Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2008)
M. Gross and D. James (Editors)

Synthesis of Interactive Hand Manipulation

C. Karen Liu

School of Interactive Computing, Georgia Institute of Technology

Abstract

We present an interactive physics-based motion synthesis technique for creating hand manipulation across a wide variety of tasks, objects, user interventions, and stylistic preferences. Given an object being manipulated, a single pose specifying the desired initial contact, and the kinematic goals of the manipulation, our algorithm automatically generates hand-object manipulation that is responsive to unscripted external disturbances. Our algorithm simulates the dynamic coupling between a passive dynamic system and an active dynamic system by formulating a sequence of constrained optimizations. This formulation allows the user to synthesize a manipulation task by describing simple, keyframe-like kinematic goals in the domain of object configuration. The algorithm will automatically produce the hand motion that achieves the kinematic goals via coupled dynamic equations of motion.

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

1. Introduction

Animated movies, video games, and virtual environments frequently feature human characters performing seemingly mundane but remarkably dextrous manipulation tasks, such as pouring a glass of water or hammering a nail. These manipulations demand robust prehension in the face of possible disturbances, as well as fine maneuvers to achieve delicate tasks. Common solutions for free-space hand animation, such as keyframing and inverse kinematics, are ill suited for hand manipulation. First, it is difficult to keyframe or procedurally generate subtle contact phenomena between the hand and the object. Furthermore, the hand interaction depending on innate characteristics of the object, such as the surface friction, cannot be measured by kinematic quantities.

These challenges suggest that a physics-based approach might be suitable for animating hand manipulation of objects. Most state-of-the-art methods are either based on robotic controllers or controllers augmented by motion capture data. Designing active control procedures that precisely achieve complex manipulation in the presence of unpredictable contacts requires careful engineering efforts and often produces controllers that only operate in specific environments. Motion capture technology can simplify the designing process by automatically extracting physical parameters from the recorded hand motion. However, a controller designed with the aid of motion capture data is still restricted to operating in situations similar to the recorded motion.

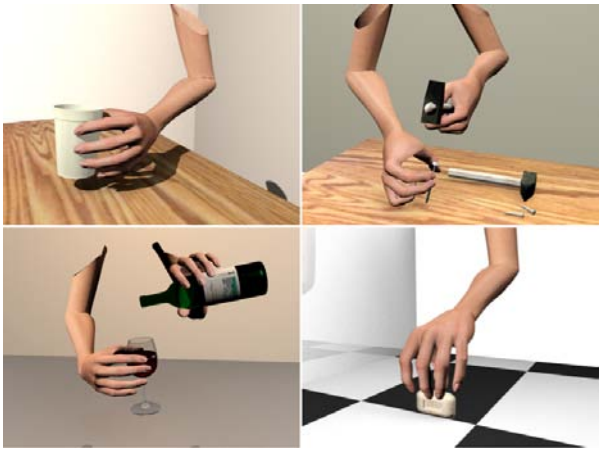

Figure 1: *A physics-base algorithm for synthesizing dextrous manipulations across a wide variety of tasks, objects, user interventions, and stylistic preferences.*

This paper describes an interactive physics-based simulation algorithm capable of synthesizing hand animations

© The Eurographics Association 2008.

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

across a wide variety of tasks, objects, user interventions, and stylistic preferences. Although we refer to the motion as hand manipulation, the scope of the motion includes the shoulder, elbow, wrist and detailed hand. The inputs to our algorithm comprise the object being manipulated, a single approximated pose specifying the desired initial contact, and the kinematic goals of the manipulation. The output is a physically realistic animation of the hand-object manipulation that is responsive to unscripted external disturbances. Internally, the algorithm transforms the hand manipulation problem to a sequence of constrained optimization problems, one for each simulation time step, that solve for the present configuration of the hand, the object, and the contact forces. The preferred initial contact and kinematic goals serve the role of objectives that guide the output motion. At first glance, forward simulation seems like a more efficient method for synthesizing the motion of a passive object. During manipulation, however, sustained contacts mean that the motion of the object can no longer be solved independently from the unknown forces exerted by the hand. By solving the hand, the object, and the contact forces in one procedure, the constrained optimization can easily enforce the dynamic coupling between the hand and the object using a realistic frictional contact model. Moreover, explicitly computing contact forces allows us to synthesize complex manipulation tasks that require careful force control.

