Interactive Animation of Precision Manipulations with Force Feedback

Ben Humberston and Dinesh K. Pai
University of British Columbia

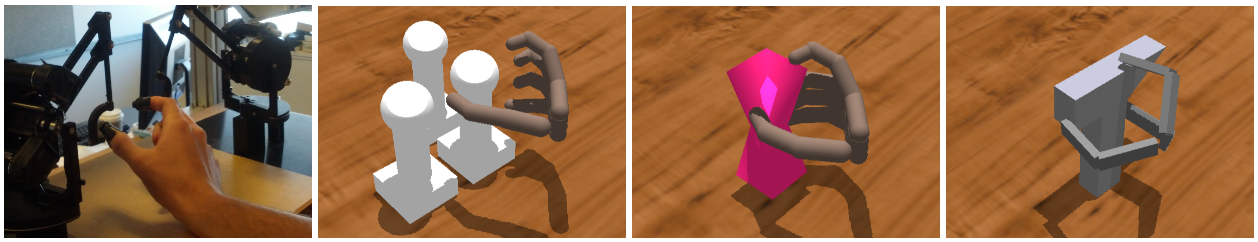


Fig. 1.

Humans show effortless dexterity when controlling their own hands. Specifying the motion of graphical or robotic manipulators, however, remains a difficult task that requires animation expertise or complicated measurement hardware, and the resulting motions have limited generalization capabilities. We present a real-time, adaptive interface based on a bidirectional haptic interface for the animation of precision reaching and carrying motions using the hand. Using our interface, the animator records a high-level grasper trajectory and a full hand pose is automatically determined by compliantly interacting with an animated scene. Haptic feedback enables intuitive control by mapping interaction forces from the full animated hand down to the reduced animator feedback space, invoking the same sensorimotor control systems utilized in natural precision manipulations. We outline an approach for online, adaptive shaping of the animated hand based on prior interactions, resulting in more functional and appealing motions. The effectiveness of our interface is investigated in a user study that examines the grip forces, interaction trajectories, and task performance of non-expert participants. Comparing the quality of motions produced with and without force rendering, haptic feedback is demonstrated to be critical for efficiently communicating contact forces, inertia, and other dynamic properties to the user. Our adaptive shaping technique is shown to yield more natural interaction qualities such as simultaneous contact timing and grip force balance across fingertips.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Animation*

DOI 10.1145/XXXXXXX.YYYYYYY
http://doi.acm.org/10.1145/XXXXXXX.YYYYYYY

General Terms:

Additional Key Words and Phrases: Hand animation, grasping, sensorimotor control

ACM Reference Format:

1. INTRODUCTION

Many animations require characters to grasp and manipulate objects in the scene using their hands. Such animations are remarkably challenging, both due to the large number of degrees of freedom of the hand and the difficulty complying realistically with objects in the scene. Current solutions for creating such interactions use three distinct approaches (see Sec. 3 for a review). The first, most common, approach places the creative burden entirely on an expert animator, who uses traditional modeling software to create complete hand motions for a required sequence. This gives maximum flexibility in the motion produced, but is slow and labor-intensive. The second approach, motion capture, simplifies production but is limited to motions that can be physically performed by an actor, and is also labor-intensive in a different way. The third, more recent, approach is to synthesize complete hand motions in a procedural manner. This approach can simulate the physics of interaction well, but does not exploit knowledge of human manipulation skills or animator expertise.

Our goal is to develop a new intuitive animation interface that utilizes the talents of animators, while exploiting knowledge of both the physics and the physiology of human manipulation skills to simplify their task.

The fundamental shortcoming of previous approaches is that the animator focuses purely on kinematic quantities, such as the motions of objects or fingers, and does not get any feedback on fingertip forces. Such forces are essential for skilled manipulation. They are rapidly processed by the human sensorimotor system to control hand movement [Jones and Lederman 2006]. We address this shortcoming by using haptic interfaces to display forces to the animator's hand. Animators can exploit this information intuitively, without conscious thought.

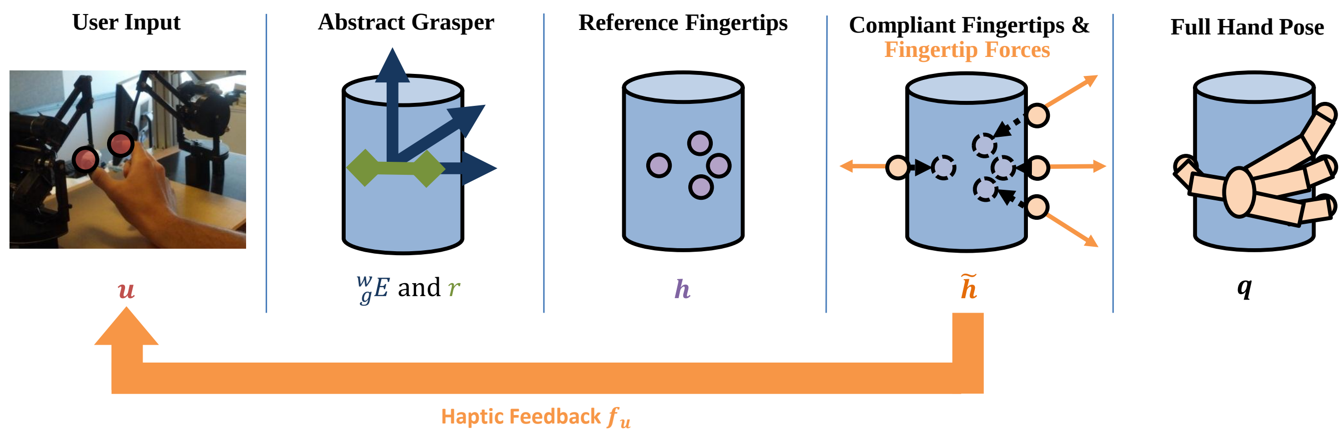


Fig. 2. The steps in constructing a full hand pose based on user input

A second major shortcoming is that previous approaches require animators to control the large number of degrees of freedom of the hand directly. While introducing haptic feedback to an animation workflow is advantageous, current haptic systems are a far cry from the holodeck of science fiction. It is both difficult and expensive to provide high fidelity force feedback to the whole hand.

We address these shortcomings by exploiting a specific type of dimensionality reduction observed in human grasping and manipulation. Many multifingered human grasps can be functionally abstracted as grasping with two virtual fingers [Iberall et al. 1986; Baud-Bovy and Soechting 2001]. It has been hypothesized that these virtual fingers encode the high level neural control of the hand, which is then elaborated at more peripheral levels of the nervous system to generate detailed hand movements For our purposes, the two virtual fingers provide a natural, but abstract, *grasper* with which the animator can interact with objects in the scene *with force feedback*. It is possible to provide high quality force feedback to two fingers using existing commercial haptic interfaces, such as the Phantom devices (SensAble/Geomagic) we use in our system. An additional benefit of adding this level of abstraction is ease of generalization, to different types of hands (ranging from human hands to three-fingered claws). Note that the animator still sees the full hand interacting with the 3D scene, in real time, synchronized with the forces on the fingers. With this system, animators can control the high degree of freedom hand intuitively, without much training.

One important question remains: how to map low dimensional user interaction to high dimensional 3D interaction with a full multifingered hand, in a natural way? Briefly, we achieve natural-looking movements in two ways: starting with a nominal mapping, fingertip positions are modified both *reactively*, based on contact with scene objects, and *proactively* based on prior interaction knowledge. This interaction knowledge is acquired with a simple grasp adaptation algorithm using a small number of training grasps performed by an animator. After training, this knowledge can be reused for different interactions and by different animators. It also produces realistic preshaping of the hand when it approaches an object, without expensive geometric computations.

