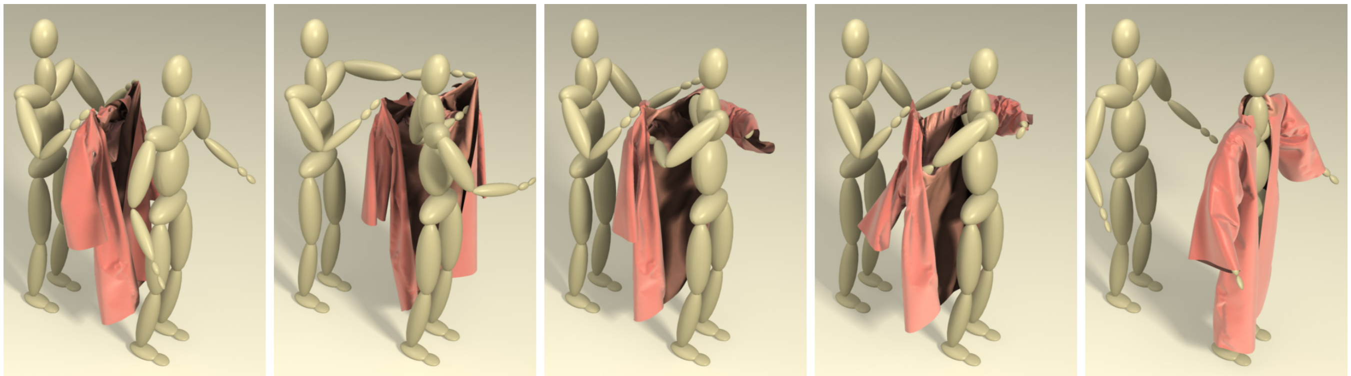

Figure 8: *A character puts on a robe with the help from another character.*

portion of the cloth as we currently do. Another important aspect of dexterous manipulation that we have not explored is the use of hands in fastening the garments, as is needed to use buttons, zippers and laces. As suggested in the Limitations section, we might want a system that can figure out a high level dressing strategy for a newly presented garment. Such a system would likely need to do some form of search across a variety of possible dressing strategies. There are potential improvements to cloth simulation that could lead to higher quality dressing animation. Current cloth simulators do not handle multiple layers of cloth well, such as putting on a jacket over a shirt. The ability to handle tight contact between cloth and the human figure would also increase the range of possible dressing simulations.

Acknowledgements

We want to thank the reviewers for their feedback, Yuting Gu for her large contribution to the visualization of our results, the GVU center and all members of the Graphics Lab at Georgia Tech. This work was funded by NSF grants IIS 1130934 and EFRI-M3C 1137229 as well as NIH grant R01 114149.

References

BAI, Y., AND LIU, C. K. 2014. Coupling cloth and rigid bodies for dexterous manipulation. In *Proceedings of the Seventh International Conference on Motion in Games*, ACM, New York, NY, USA, MIG '14, 139–145.

BAI, Y., AND LIU, C. K. 2014. Dexterous manipulation using both palm and fingers. In *ICRA*, IEEE, 1560–1565.

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

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

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

GLEICHER, M. 1998. Retargetting motion to new characters. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '98, 33–42.

HARADA, K., KAJITA, S., KANEKO, K., AND HIRUKAWA, H. 2003. Pushing manipulation by humanoid considering two-kinds of ZMPs. In *IEEE International Conference on Robotics and Automation*, vol. 2, 1627–1632.

HARMON, D., VOUGA, E., TAMSTORF, R., AND GRINSPUN, E. 2008. Robust treatment of simultaneous collisions. *ACM Trans. Graph. 27*, 3 (Aug.), 23:1–23:4.

HO, E. S. L., AND KOMURA, T. 2009. Character motion synthesis by topology coordinates. *Comput. Graph. Forum 28*, 2, 299–308.

HO, E. S. L., KOMURA, T., AND TAI, C.-L. 2010. Spatial relationship preserving character motion adaptation. *ACM Trans. Graph. 29*, 4 (July), 33:1–33:8.

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

KALLMANN, M., AUBEL, A., ABACI, T., AND THALMANN, D. 2003. Planning collision-free reaching motions for interactive object manipulation and grasping. *Comput. Graph. Forum 22*, 3, 313–322.

KIM, M., HYUN, K., KIM, J., AND LEE, J. 2009. Synchronized multi-character motion editing. *ACM Trans. Graph. 28*, 3 (July), 79:1–79:9.

KOSUGE, K., YOSHIDA, H., FUKUDA, T., SAKAI, M., AND KANITANI, K. 1995. Manipulation of a flexible object by dual manipulators. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 1, IEEE, 318–323.