Our formulation of hand manipulation leads to two key advantages. First, our algorithm is generic enough to synthesize manipulation animations ranging from simple tasks, such as lifting a cup, to complex tasks, such as hammering a nail. Second, the inputs that the user needs to specify to generate an interactive motion comprise a keyframe and some kinematic goals. Consequently, our system makes specifying hand animation accessible to programmers with little knowledge of physics or biomechanics. Both of these advantages derive from the property that our algorithm does not explicitly calculate internal dynamics of the hand during the simulation. This removes the need to specify or tune internal biomechanical parameters that have a highly nonlinear effect on the resulting motion.

2. Related work

Modeling the motion of the human hand has been an active research area across many disciplines. In robotics, a rich body of research literature has addressed the problems of dextrous manipulation and robust grasping. A rather comprehensive overview can be found in [VI90, BK00] and we do not attempt an exhaustive review of these topics. Grasp planning determines finger contact point locations based on the closure properties defined by Reuleaux [Reu63]. Our method solves a much simpler problem where the only closure condition for the grasp is to withstand the gravitational force. Another crucial problem in robot grasping is to determine the grasp forces in the presence of friction constraints [BHM96]. Different optimization approaches have been proposed to solve for optimal contact forces that yield stable grasps. We also formulate a nonlinear constrained optimization to solve for contact forces. However, we do not exploit the property of convexity to speed up the computation performance. In general, robotic algorithms and hand models are usually designed to effectively control an actual physical hand system, leading to requirements that might not be necessary for depicting realistic hand animation.

Hand animation is also well studied within computer graphics. Inverse kinematic systems can be used to synthesize sequences of hand poses that depict specific tasks [AN99, HBT95, KCMT00, ES03]. Kinematic-based methods can be augmented with path planning algorithms [KKKL94, KL00]. Yamene et. al. [YKH04] proposed a method for synthesizing motion for manipulation tasks where the object's motion is planned and used to impose kinematic constraints on the human's motion. Our method uses a similar idea in that we let the user animate the simpler object motion and automatically generate the more complex hand motion. However, we propose a physics-based method focusing on the dynamic aspects of the hand manipulation. Most kinematic-based methods are more suited for precisely controlled motion without dynamic feedback from the environment. Consequently, these methods cannot capture many dynamic subtleties crucial to hand-object interaction.

Dynamic simulation has also been extensively applied to the synthesis of hand animation. Most of these approaches fall under the realm of controller design [BLTK93, Sta91, Ibe97]. One obvious drawback of this approach is the overspecialization. Pollard and Zordan [PZ05] combined motion capture data and dynamic simulation to synthesize grasping and two-hand interactions. Their method extracted the passive and active parameters from the motion capture data, largely simplifying the process of controller design. In contrast, our method eliminates all dependency on the data and explicitly optimizes the contact forces as the hand interacting with the object. As a result, we are able to design a wide range of hand manipulations beyond simple grasps. Using inverse dynamics techniques to recover the muscle activations has also been successfully applied to simulating unconstrained hand motion [TSF05]. However, for constrained hand manipulation, simultaneously optimizing the actuation and estimating the contact forces might lead to suboptimal solutions. Kry and Pai [KP06] proposed a novel technique for measuring both the hand motion and the contact forces to extract the joint trajectories and joint compliances. They demonstrated that by adjusting the joint compliances the captured interaction can be retargeted to new dynamic situations. Our method gives an alternative to simulating such dynamic behaviors without relying on motion data.

Modeling the compliance of the hand is vital for synthesizing natural hand animation that exhibits both active and passive behaviors. Biomechanically based models simulate

© The Eurographics Association 2008.