Our main contributions are: (1) An interface for interactive animation of hands interacting with 3D objects. It provides force feedback signals that are very important for intuitive manipulation of objects with the hand. (2) A method for bidirectional mapping of motions and forces between the low dimensional physical user interface and high dimensional animated hands. (3) A process for automatically adapting the pose of an animated hand based on prior sampled knowledge and the current interaction context, which simplifies the creation of new, rich interactions using our interface.

The rest of the paper is organized as follows. After providing a brief overview of how our interface works in Section 2, we review relevant work in hand animation in Section 3. We explain in detail how our interface maps animator control input into full hand motions and reflects forces on the animated hand back to the animator in Section 4. We discuss the notion of adaptive grasp shaping and provide details on our current implementation of this concept in Section 5. A sampling of animations created using our interface as well as the results of our user study are given in Section 6. Some of the limitations of our interface are listed in Section 7, and we conclude with a discussion of the possible future enhancements in Section 8.

2. SYSTEM OVERVIEW

We describe how, via our interface, an animator actively controls an animated hand interacting with a digital scene with visual and haptic feedback. Figure 3 illustrates the workspace setup, which is suitable for a desktop setting and utilizes a pair of Phantom Premium 1.0 devices. The thumb and index finger of one hand are inserted into thimbles on the endpoint of each device. The position of these endpoints are tracked with high precision by rotary encoders integrated into the device motors. In comparison to vision-based motion capture systems, minimal calibration time is needed before recording an interaction, and there is no risk of losing data to occlusions or incorrectly inferred poses during complex manipulations.

After loading a digital scene, the animator can practice, record, and review a motion sequence while monitoring the interaction on a stereo display, which assists in accurately gauging depth in the scene. The animator may define basic shaping properties of the hand using on-screen controls, such as how much individual fingertips respond to an input grasp aperture and static offsets from the center of a grasp. Once the animator is satisfied with a recorded motion, the trajectory of the animated hand and other objects in the scene are exported to disk for rendering.

Figure 2 illustrates the pipeline which constructs the full animated hand pose at each interactive control update. User input $\boldsymbol{u}$ consists of the position and, optionally, the orientation of each device endpoint (we use a custom encoder gimbal as shown in Fig-

(a) Workspace for our interface (b) Custom gimbal for index finger

Fig. 3.

ure 3b for this purpose). The user input controls the state of an *abstract grasper*. Think of this grasper as the most basic representation of a grasping motion; it includes a rigid transform ${}^{w}_{g}E$ as well as an aperture r. This grasper must be instantiated as a particular hand morphology in order to interact with the scene. To accomplish this, several reference fingertips $\boldsymbol{h}$ are arrayed in the grasper frame based on the current aperture and shaping parameters for the grasp. Each reference fingertip $\boldsymbol{h}_i$ functions as the pose which an animated fingertip assumes in the absence of contact with the environment. However, while gripping an object, this reference pose will penetrate its surface and exhibit otherwise non-physical behavior. Thus, the true transforms of the animated fingertips are determined using a set of compliant fingertips $\tilde{\boldsymbol{h}}$ which track the reference pose while still respecting non-penetration, friction, and other constraints in a scene. Finally, full hand pose $\boldsymbol{q}$ (joint angles) is found via inverse kinematics at the end of each update and displayed to the animator.

Since it is our belief that manipulations using the hand are as much a tactile experience as they are visual, we use haptic feedback to help communicate important information about the state of the interaction back to the animator. When the fingertips of the animated hand contact objects in the scene, rapidly updated feedback force are applied to the animator's fingertips which simulate the contact with a solid surface, including sensations of the object mass, friction, surface shape, and other physical aspects. As demonstrated by our user study in Section 6.2, motions created with this feedback are more natural in appearance than those created using only visual guidance. Our current solution uses a soft fingerpad approximation which determines explicit contact forces that allow the animated hand to grip and manipulate objects. Since the number of animated fingertips is different than the animator's input space, these forces are reflected back to the animator in a dimension-reducing transform that is mediated by grasp shaping parameters and the abstract grasper.

Many simple motions may be created using the reactive version of our interface described thus far. However, more realistic and functional manipulations are made possible by proactive adaptation of the grasp shape based on previously sampled interactions with a particular object.

3. RELATED WORK

DKP moved this from Intro.. merge in?

These existing approaches are imperfectly suited for animating rich interactions with the hand for two primary reasons. First, they neglect to tap into the powerful touch-driven control pathways used by humans in real grasping interactions. Previous studies (TODO: citation) have shown that the touch modality is even more important than vision when executing a grasping task. Haptic feedback technology has proven to be a helpful augmentation when users are executing manipulation tasks in virtual environments (TODO: Kuchenbecker or peg-in-hole study citation). We expect that animations should be both easier to create and more natural in appearance when produced via a haptic interface.

Due to mechanical constraints on device design, however, there are necessary tradeoffs between measurement completeness, feedback quality, and the intrusion of device bulk on interactions. For example, the CyberTouch glove (CyberGlove Systems) records more complete hand pose information than the reduced end effector measurement from a Phantom haptic device (SensAble/Geomagic), but it provides only light vibrotactile feedback compared to the Phantom's compelling grounded forces. A related glove-based solution, the CyberForce (CyberGlove Systems), is able both to recreate a limited range of feedback forces and to record full hand poses; but it does so by requiring the use of a bulky interface which limits the naturalness of interactions.

TODO: Merge in -bh A popular class of algorithms in point-based haptic rendering is known as *god-object* or *proxy* methods [Zilles and Salisbury 1995]. In addition to the measured pose of a user's fingertip which may penetrate the surface of virtual objects, known as the haptic interface point (HIP), these algorithms maintain a *proxy* fingertip which obeys constraints on object penetration, friction, or other physical factors in the scene. Feedback forces are applied to the user which are proportional to the magnitude of the difference between the HIP and the proxy pose. This creates a stable coupling between the external world and a virtual scene.

♠ Filling this out & providing details on each work is a TODO -bh ♠

TODO: Works from hand animation using IR mocap, vision, or glove-based systems

—[Ullmann and Sauer 2000]
—[Wang and Popović 2009]
—[Zhao et al. 2012]

Algorithmic Hand Animation Works:

—[Rijpkema and Girard 1991]
—[Sanso and Thalmann 1994]
—[Pollard and Zordan 2005]
—[Kry and Pai 2006]
—[Sueda et al. 2008]
—[Liu 2009]
—[Ye and Liu 2012]
—[Mordatch et al. 2012]
—[Andrews and Kry 2012]

TODO: Works on grasping in robotics
TODO: Haptic works on grasping

—[Barbagli et al. 2004]
—[Garre et al. 2011] (note how they concede that CyberGrasp gives poor feedback, limiting dexterity)
—[Pacchierotti et al. 2012]
—[Mulatto et al. 2013] (somewhat close cousin to our approach)

Subjects appear to use similar control pathways in haptic VR as in real life: [Bianchi et al. 2010]

TODO: Works from telemanipulation that are similar in spirit to our setup

TODO: Hodgins work for adding expressive hand gestures as example of why hand motions are important

TODO: Regarding [Ye and Liu 2012], it's a nice example of combining high level user input motion w/ low level automation from computer. However, note that we have a different approach in that we capture a "grasper trajectory" (pose & aperture) from which the hand pose is inferred backward (fingertips-to-wrist) rather than forward using contact sampling (wrist-to-fingers). Additionally, our technique is designed for real-time, interactive authoring

4. METHODS

In this section we describe details of each stage of the process shown in Figure 2 and outlined in Section 2. Our software implementation is built on top of a modified version of the CHAI 3D haptic environment library [F. Conti and Sewell 2005] and uses the Open Dynamics Engine [Smith 2008] to simulate rigid body dynamics in the virtual scene. Digital objects may be rendered both visually and haptically in the workspace using specified mass and frictional properties. Captured real-time interactions are exported to disk as skeletal animations for integration into a complete animated sequence.