KRY, P. G., AND PAI, D. K. 2006. Interaction capture and synthesis. *ACM Trans. Graph. 25*, 3 (July), 872–880.

LAVALLE, S. M., AND KUFFNER, J. J. 2001. Randomized kinodynamic planning. *I. J. Robotic Res. 20*, 5, 378–400.

LIU, C. K., AND JAIN, S. 2012. A short tutorial on multibody dynamics. Tech. Rep. GIT-GVU-15-01-1, Georgia Institute of Technology, School of Interactive Computing, 08.

LIU, C. K., HERTZMANN, A., AND POPOVIĆ, Z. 2006. Composition of complex optimal multi-character motions. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '06, 215–222.

MACKLIN, M., MÜLLER, M., CHENTANEZ, N., AND KIM, T.-Y. 2014. Unified particle physics for real-time applications. *ACM Transactions on Graphics (TOG) 33*, 4, 153.

MIGUEL, E., AND OTADUY, M. A. 2011. Efficient simulation of contact between rigid and deformable objects. In *Multibody Dynamics, ECCOMAS Thematic Conference*.

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.

NARAIN, R., SAMII, A., AND O'BRIEN, J. F. 2012. Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. Graph. 31*, 6 (Nov.), 152:1–152:10.

NARAIN, R., PFAFF, T., AND O'BRIEN, J. F. 2013. Folding and crumpling adaptive sheets. *ACM Trans. Graph. 32*, 4 (July), 51:1–51:8.

NISHIWAKI, K., YOON, W.-K., AND KAGAMI, S. 2006. Motion control system that realizes physical interaction between robot's hands and environment during walk. In *International Conference on Humanoid Robots*, 542–547.

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. H. 2009. Implicit contact handling for deformable objects. *Comput. Graph. Forum 28*, 2, 559–568.

POLLARD, N. S., AND ZORDAN, V. B. 2005. Physically based grasping control from example. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '05, 311–318.

SHINAR, T., SCHROEDER, C., AND FEDKIW, R. 2008. Two-way coupling of rigid and deformable bodies. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '08, 95–103.

SIFAKIS, E., SHINAR, T., IRVING, G., AND FEDKIW, R. 2007. Hybrid simulation of deformable solids. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '07, 81–90.

TAKUBO, T., INOUE, K., AND ARAI, T. 2005. Pushing an object considering the hand reflect forces by humanoid robot in dynamic walking. In *IEEE International Conference on Robotics and Automation*, 1706 – 1711.

TAMEI, T., MATSUBARA, T., RAI, A., AND SHIBATA, T. 2011. Reinforcement learning of clothing assistance with a dual-arm robot. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, IEEE, 733–738.

TANG, M., MANOCHA, D., AND TONG, R. 2010. Fast continuous collision detection using deforming non-penetration filters. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, I3D '10, 7–13.

WANG, H., AND KOMURA, T. 2012. Manipulation of flexible objects by geodesic control. In *Computer Graphics Forum*, vol. 31, Wiley Online Library, 499–508.

WANG, H., RAMAMOORTHI, R., AND O'BRIEN, J. F. 2011. Data-driven elastic models for cloth: Modeling and measurement. *ACM Transactions on Graphics 30*, 4 (July), 71:1–11. Proceedings of ACM SIGGRAPH 2011, Vancouver, BC Canada.

WANG, H., SIDOROV, K. A., SANDILANDS, P., AND KOMURA, T. 2013. Harmonic parameterization by electrostatics. *ACM Trans. Graph. 32*, 5 (Oct.), 155:1–155:12.

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 Trans. Graph. 32*, 4 (July), 43:1–43:14.

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.

YAMANE, K., KUFFNER, J. J., AND HODGINS, J. K. 2004. Synthesizing animations of human manipulation tasks. *ACM Trans. Graph. 23*, 3 (Aug.), 532–539.

YE, Y., AND LIU, C. K. 2012. Synthesis of detailed hand manipulations using contact sampling. *ACM Trans. Graph. 31*, 4 (July), 41:1–41:10.

YOSHIDA, E., BELOUSOV, I., ESTEVES, C., AND LAUMOND, J.-P. 2005. Humanoid motion planning for dynamic tasks. In *IEEE International Conference on Humanoid Robotics*.

ZHAO, W., ZHANG, J., MIN, J., AND CHAI, J. 2013. Robust realtime physics-based motion control for human grasping. *ACM Trans. Graph. 32*, 6 (Nov.), 207:1–207:12.