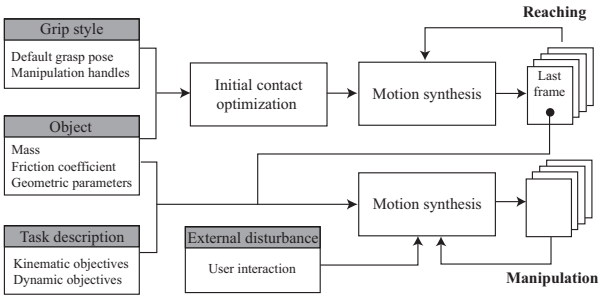

Figure 2: *Overview of the algorithm.*

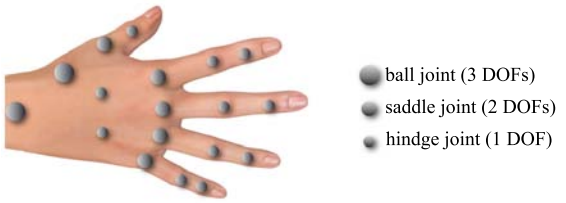

Figure 3: *Degrees of freedom on the hand model.*

the detailed interplay of musculo-tendons and muscle activations [TSF05, AHS03, Bir94], using the parameters derived directly from anatomical data. The joint compliances can also be approximated from the recorded motion and contact forces [KP06], estimated from perturbation experiments [HH97], or specified by the animator [PZ05]. To create active hand motion with passive characteristics, our framework minimizes the change of internal joint torques such that the hand cannot abruptly change its muscle activation to counteract the external perturbations.

3. Overview

Figure 2 gives an overview of our algorithm. We divide the motion synthesis into two stages: reaching for the initial grasp and the manipulation of objects. The input to the reaching stage comprises the object of interest and a grip style that describes the desired initial contact. We use the grip style to optimize for the initial grasp configuration at the point of contact with the object (detailed in Section 6). We then synthesize the reaching motion from an arbitrary hand position to the initial grasp configuration. Starting from the initial grasp (i.e. the last frame of the reaching motion), the object description, and the task description, we synthesize the hand-object manipulation under the presence of unscripted external disturbances, such as user interactions or environment perturbations.

Our algorithms allows the programmer to decouple a hand manipulation into three independent components: the task description, the grip style, and the object description. The high-level task description is a set of kinematic objectives represented as functions of the hand position and its derivatives, or a function of the object's position and its derivatives. In addition, the task description can include dynamic objectives that provide high-level control over the contact forces used for manipulation (Section 5). We define a grip style as a preferred hand pose and a set of initial contact points on the hand. The description of the object includes the mass, the inertial tensor, the friction coefficient of the surface, and the geometric parameters.

4. Motion synthesis

The heart of the reaching and the manipulation stages is an algorithm that casts the motion synthesis as a sequence of constrained optimizations. At each time step, we optimize the hand configuration, the object configuration, and the contact forces, subject to dynamic constraints and objectives provided by the high-level task description. We represent the arm and hand as an articulated body system that comprises 8 degrees of freedom (DOFs) on the arm and the 27 DOFs on the hand (Figure 3). The position of the root of the arm is fixed in the space. The object is represented as a rigid body system with six degrees of freedom.

We start out by formulating an optimization that solves for the fundamental manipulation where the hand holds an object in equilibrium under the gravity. To further simplify the problem, the contact points in this case are predefined bilateral constraints which do not slip or break off. Note that this formulation can also handle the reaching stage. We then generalize the problem to handle arbitrary unilateral contacts with a realistic friction model in section 4.2. In Section 5, we will add the manipulation objectives to formulate the full optimization problem used in the manipulation stage. Finally, in Section 6, we will discuss the details of the reaching stage.