4.1 Reference Pose Control

Since the space of our user input and feedback is of reduced dimension compared to the configuration space of the full animated hand, we use an augmented version of the proxy method that translates between these different spaces. Rather than associate each device endpoint reading with a single HIP, we employ an abstracted kinematic mapping $\boldsymbol{h}(\boldsymbol{u}, \boldsymbol{\theta}(\boldsymbol{c}))$ which transforms user input $\boldsymbol{u}$ into a set of *reference fingertips* $\boldsymbol{h}$ of the animated hand. The animated hand may have any number of fingers n, but it is generally greater than two (eg: $n = 5$ fingers are used for anthropomorphic hands). $\boldsymbol{\theta}(\boldsymbol{c})$ are parameters which affect the intrinsic shape of the generated reference pose and which may vary based on the information in some interaction context $\boldsymbol{c}$; for brevity we will write this as $\boldsymbol{\theta}$. In the simplest case, these parameters are statically defined to produce a basic circular grasping shape $\boldsymbol{\theta}_d$. However, as discussed in Section 5, dynamically setting $\boldsymbol{\theta}$ enables adaptive grasp shapes that respond to contextual information about the interaction.

To make this concrete, observe that user input $\boldsymbol{u}$ in our setup consists of the position and orientation of the user's thumb (a) and index finger (b), all provided in the virtual world frame w:

$$\boldsymbol{u} = \begin{bmatrix} {}^w_aE \\ {}^w_bE \end{bmatrix}, \text{where} \tag{1}$$

$${}^w_aE = \begin{bmatrix} {}_a\boldsymbol{R} & {}_ap \\ 0 & 1 \end{bmatrix} \text{ and } {}^w_bE = \begin{bmatrix} {}_b\boldsymbol{R} & {}_bp \\ 0 & 1 \end{bmatrix}$$

It should be noted that, depending on the end effector hardware used, the user's fingertip orientations may or may not be encoded. In this case, it is assumed that the the rotational component of these transforms is identity.

When controlling an animated hand with n fingers, $\boldsymbol{h}$ is a concatenation of n affine transforms which determine the position and orientation of each animated fingertip in the world frame:

$$\boldsymbol{h}(\boldsymbol{u}, \boldsymbol{\theta}) = \begin{bmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \\ \vdots \\ \boldsymbol{h}_n \end{bmatrix} = \begin{bmatrix} {}^w_1E \\ {}^w_2E \\ \vdots \\ {}^w_nE \end{bmatrix} \tag{2}$$

Here, ${}^w_iE$ is the 4×4 matrix specifying the homogeneous coordinates of the ith fingertip with respect to the animated world frame.

The specific form of the mapping in Equation 2 is a critical choice. Though there is some abstraction of input based on the increase in dimensionality, it should still allow the animator to control the animated hand in as intuitive a fashion as possible. We found that control was highly compelling when input $\boldsymbol{u}$ is first translated into the state of an *abstract grasper*, consisting of a scalar grasp aperture r and a rigid pose ${}^w_gE$, where g is the reference frame of the grasper. The aperture value is proportional to the distance between the animator's fingertips:

$$r(\boldsymbol{u}) = \| {}_ap - {}_bp \| \tag{3}$$

Via ${}^w_gE$, the grasper is placed at the average position of the thumb and index finger inputs, scaled for workspace convenience by some s, if desired (we use $s = 2.5$). The grasper orientation ${}^w_g\boldsymbol{R}$ is defined based on the line between the animator's two control fingertips and the encoded pointing direction of the animator's index fingertip, ${}_b\boldsymbol{e}_1$: ♠

♠ To Dinesh: The below equation is updated as of 8.3.2013 to properly support the gimbal input. Is the explanation clear? The math looks a bit thick but the concept is quite simple (it's just what you recommended for how to use the gimbal)...should I drop the math details and provide a slightly fuller verbal description instead? -bh ♠

$${}^w_gE(\boldsymbol{u}) = \begin{bmatrix} {}^w_g\boldsymbol{R} & {}^w_gp \\ 0 & 1 \end{bmatrix} \tag{4}$$

$${}^w_gp = \frac{s}{2}({}_ap + {}_bp), \; {}^w_g\boldsymbol{R} = \begin{bmatrix} {}_g\boldsymbol{e}_1 & {}_g\boldsymbol{e}_2 & {}_g\boldsymbol{e}_3 \end{bmatrix}$$

$${}_g\boldsymbol{e}_1 = \frac{{}_ap - {}_bp}{\| {}_ap - {}_bp \|}, \; {}_g\boldsymbol{e}_2 = \frac{{}_g\boldsymbol{e}_1 \times {}_b\boldsymbol{e}_1}{\| {}_g\boldsymbol{e}_1 \times {}_b\boldsymbol{e}_1 \|}, \; {}_g\boldsymbol{e}_3 = {}_g\boldsymbol{e}_1 \times {}_g\boldsymbol{e}_2$$

We call ${}_g\boldsymbol{e}_1$ the *grasp axis* and the plane defined by ${}_g\boldsymbol{e}_1$ and ${}_g\boldsymbol{e}_2$ the *grasp plane*.

Once this extrinsic grasper frame g is defined, the reference pose of individual animated fingertips within the frame are set based on the grasp aperture and grasp shape parameters $\boldsymbol{\theta}$. We use an aperture scale factor σ_i and cylindrical coordinates for the shaping parameters, which enables arbitrary placement of fingertips within the grasper frame:

$${}^g_iE(r, \boldsymbol{\theta}_i) = \begin{bmatrix} {}^g_i\boldsymbol{R} & \begin{matrix} (\sigma_i r + \rho_i)\cos(\phi_i) \\ (\sigma_i r + \rho_i)\sin(\phi_i) \\ z_i \end{matrix} \\ 0 & 1 \end{bmatrix} \tag{5}$$

$$\boldsymbol{\theta}_i = \begin{bmatrix} \sigma_i \\ \rho_i \\ \phi_i \\ z_i \end{bmatrix}, \boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}_1 \\ \boldsymbol{\theta}_2 \\ \vdots \\ \boldsymbol{\theta}_n \end{bmatrix} \tag{6}$$

Offsetting the radial component by the scaled input aperture $\sigma_i r$ allows the animator to open or close all of the fingers about the center of the grasp. Rotation matrix ${}^g_i\boldsymbol{R}$ simply orients the fingertip such that the "pad" half points toward the local grasp origin.

The world-frame reference pose of each fingertip i is found by combining equations 4 and 5:

$${}^w_iE(\boldsymbol{u}, \boldsymbol{\theta}) = {}^w_gE(\boldsymbol{u})\, {}^g_iE(r, \boldsymbol{\theta}) \tag{7}$$

Note how the motion is broken into extrinsic grasper motion in the first term and intrinsic shaping in the second. While the animator guides the trajectory g and aperture r of the abstract grasper, individual reference fingertips are automatically arrayed within g based on that aperture and the active shaping parameters.

4.2 Compliant Fingertip Motion

♠ Motivate a bit biologically, as in Kry and Pai 2006 ♠

The spherical fingertip proxies in our implementation are based on the soft fingerpad approximation found in [Barbagli et al. 2004]. Linear and torsional friction constraints are determined using a simple model of the fingerpad-object contact area. In each haptic update, the proxy is projected onto the object surface in the direction of the reference position, sliding tangentially if necessary until it lies within the friction cone projected upward from the reference. The proxy's rotation about the fingerpad normal is likewise restricted based on torsional friction which is proportional to the size of the friction cone and a torsional friction coefficient.