4.1. Basic optimization

The motion of the dynamic system is solved numerically at each discrete time instance. However, instead of integrating the equations of motion, our approach formulates an optimization problem subject to the equations of motion as constraints. At each current discrete time step, the optimizer solves for the next hand configuration $\mathbf{q}^{t+1}$, the next translation and rotation of the object $\mathbf{s}^{t+1} = \{\mathbf{x}^{t+1}, \mathbf{R}^{t+1}\}$, and the current contact forces $\mathbf{f}$. The physical realism of the motion is governed by the following discretized equations of motion.

$$\dot{\mathbf{P}}(\mathbf{s}^{t+1}) = m\mathbf{g} - \sum_k \mathbf{f}_k \tag{1}$$

$$\dot{\mathbf{L}}(\mathbf{s}^{t+1}) = -\sum_k (\mathbf{p}_k - \mathbf{x}^{t+1}) \times \mathbf{f}_k \tag{2}$$

© The Eurographics Association 2008.

$$Q_{M_j} = \sum_{i \in N(j)} \mathrm{tr}\left(\frac{\partial \mathbf{W}_i}{\partial q_j}\mathbf{M}_i \ddot{\mathbf{W}}_i{}^T(\mathbf{q}^{t+1})\right) - \sum_{i \in N(j)} m_i \mathbf{g} \cdot \frac{\partial \mathbf{c}_i}{\partial q_j} - \sum_k \mathbf{f}_k \frac{\partial \mathbf{p}_k}{\partial q_j} \quad (3)$$

Based on Newton's third law, these equations couple the hand and object by applying opposite contact forces $\mathbf{f}$ at the contact points $\mathbf{p}$. Equation 1 and 2 enforce the Newton-Euler equations of motion for a rigid body in contact with the hand. $\mathbf{P}$ and $\mathbf{L}$ indicate the linear and angular momentum of the object, while $\mathbf{f}_k$ indicates the contact force at contact point $\mathbf{p}_k$. Each degree of freedom on the hand q_j is constrained by Lagrange's equation (Equation 3). Q_{M_j} denotes the internal joint torque exerted on q_j. The three terms on the right hand side compute the inertia force, the gravitational force, and the contact force in a generalized coordinate frame. $\mathbf{W}_i$ represents the transformation chain from the world coordinates to the local coordinates of body i and $N(j)$ is the set of body nodes in the subtree of DOF q_j. $\mathbf{M}_i$ denotes the mass tensor of the i^{th} body node defined in [LHP05]. $\mathbf{c}_i$ is the center of mass of the i^{th} body node in the world coordinate frame.

A contact point $\mathbf{p}_k$ can be defined as a function of the hand configuration $\mathbf{p}_k(\mathbf{q}) = \mathbf{W}_i(\mathbf{q})\mathbf{h}_k$ or a function of the object configuration $\mathbf{p}_k(\mathbf{s}) = \mathbf{x} + \mathbf{R}\mathbf{r}_k$, where $\mathbf{h}_k$ and $\mathbf{r}_k$ represent the local coordinates of $\mathbf{p}_k$ in the i^{th} body node space and object space respectively. We need one additional contact constraint to ensure that the hand and the object coincide at $\mathbf{p}_k$ in the world coordinates:

$$\mathbf{C}_{M_k}(\mathbf{q}^{t+1}, \mathbf{s}^{t+1}) = \mathbf{R}^{t+1}\mathbf{r}_k + \mathbf{x}^{t+1} - \mathbf{W}_i(\mathbf{q}^{t+1})\mathbf{h}_k = \mathbf{0} \quad (4)$$

In our formulation, we do not solve for joint torque Q_M explicitly. Instead, we minimize the time derivative of Q_M in a discrete formulation. Intuitively, this objective function discourages sudden changes in joint torques, similar to the minimal torque change model described by Uno and Kawato [UKS89, Kaw99]. We represent Equations 1 and 2 as equality constraints $\mathbf{C}_p$ and $\mathbf{C}_L$ and formulate the following optimization at each time step:

$$\underset{\mathbf{q}^{t+1}, \mathbf{s}^{t+1}, \mathbf{f}}{\text{argmin}} \quad E = \sum_j \|\dot{Q}_{M_j}(\mathbf{q}^{t+1}, \mathbf{f})\| \quad \text{subject to} \quad \begin{cases} \mathbf{C}_P(\mathbf{s}^{t+1}, \mathbf{f}) = \mathbf{0} \\ \mathbf{C}_L(\mathbf{s}^{t+1}, \mathbf{f}) = \mathbf{0} \\ \forall k,\ \mathbf{C}_{M_k}(\mathbf{q}^{t+1}, \mathbf{s}^{t+1}) = \mathbf{0} \end{cases} \quad (5)$$

4.2. Contact model

To synthesize realistic contact phenomena, we describe a friction contact model that handles general collisions between arbitrary rigid bodies. This contact model adds the following features to the basic algorithm describe in Section 4.1. First, the motion synthesizer can add new resting or colliding contacts when a collision is detected. Second, the existing contact points can slip or break off according to Coulomb's friction model. Third, the contact forces are unilateral with realistic friction coefficients.

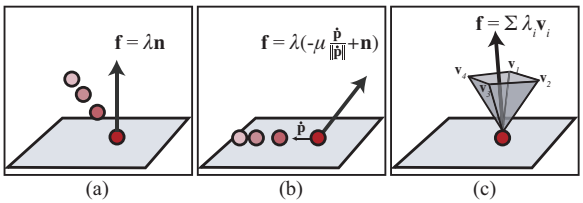

Figure 4: *Contact models.*

Standard optimization approaches usually handle a contact as an additional algebraic equation and the matching Lagrange multiplier in a dynamic system. This formulation can model unilateral contacts with realistic friction when the contact location and the timing can be determined in advance. However, it does not allow for unscripted changes in contacts, nor does it handle slipping contacts when the maximal static friction is reached.

By breaking a single trajectory optimization problem into a sequence of constrained optimizations, our algorithm can determine and add a new set of contact constraints on the fly at each time step. However, removing an existing contact constraint remains challenging with an optimization framework. When the external force overcomes the friction that holds two objects in contact, the dynamic constraint can no longer be satisfied without violating the contact constraint. To avoid this infeasible situation, the optimizer will try to maintain the contact by using excessive internal forces instead of naturally breaking the contact constraint and allowing the body to slip or break off of the surface.

We handle this issue by relaxing a contact constraint $\mathbf{C}_M$ to an inequality constraint, $\mathbf{p}(\mathbf{q}^{t+1}) > f_{sur}(\mathbf{s}^{t+1})$, that prevents penetration, and an objective function that discourages relative movement at the contact point: $G_M(\mathbf{q}^{t+1}, \mathbf{s}^{t+1}) = \|\mathbf{R}^{t+1}\mathbf{r} + \mathbf{x}^{t+1} - \mathbf{W}(\mathbf{q}^{t+1})\mathbf{h}\|$. Minimizing G_M yields a set of contact forces $\mathbf{f}$ that maintain a stationary contact point, whenever $\mathbf{f}$ exist within the friction threshold. When the large forces act against the object, the optimizer might not be able to yield a set of feasible $\mathbf{f}$ which minimizes G_M to zero. Consequently, the contact point will naturally slip or break off.

When a collision is detected, we introduce new contact force variables λ as Lagrange multipliers in the dynamic system. Depending on the type of the contact, λ has different degrees of freedom and thus the contact force is computed differently. We adapt the notion of *colliding contact* and *resting contact* defined by Baraff and Witkin [BW92]. For a colliding contact, we introduce a new free variable λ to represent the contact force: $\mathbf{f} = \lambda\mathbf{n}$ (Figure 4(a)). $\mathbf{n}$ is the normal vector of the contact surface and λ must be nonnegative to maintain a repulsive contact force. A resting

© The Eurographics Association 2008.

contact can be slipping along the surface or completely stationary depending on the relative tangential velocity. When the resting contact is stationary, the contact force is limited to the space spanned by a set of bases $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$ that approximates Coulomb's friction cone, with nonnegative basis coefficients $\lambda_0, \lambda_1, \lambda_2, \lambda_3$ (Figure 4(c)). The contact force can be represented in terms of four free variables λ: $\mathbf{f} = \mathbf{V}\lambda$. When the resting contact is slipping, the contact force is determined by the static friction coefficient μ, the direction of the contact point velocity $\dot{\mathbf{p}}$ and a free variable representing the magnitude of the normal force λ (Figure 4(b)): $\mathbf{f} = \lambda(-\mu \frac{\dot{\mathbf{p}}}{\|\dot{\mathbf{p}}\|} + \mathbf{n})$.