Denoting this proxy algorithm as $\mathcal{P}$, the pose $\tilde{\boldsymbol{h}}$ of the *compliant fingertips* is the concatenation of each fingertip's proxy transform at the current time:

$$\tilde{\boldsymbol{h}} = \begin{bmatrix} \tilde{\boldsymbol{h}}_1 \\ \tilde{\boldsymbol{h}}_2 \\ \vdots \\ \tilde{\boldsymbol{h}}_n \end{bmatrix}, \text{where} \tag{8}$$

$$\tilde{\boldsymbol{h}}_i \in SE(3) = \begin{cases} \mathcal{P}(\boldsymbol{h}_i) & : \text{if fingertip } i \text{ is in contact} \\ \boldsymbol{h}_i & : \text{otherwise} \end{cases} \tag{9}$$

This fingerpad approximation was chosen for its computational speed and its effectiveness for dexterous interactions as seen in previous haptic works. Other models of contact may be substituted; for example, [Kry and Pai 2006] uses a quasi-static LCP to compute the true motion of a finger in contact with the environment, given the measured reference trajectory.

♠ To Dinesh: Is this a reasonable brief description of Kry and Pai 2006? I admit that I still don't have a great understanding of the formulation in Section 7 of that work. -bh 8.5.2013 ♠

4.3 Interaction Forces

The contact force $\boldsymbol{f}_i$ and torque $\boldsymbol{\tau}_i$ experienced by each animated fingertip i are set proportionally to the difference between the finger's compliant pose $\tilde{\boldsymbol{h}}_i$ and its reference pose $\boldsymbol{h}_i$, with optional velocity damping:

$$\begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{\tau}_i \end{bmatrix} = \boldsymbol{C}(\tilde{\boldsymbol{h}}_i - \boldsymbol{h}_i) + \boldsymbol{B}(\dot{\tilde{\boldsymbol{h}}}_i - \dot{\boldsymbol{h}}_i) \tag{10}$$

♠ To Dinesh: Eq 10 is fairly sloppy notation, but I'm a bit unsure how to quickly get across the basic idea of "spring-damper in SO(3)" without introducing a good deal more variables for the reader to process. Suggestions? -bh, 6.17.2013 ♠

Many haptic applications set the proportional gain $\boldsymbol{C}$ as high as possible before feedback becomes unstable in order to simulate high surface stiffness values (eg: 1000 N/m). In contrast, we set our proportional gain $\boldsymbol{C}$ to a relatively modest constant of 200 N/m that allows the reference fingertips to significantly penetrate an object surface (on the order of a few centimeters) before contact forces become appreciable. This results in highly compliant fingers which are better suited for creating stable multifinger grasps.

The negation of these contact forces/torques are applied to the objects that each finger is contacting, which emulates the grip and manipulation forces from a real hand. By timestepping the underlying dynamics simulation, objects may be easily grasped, lifted, and otherwise manipulated in the animated scene.

4.4 Haptic Feedback

Given the final animated fingertip forces $\boldsymbol{f} = \left[\boldsymbol{f}_1^T \ \boldsymbol{f}_2^T \ \ldots \ \boldsymbol{f}_n^T\right]^T$, our goal is to provide the animator with haptic feedback which accurately captures contact events, the inertial properties of objects, and applied grip forces. Based on our hardware constraints, this requires mapping the forces across n animated fingers down to only $\boldsymbol{f}_a$ and $\boldsymbol{f}_b$, the forces on the animator's thumb and index finger. A reasonable method for doing so, based on the principle of virtual work, is to calculate the Jacobian of the current reference kinematic mapping $\boldsymbol{h}(\boldsymbol{u}, \boldsymbol{\theta})$ with respect to the input thumb and index finger positions, ${}_a p$ and ${}_b p$, and apply its transpose to the animated fingertip forces:

$$J = \begin{bmatrix} \frac{\partial \boldsymbol{h}}{\partial\, {}_a p} & \frac{\partial \boldsymbol{h}}{\partial\, {}_b p} \end{bmatrix} \tag{11}$$

$$\boldsymbol{f}_u = \begin{bmatrix} \boldsymbol{f}_a \\ \boldsymbol{f}_b \end{bmatrix} = J^T \boldsymbol{f} \tag{12}$$

Although there is a necessary loss of some nuanced force information in this conversion, we found that the essential sensory qualities of the interaction are retained. The animator is able to sense and respond to force events across all fingers of the animated hand in a largely intuitive fashion. Without this haptic information, it is difficult to emulate several aspects of an interaction using vision alone, as highlighted by our user study (see Section 6.2).

4.5 Full Hand Pose

After determining the final compliant transform $\tilde{\boldsymbol{h}}_i$ of each animated fingertip i for the current timestep, the complete hand pose $\boldsymbol{q}$, consisting of the angles on each joint of an articulated hand, is generated via inverse kinematics. Typically, a base wrist transform is defined which is offset some distance from the origin of the grasper frame g and each finger is modeled as an articulated chain connected to this wrist and using the fingertip as a target end effector. By constraining the motion of the wrist, it is possible to produce animations that switch between gross wrist and fine finger motions.

The trajectory of the articulated hand pose $\boldsymbol{q}$ is suitable for export as a skeletal animation for use in an external rendering pipeline.

5. GRASP ADAPTATION

While the interface described in Section 4 allows an animator to interactively craft sequences of object manipulation using the hand, the resulting motions lack many of the fine adjustments exhibited by human actors when grasping and manipulating objects. Various phenomena that appear in human grasping are caused by adapting the hand shape to the local geometry of an object, to its surface material properties, or even to an actor's intended action.

For instance, in human grasping, one particularly notable feature is *preshaping* of the grasp. Due to finger compliance when enclosing an object, some grasps are stable even without shaping the hand to accommodate it. However, this leads to awkward imbalances in contact timing and forces across fingers. Preshaping modulates the grasp shape to approximately match the object surface during reaching, before object contact, in order to bring about a more natural grip [Santello and Soechting 1998]. When human actors execute a grasp, they use vision and sensorimotor memory to shape their hand appropriately, and our tool should recreate that form of proactive adaptation.

In previous works, preshaping is sometimes emulated either by choosing a closest object fit from a set of predefined grasp shapes [Sanso and Thalmann 1994] or by conducting a search over physically possible hand poses using a known trajectory and dynamics model [Ye and Liu 2012]. A more appealing option is to draw from prior experiences: when encountering a grasping situation that bears *contextual similarities* to previous grasps, the hand should automatically adjust its behavior to suit the new interaction. This removes the limitations of using only predefined grasp shapes while sidestepping the biologically implausible computation whereby the grasp shape is computed from scratch during an interaction. Moveover, learning from experience makes it possible to capture an individual user's motion style.

Using our setup, the animator specifies a high-level grasper trajectory while the interface is responsible for adapting the precise grasp shape (5 and Equations 6) by setting context-dependent grasp shape parameters $\boldsymbol{\theta}$. To facilitate adaptive grasp shaping, we define an *adaptive shaping process* $\boldsymbol{\theta} = \mathcal{A}(\boldsymbol{c}, \mathcal{S})$ which determines the parameters for a given context. In addition to the active grasp context $\boldsymbol{c}$, $\mathcal{A}$ uses a set of m previously sampled shaping parameters, each paired with their associated context:

$$\mathcal{S} = \{({}^s\boldsymbol{c}, {}^s\boldsymbol{\theta}) | s = (1 \ldots m)\} \tag{13}$$

For our current purposes, grasp context may be intuitively and simply defined as the spatial relationship between the hand and the object to be grasped: as the hand approaches some part of an object, it should change shape in order to conform to the local object surface geometry. In practice, we found that the position of the abstract grasper in the object's local frame o at a particular moment during an interaction is sufficient:

$$\boldsymbol{c} \in \mathbb{R}^3 = {}^o_g p \tag{14}$$

Thus, given shaping function $\mathcal{A}$, a set of context-associated shaping samples $\mathcal{S}$, and a basic interaction context $\boldsymbol{c}$ updated at each control step, our interface will adaptively change the grasp shape during a reaching movement, yielding richer and more functional grasp motions as a result. Figure 4 illustrates this process.