We augment the basic optimization described in Section 4.1 with the realistic contact model:

$$\underset{\mathbf{q}^{t+1}, \mathbf{s}^{t+1}, \lambda}{\operatorname{argmin}} \quad E = \sum_j \|\dot{Q}_{M_j}(\mathbf{q}^{t+1}, \lambda)\| + \sum_k \|G_{M_k}(\mathbf{q}^{t+1}, \mathbf{s}^{t+1})\|$$
$$\text{subject to} \quad \begin{cases} \mathbf{C}_P(\mathbf{s}^{t+1}, \lambda) = \mathbf{0} \\ \mathbf{C}_L(\mathbf{s}^{t+1}, \lambda) = \mathbf{0} \\ \forall k,\ \mathbf{p}_k(\mathbf{q}^{t+1}) > f_{sur}(\mathbf{s}^{t+1}) \end{cases} \tag{6}$$

Contact force correction. To prevent the contact forces from generating additional kinematic energy to the dynamic system, we need to consider the following two conditions. When either condition is violated, we backtrack to the previous time step, enforce additional constraints on the contact force variables, and resolve the motion. These conditions also depend on whether the object is passive or active. A passive object has no actuator that generates internal propelling forces. However, within our definition, a once passive object becomes active as soon as it is in contact with an active object.

1. If an object A breaks a resting contact from a passive object B, the normal contact force at the moment of contact breakage must be zero. If the contact force does nonzero work, we roll back the motion and enforce the contact force variables to be zero.
2. If an object A starts slipping tangentially on a passive object B, the ratio of tangential to normal contact forces must be greater than the coefficient of the static friction. If this threshold is not reached, we roll back the motion and enforce the contact force to be on the boundary of the friction cone.

In our experience, most backtracking situations can be resolved in one time step because the hand will immediately provide the required forces to minimize the kinematic objective once the contact forces are limited by additional constraints. For example, when the table "helps" the hand to lift the book by giving it a push at the moment of the contact breakage, the algorithm will roll back and formulate the same optimization with additional constraint on contact forces (i.e. the contact force between the table and the book must be zero). In order to achieve the kinematic goal of lifting the book, the optimizer must use the hand to provide the lift force. As long as the hand is capable of producing such a force, the contact point will not be detected in the next time step and the backtracking will be resolved.

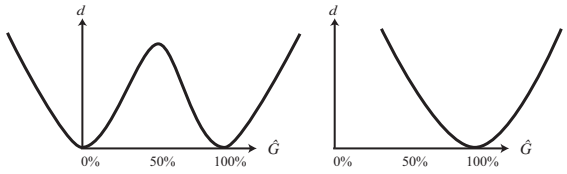

Figure 5: *Velocity profiles for reaching the goal state (left) and retaining the goal state (right).*

5. Manipulation

In this section, we describe how to augment the basic optimization problem shown in Equation 6 with kinematic goals to generate motion in the manipulation stage. We also introduce objectives to control the grip force and the lift force of the hand. These objectives are used to model two dynamic behaviors of the hand: the anticipation of the object's physical properties and the muscle responsiveness.

5.1. Kinematic objectives

A single kinematic objective consists of a general function of the hand and/or the object configurations, and a desired goal state: $G_U = f(\mathbf{q}^{t+1}, \mathbf{s}^{t+1}) - \bar{G}$. We include each kinematic objective as one additional term in the cost function E in Equation 6. For example, we can specify a desired position for the tip of the index finger as: $G_U(\mathbf{q}^{t+1}) = \|\mathbf{p}(\mathbf{q}^{t+1}) - \bar{\mathbf{p}}\|$, where $\mathbf{p}$ computes the location of index finger tip in the world space and $\bar{\mathbf{p}}$ indicates the desired location.

Natural looking hand motion must carefully balance between two competing components in the cost function: the minimal joint torque change objective, and the kinematic goals that cause changes in joint torque. Overemphasizing one set over another can lead to motions that are unnaturally abrupt or unresponsive. The presence of multiple kinematic objectives complicates the issue even further. To generate an objective function that appropriately weighs the various components, we developed an adaptive weighing algorithm that utilizes a biomechanically-inspired velocity profile to determine the appropriate tradeoff between the different components of the objective function at each simulation step.