In the following sections, we describe how the parameters sample set $\mathcal{S}$ is generated and our implementation of the shaping process $\mathcal{A}$.

5.1 Grasp Shape Sampling

Each sampled interaction s in $\mathcal{S}$ should specify a useful grasp shape, encoded by ${}^s\boldsymbol{\theta}$, that was employed while context ${}^s\boldsymbol{c}$ was active in a previous interaction. In fact, while the hand is grasping an object, these parameters may be inferred by observing the shape the hand actually takes, which is defined by the set of compliant fingertips $\tilde{\boldsymbol{h}}$. We calculate the shaping parameters ${}^s\breve{\boldsymbol{\theta}}_i = ({}^s\breve{\rho}_i, {}^s\breve{\phi}_i, {}^s\breve{z}_i)$ for each fingertip which match this compliant shape and normalize the radial component by subtracting the mean value of ${}^s\breve{\rho}_i$ across all fingertips. This yields the set of grasp shaping parameters ${}^s\boldsymbol{\theta}$ for the sample:

$${}^s\boldsymbol{\theta}_i = ({}^s\breve{\rho}_i - \frac{\sum_{i=1}^{n} {}^s\breve{\rho}_i}{n}, {}^s\breve{\phi}_i, {}^s\breve{z}_i) \tag{15}$$

$${}^s\boldsymbol{\theta} = \begin{bmatrix} {}^s\boldsymbol{\theta}_1 \\ {}^s\boldsymbol{\theta}_2 \\ \vdots \\ {}^s\boldsymbol{\theta}_n \end{bmatrix}$$

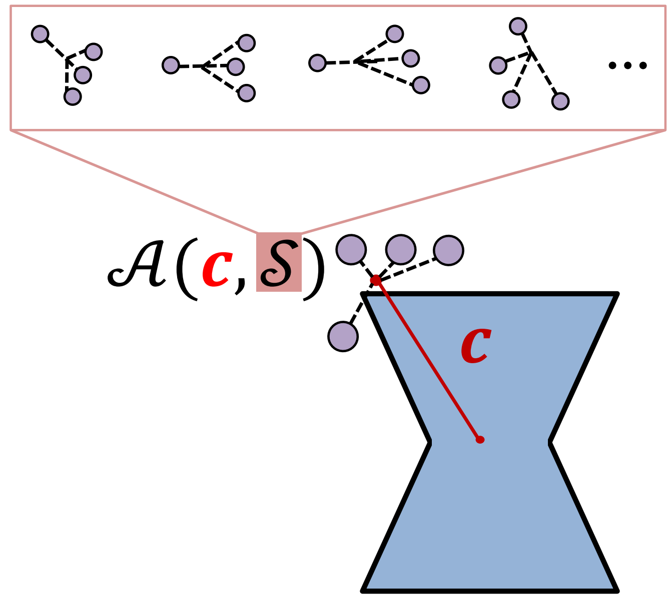


Fig. 4. The grasp shaping for a new context $\boldsymbol{c}$ is generated by $\mathcal{A}$ using the information from a set of previously sampled shaping parameters in $\mathcal{S}$.

Although determining these "ideal" shaping parameters is a purely kinematic operation when considering a particular hand pose, it may be helpful to think of it as a type of energy minimization. The best hand shape for a given context is assumed to be that which minimizes the difference in grip forces across all of the fingertips. This condition is met by setting ${}^s\boldsymbol{\theta}$ such that the hand conforms to the local object surface when the input aperture r is equal to the object's width.

Figure 5 shows an example of sampling in two dimensions. The sampled shaping parameters ${}^s\boldsymbol{\theta}$ are paired with the context ${}^s\boldsymbol{c}$ that is active at the time the sample is taken and added to the set $\mathcal{S}$. In our current solution, users are able to manually add shaping samples during an interaction session, building a library of samples that describe the relationship between the animated hand and a particular object.

5.2 Shaping Interpolation

The adaptive shaping process $\mathcal{A}(\boldsymbol{c}, \mathcal{S})$ must generate a new set of shaping parameters for the current interaction context $\boldsymbol{c}$ using the collection of shaping samples $\mathcal{S}$. Because grasp shapes tend to vary smoothly based on context (position of the grasper relative to an object), we chose an interpolation scheme using Gaussian Process Regression (GPR) as outlined in [Rasmussen 2006]. GPR provides non-linear, smooth interpolation of a target value based on the distribution of samples in the input space; in our case, this is equal to the grasp context space. Other choices of non-linear interpolation methods are also feasible; however, as discussed below in Section 8, GPR provides useful mechanisms for varying the influence of samples that we intend to exploit in future work.

Our covariance function between contexts follows the common choice of a squared exponential distribution:

$$k(\boldsymbol{c}, \boldsymbol{c}') = \sigma_f^2 \exp\left[\frac{-\|\boldsymbol{c} - \boldsymbol{c}'\|^2}{2l^2}\right] \tag{16}$$

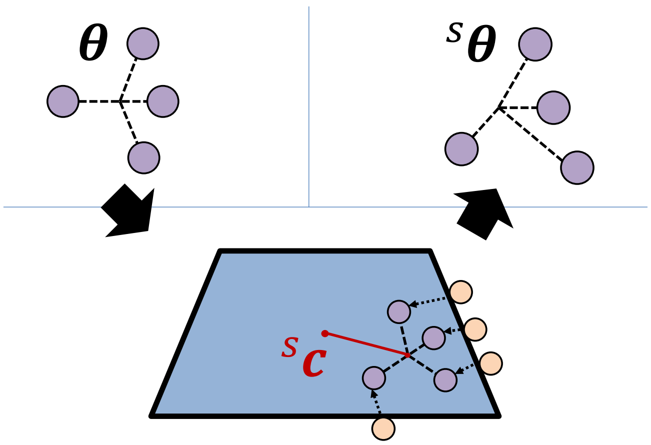


Fig. 5. During scene interactions, some prior shaping parameters $\boldsymbol{\theta}$ are used to control the hand's *reference* pose. However, when taking an adaptation sample, we calculate and store the shaping parameters $^s\boldsymbol{\theta}$ that match the hand's *compliant* pose. This effectively captures information about the object surface geometry in the local context $^s\boldsymbol{c}$.

The maximum covariance is equal to σ_f^2, and the effective width of the kernel function depends on l. These hyperparameters may be set using a maximum likelihood optimization or manually determined for a particular interaction based on the approximate size of object features and distribution of samples.

For notational convenience, we collect the shaping parameters from each of our m samples into a single parameter matrix for the sample set:

$$\boldsymbol{\Theta}_{\mathcal{S}} = \begin{bmatrix} ^1\boldsymbol{\theta} & ^2\boldsymbol{\theta} & \ldots & ^m\boldsymbol{\theta} \end{bmatrix} \tag{17}$$

Using a zero value for our prior mean on the grasp parameters would not be sensible, as this describes a grasp where all the fingertips are collapsed to the origin and oriented identically. Instead, we use the some default, user-defined grasp $\boldsymbol{\theta}_d$ as a non-zero prior mean. Thus, our adaptation process is defined as follows, where $\boldsymbol{\Theta}_d$ is a matrix whose m columns are each equal to $\boldsymbol{\theta}_d$:

$$\mathcal{A}(\boldsymbol{c}, \mathcal{S}) = \boldsymbol{\theta}_d + \left[K_* K^{-1} (\boldsymbol{\Theta}_{\mathcal{S}} - \boldsymbol{\Theta}_d)^T\right]^T \tag{18}$$

$$K_* = \begin{bmatrix} k(\boldsymbol{c}, {}^1\boldsymbol{c}) & k(\boldsymbol{c}, {}^2\boldsymbol{c}) & \ldots & k(\boldsymbol{c}, {}^m\boldsymbol{c}) \end{bmatrix}$$

$$K = \begin{bmatrix} k({}^1\boldsymbol{c}, {}^1\boldsymbol{c}) & \ldots & k({}^1\boldsymbol{c}, {}^m\boldsymbol{c}) \\ \vdots & \ddots & \vdots \\ k({}^m\boldsymbol{c}, {}^1\boldsymbol{c}) & \ldots & k({}^m\boldsymbol{c}, {}^m\boldsymbol{c}) \end{bmatrix}$$

Consult [Rasmussen 2006] for details on this regression process. The upshot is that the hand produces adapted grasp shapes when it is contextually "nearby" to previous shape samples but reverts to a basic (yet functional) grasp for situations where no samples are in range. Using this process, preshaping of the hand when in proximity to the target object arises as a natural consequence.

6. RESULTS

6.1 Animations

We present several example applications for our interface. Using the haptic interface, a manipulation sequence may be interactively performed and recorded by an animator. Figure 6 shows frames from a few sequences produced in this manner. The same two-finger hardware interface enables the control of various animated hand morphologies with different numbers of fingertips. For example, a five-fingered human hand is used in the chess piece sequence, whereas the hammer is grabbed using a robotic tripod manipulator. ♠ **This is different from our main point, which is retargeting the *object* to different interactions. Worth emphasing the difference** ♠ ♠ **To Dinesh: True. Should we just drop this paragraph and Figure entirely? -bh, 8.6.2013** ♠ After recording an interaction trajectory, it may be retargeted to a different grasped object in a manner similar to [Kry and Pai 2006] or [Ye and Liu 2012]. The compliant pose $\tilde{\boldsymbol{h}}$ is influenced both by the reference pose trajectory as well as the properties of the object with which the animated hand interacts. In order to create a retargeted sequence, we preserve the reference fingertips pose and object trajectories, replace the grasped object of the original recording with a different object, and then recalculate the compliant fingertips pose for each sampled timestep. While this cannot produce the complex finger gaiting seen in [Ye and Liu 2012], it lets an animator quickly test variations of a single recorded interaction when applied to a set of similar objects. Figure 7 shows an example of the results from retargeting.

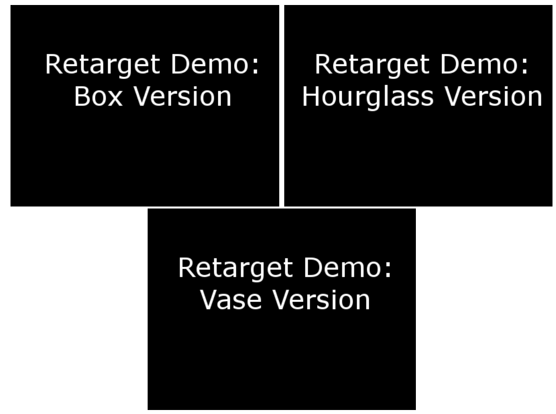


Fig. 7. A single captured interaction is retargeted to different grasped objects

Along with its role for creating animations, because of its real-time nature our interface may also function as a user-controlled manipulator in a live virtual reality environment. The accompanying video contains an example of this alternate mode, with a user manipulating various objects in a desktop VR scene.

Figure 8 highlights how adaptation of the grasp shape based on context creates more functional and natural grasps. A non-adapted hand, using a static grasp shape mapping, fails to conform to the concave object surface and yields unbalanced contact forces once the object is grasped. By enabling our adaptation process and adding just three canonical grasps to the prior shaping set $\mathcal{S}$, the hand dynamically changes its shape to accommodate the object surface, similar to a human actor.

6.2 User Study

In order to test the efficacy of haptic feedback and our adaptation scheme for creating animated hand motions, we conducted a study in which participants were asked to complete a number of simple tasks using our interface. Similar to [Bianchi et al. 2010], in each trial a virtual scene was shown to the participant using stereoscopic

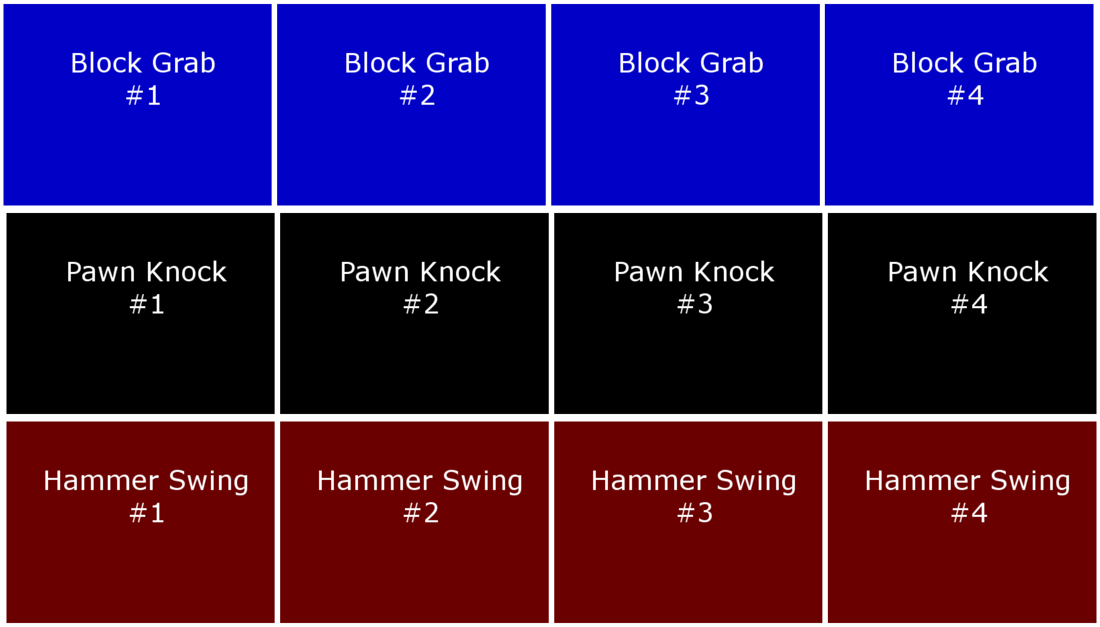


Fig. 6. Example animation sequences produced using our interface, including picking up a block, manipulating a chess piece, and swinging a hammer.

rendering that was colocated, via a mirror setup, with the participant's hand. After a brief period of familiarization with controlling the animated hand and experiencing haptic feedback from the devices, each of seven participants (age 19 - 25, 5 male) completed some or all of 4 different tasks, each separated into blocks of 10 trials. With the exception of Task 2, in each trial participants controlled the motion of a five-fingered, anthropomorphic hand. Position and force trajectories for the user input, animated hand, and objects in the scene were recorded at 100 Hz over the course of each trial. When participants made an error which prevented completion of a task, the trial was reset for a single repeat attempt before moving on to the next. In Tasks 1, 2, and 3, haptic feedback was alternatively enabled and disabled per block; whether or not feedback was enabled in the first block of each task was counterbalanced across participants. Haptic feedback was enabled in all trials for Task 4.

Task 1 (4 participants, 2 - 4 blocks per participant, 136 total trials) consisted of reaching out to grasp and lift either a 100 g or 300 g block to a specified height before replacing the block on the ground. The mass of the block in each trial was indicated to the participant by its color and size before making contact. *TODO: I'm still working on analyzing this data as of 8.1.2013 -bh. TODO: Discuss lift profiles (faster & more natural with haptics?) and GF/LF for lifting in Task 1 (aka: Experiment 4). Note how our focus on GF vs. LF is prompted by [Westling and Johansson 1984]; is lack of haptic feedback similar to 'numb fingertips'?*

Task 2 (7 participants, 2 blocks per participant, 140 total trials) was designed to test the level of control of very light contacts using the animated hand. Participants were asked, using a single animated fingertip, to scratch *as lightly as possible* across a virtual surface for a duration of at least 1 second in each trial. Figure 9 shows the magnitude of forces on the fingertip as it makes contact with the surface in each trial. For this precision task, there are pronounced benefits to adding haptic feedback: the finger contact is lighter, more consistent in magnitude, and quicker to reach a steady contact state. When guiding their motion using only visual feedback, participants were slow to react to the finger's contact with the surface and unable to reproduce and maintain a constant, light force.

In Task 3 (7 participants, 2 - 4 blocks per participant, 240 total trials), participants were asked to grasp either a 100 g or 300 g block and use it to tap once on the surface of a raised platform before replacing the block at its original position. We focus on the quality of the motion during the "tap", shown in Figure 10. As in the results for Task 2, there is a delayed and drawn out reaction to the moment of contact when haptic feedback is disabled. When the participant taps the block on the platform, haptic feedback facilitates a sharp response to the event, on the order of 10s of ms, while motions without feedback show a mushy reaction of 200 - 400 ms. This unnaturally slow response is readily observable when playing back the motion.

Based on the preceding results, we conclude that haptic feedback is a strong aid when creating motions which involve significant contact between a manipulator and the rest of the virtual scene. Although capture devices such as instrumented data gloves are useful for recording free-space hand motions, it is difficult to emulate the subtle dynamics of pressing, lifting, and manipulating physical objects without a reinforcing sensation of touch.

Finally, in Task 4 (4 participants, 4 blocks per participant, 160 total trials) we investigated the effectiveness the adaptive process described in Section 5 for producing improved balancing of contact forces. With proper adaptive shaping, we expect that contact forces will be evenly distributed across all fingertips rather than overconcentrated on only a few. In this task, participants once again were asked to grasp and lift objects to a specified height. The object to be lifted in each trial was either a concave hourglass-like primitive mesh or a simple chess piece. In a random subset of half the trials per block, the animated hand was adaptively shaped based on a predefined collection of grasp shaping samples for the relevant object; the remaining trials used a default circular grasping shape. Despite

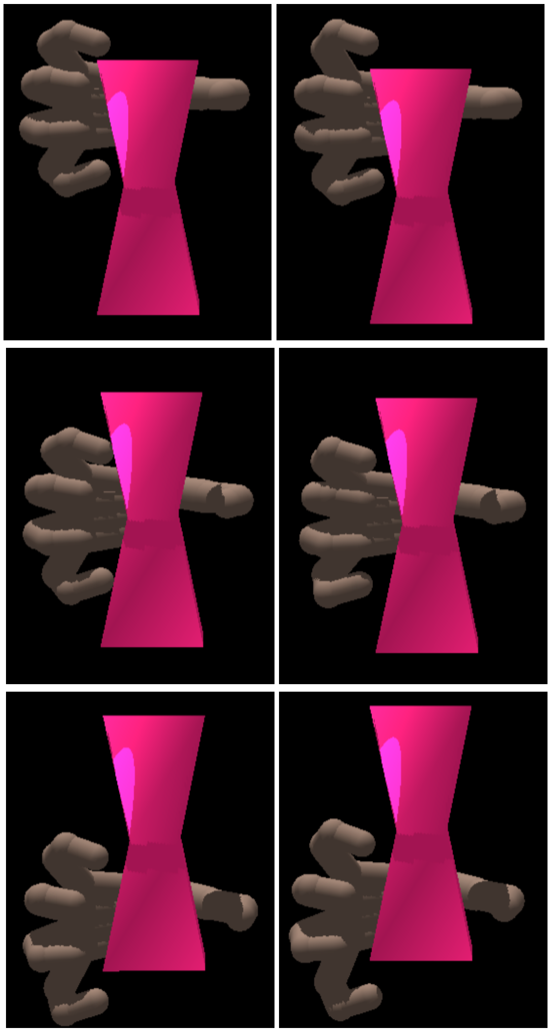
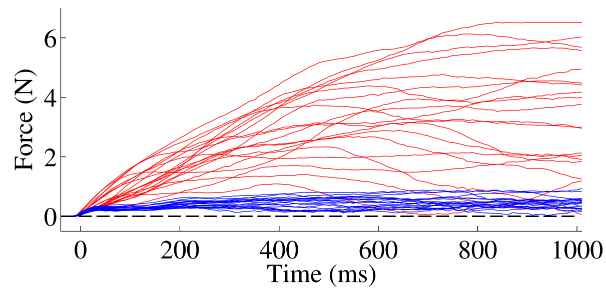
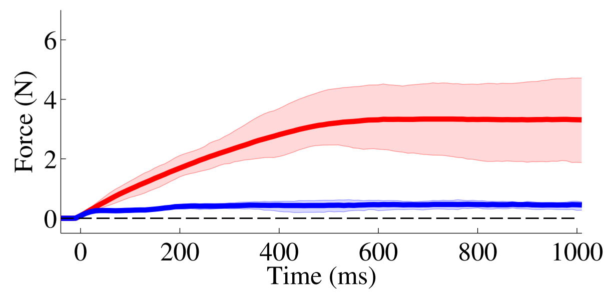


Fig. 8. Animated hand grasp shape withou (left) and with adaptation (right). Preshaping arises in the adaptively-controlled hand based on its contextual similarity to grasp shape samples from previous interactions, whereas the non-adaptive hand uses the same grasp shape in all contexts. *Draft Note: These are PLACEHOLDERS. Bad alignment, and the difference w/ and w/o adaptation is laughably subtle at the moment...*



(a) Single Participant



(b) All Participants

Fig. 9. Task 2: Contact force on fingertip while attempting lightest possible scratch on a surface, with (blue) and without (red) haptic feedback. **(a)** Trials for a representative participant. **(b)** Average across participants with first/third quartiles given.

clear visual differences in the shape of the grasp, we found that contact forces between fingertips were not significantly more balanced with adaptive shaping. We attribute this negative finding to the variety of grasp approaches employed by participants which were not adequately covered by the space of our sampled grasp shapes. Thus, though visually superior motions may be created even with a small set of grasp samples, producing quantitatively better contact forces through adaptive grasp shaping remains an open task in our framework.

6.3 Performance

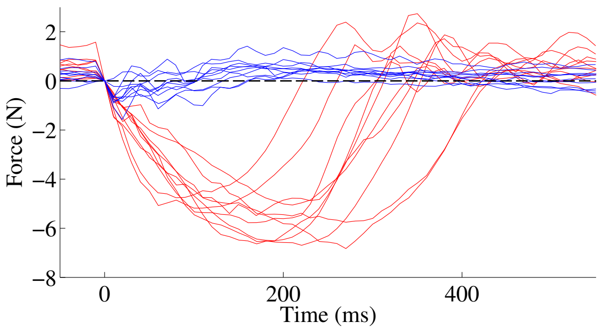
The time required for both our animator control mapping and adaptive shaping function constitutes a minimal portion of each simulation update. Simulations may be run in excess of 2000 Hz for simple scenes on a modern desktop computer, though in practice we limit the update rate to 1500 Hz to avoid some undesired hardware vibrational modes. The complexity of environments is limited primarily by the dynamics simulation and collision detection between the fingertip proxies and animated scene. Using axis-aligned bounding box or sphere trees to accelerate collisions, it is generally possible to interact stably with meshes consisting of 1000 or more faces, depending on the specific geometry. In the absence of a more sophisticated haptic collision algorithm such as that in [Barbic and James 2008], more detailed object geometries are handled by recording a sequence with a reduced-complexity version, then running an offline retargeting of the finger motion against the original geometry.

♠ What other performance details/numbers would be of interest to readers? -bh, 8.5.2013 ♠

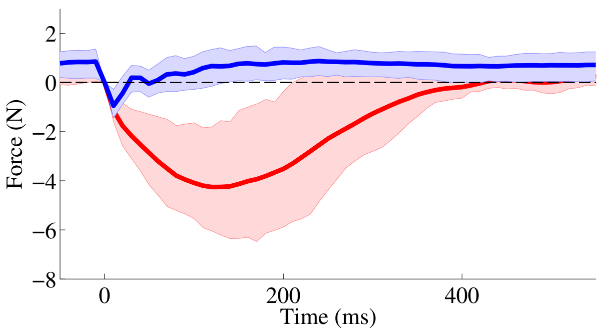
7. LIMITATIONS

At present, our interface is limited to creating precision interactions using the fingertips. Apart from the fingerpads, the animated hand is non-physical and kinematically posed after the main control update logic is completed. This limitation is due to our emphasis on producing high quality fingertip contact forces. We expect that our approach may be extended to whole hand interactions by mapping user input to compliant joint control rather than just the fingertip transforms.

The size of the virtual workspace for creating animations is limited by the shared workspace between the two Phantom devices in our hardware setup. Although small motions within the workspace are easy to produce, large translations or motions where the hand twists to a large degree may encounter the rotational limits of the device end effectors. Additionally, there is a tradeoff between the



(a) Single Participant



(b) All Participants

Fig. 10. Task 3: Magnitude of vertical force on a gripped block while using it to tap on a platform, with (blue) and without (red) haptic feedback. Time and force values are centered to zero at start of tap. **(a)** Trials for a representative participant. **(b)** Average across participants with first/third quartiles given

effective precision of the animated hand's motion and the scale of the virtual workspace depending on the scaling parameter s found in Equation 4

As noted above, the hyperparameters in the interpolation scheme of our adaptive controller must be set manually based on the size of object features. It would be preferable if these parameters changed automatically based on some object feature criteria.

8. FUTURE WORK

Our interface is able to modify its behavior based on previous experiences, but it is currently the responsibility of the user to manually cue when these experiences are sampled. Our next goal is to remove the need to explicitly add grasp shape adaptation samples and instead to sample interactions continually so that improvements in grasp shape adaptation occur in an automatic fashion. To accomplish this, it is necessary to modify our GPR kernel function so that the influence of older or lower quality samples may be reduced or eliminated. We are currently experimenting with this form of automatic adaptation.

♠

♠ TODO: This is off-message. Remove? -bh, 8.6.2013 ♠ There is also an opportunity to expand the scope of animation retargeting beyond our current solution. The adaptive shaping process $\mathcal{A}$ allows for different reference hand shapes to be created based on a single animator input trajectory. Thus, rather than recalculate only the *compliant pose trajectory* during retargeting, we may instead go one step further and create a new trajectory for the *reference pose* of the animated hand by altering the output of grasp shaping function $\mathcal{A}$. This would allow a single input trajectory to be applied to objects of even greater variation in surface geometry.

Finally, our long term plans include a generalization of the grasp context and the knowledge encoded by the adaptation process. Higher-level contextual cues such as animator task goals or functional properties of objects in the environment should modify the grasp adaptation appropriately. For example, when grasping a screwdriver, the hand should be shaped in preparation for applying torque about the tool's major axis. This also suggests that sampled interactions for one object may be reused on other objects with a similar functional purpose. Thus, we hope to investigate ways of combining prior grasp knowledge across different classes of objects, automatically calling up adaptive grasp shaping for a novel object based on its similarity to previously encountered objects.

ACKNOWLEDGMENTS

REFERENCES

ANDREWS, S. AND KRY, P. G. 2012. Policies for goal directed multi-finger manipulation. In *VRIPHYS*. 137–145.

BARBAGLI, F., FRISOLI, A., SALISBURY, K., AND BERGAMASCO, M. 2004. Simulating human fingers: a soft finger proxy model and algorithm. In *In Haptics Symposium*. 9–17.

BARBIC, J. AND JAMES, D. 2008. Six-dof haptic rendering of contact between geometrically complex reduced deformable models. *Haptics, IEEE Transactions on 1,* 1, 39–52.

BAUD-BOVY, G. AND SOECHTING, J. F. 2001. Two virtual fingers in the control of the tripod grasp. *Journal of Neurophysiology 86,* 2, 604–615.

BIANCHI, M., GRIOLI, G., SCILINGO, E. P., SANTELLO, M., AND BICCHI, A. 2010. Validation of a virtual reality environment to study anticipatory modulation of digit forces and position. In *Proceedings of the 2010 international conference on Haptics - generating and perceiving tangible sensations: Part II*. EuroHaptics'10. Springer-Verlag, Berlin, Heidelberg, 136–143.

F. CONTI, F. BARBAGLI, D. M. AND SEWELL, C. 2005. Chai 3d: An open-source library for the rapid development of haptic scenes. In *IEEE World Haptics*. Pisa, Italy.

GARRE, C., HERNANDEZ, F., GRACIA, A., AND OTADUY, M. 2011. Interactive simulation of a deformable hand for haptic rendering. In *World Haptics Conference (WHC), 2011 IEEE*. 239–244.

IBERALL, T., BINGHAM, G., AND ARBIB, M. 1986. Opposition space as a structuring concept for the analysis of skilled hand movements. *Experimental Brain Research 15*, 158–173.

JONES, L. A. AND LEDERMAN, S. J. 2006. *Human hand function*. Oxford University Press.

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

LIU, C. K. 2009. Dextrous manipulation from a grasping pose. *ACM Trans. Graph. 28,* 3 (July), 59:1–59:6.

MORDATCH, I., POPOVIĆ, Z., AND TODOROV, E. 2012. Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '12. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 137–144.

MULATTO, S., FORMAGLIO, A., MALVEZZI, M., AND PRATTICHIZZO, D. 2013. Using postural synergies to animate a low-dimensional hand avatar in haptic simulation. *Haptics, IEEE Transactions on 6,* 1, 106–116.

PACCHIEROTTI, C., CHINELLO, F., MALVEZZI, M., MELI, L., AND PRATTICHIZZO, D. 2012. Two finger grasping simulation with cutaneous and kinesthetic force feedback. In *Proceedings of the 2012 international* *conference on Haptics: perception, devices, mobility, and communication - Volume Part I*. EuroHaptics'12. Springer-Verlag, Berlin, Heidelberg, 373–382.

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*. SCA '05. ACM, New York, NY, USA, 311–318.

RASMUSSEN, C. E. 2006. Gaussian processes for machine learning. MIT Press.

RIJPKEMA, H. AND GIRARD, M. 1991. Computer animation of knowledge-based human grasping. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*. SIGGRAPH '91. ACM, New York, NY, USA, 339–348.

SANSO, R. M. AND THALMANN, D. 1994. A hand control and automatic grasping system for synthetic actors. In *Computer Graphics Forum*. Vol. 13. Wiley Online Library, 167–177.

SANTELLO, M. AND SOECHTING, J. F. 1998. Gradual molding of the hand to object contours. *Journal of Neurophysiology 79,* 3, 1307–1320.

SMITH, R. 2008. Open dynamics engine. http://www.ode.org/.

SUEDA, S., KAUFMAN, A., AND PAI, D. K. 2008. Musculotendon simulation for hand animation. *ACM Trans. Graph. (Proc. SIGGRAPH) 27,* 3.

ULLMANN, T. AND SAUER, J. 2000. Intuitive virtual grasping for non haptic environments. In *Computer Graphics and Applications, 2000. Proceedings. The Eighth Pacific Conference on*. 373–457.

WANG, R. Y. AND POPOVIĆ, J. 2009. Real-time hand-tracking with a color glove. *ACM Transactions on Graphics 28,* 3.

WESTLING, G. AND JOHANSSON, R. 1984. Factors influencing the force control during precision grip. *Experimental Brain Research 53,* 2, 277–284.

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.

ZHAO, W., CHAI, J., AND XU, Y.-Q. 2012. Combining marker-based mocap and rgb-d camera for acquiring high-fidelity hand motion data. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '12. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 33–42.

ZILLES, C. B. AND SALISBURY, J. 1995. A constraint-based god-object method for haptic display. In *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*. Vol. 3. 146–151 vol.3.

Received ; accepted