Velocity profile. Biomechanics, neuroscience, and robotics research suggests that human arm manipulation motion often exhibits a strong bell-shape velocity profile [ABM82, AH89]. We use this slow-in and slow-out velocity profile to guide the minimization of the objective. We define a velocity profile as a function f_{vp} that maps the current completion percentage of the objective $\hat{G}_U$ to the desired progress rate

© The Eurographics Association 2008.

of G_U. The progress rate d is defined as

$$d \equiv -\dot{G}_U = G_U^{t-1} - G_U^t \tag{7}$$

A large progress rate d implies that G_U is quickly minimized. At each time step, based on the current desired progress rate $f_{vp}(\hat{G}_U)$, we modify the goal of the kinematic objective $\bar{G}$ such that the progress rate achieves the desired value $f_{vp}(\hat{G}_U)$ when G_U is minimized. The representation of the velocity profile is generic to any kinematic objective because it does not depend on the actual function of G_U.

We develop two distinct velocity profiles. *Reaching profile* is applied when the motion tries to reach a goal state, while *retaining profile* is for preserving the goal state. If the goal of the objective is initially met, we associate the objective with a retaining profile. Otherwise, we associate it with a reaching profile where the initial state and the goal state correspond to $\hat{G}_U = 0\%$ and $\hat{G}_U = 100\%$. In our formulation, we chose a scaled normal distribution to represent the retaining profile and the middle part of the reaching profile (Figure 5). By changing the variance of the normal distribution, the resulting movement exhibits different styles but is generally smooth and natural. We chose the variance to be 0.4 for all of our examples.

Adaptive reweighting. To balance the relative strength of multiple kinematic goals in the objective function, at each time step we adjust the weight of each kinematic objective based on whether its progress rate reaches the desired value determined by the appropriate velocity profile. If the objective is currently moving too slowly towards the goal ($d < f_{vp}(\hat{G}_U)$), we increase the weight of the objective. Otherwise, we decrease the weight.

5.2. Dynamic objectives

Realistic hand motion depends not only on physical constraints and task objectives, but also on a wide range of other properties that influence the contact force a hand applies to manipulate the object.

Anticipation of the object's mass and surface friction. During everyday manipulation, a person selects an appropriate parametric adjustment of the force output based on the visual identification of the object. The force output may be erroneous when the anticipation of the object's physical properties are inconsistent with those of reality. For example, if the weight of an object does not correspond to its visual appearance, the lifting movement may appear either jerky or slow. To model the anticipation of the mass and the friction coefficient of an object, we add the following objective and constraint whenever the hand attempts to lift an object against gravity.

$$\begin{aligned} G_L(\lambda) \;&=\; \|\sum_k \mathbf{V}\lambda_k - m_a\mathbf{g}\| \\ &\text{subject to} \quad \mathbf{V}\lambda_k \cdot \mathbf{n} < \mu_a \frac{m_a\mathbf{g}}{k} \end{aligned} \tag{8}$$

Equation 8 uses the estimated mass m_a and the friction coefficient μ_a to compute contact forces, $\mathbf{f} = \mathbf{V}\lambda$, that produce just enough "lift force" to hold the object against gravity, and just enough "grip force" (normal to the object surface) to prevent slippage. The objective minimizes the difference between the sum of the contact forces and the anticipated gravitational force. The inequality constraint ensures that the individual grip force is only as large as necessary to counteract gravity based on the estimated m_a and μ_a. If m_a is different from the actual mass, minimizing this objective causes abrupt adjustment in hand motion to balance Equation 6. If the hand expects a higher friction surface than the actual friction coefficient, minimizing Equation 8 does not produce enough lift force because the real friction cone is narrower than the estimated one. Consequently, the object slips out of the hand due to gravity.

Muscle responsiveness. The muscle responsiveness can be modeled by adjusting the drop-off rate of the anticipation effect. By gradually decreasing the weight of the objective and increasing the slack of the constraints in Equation 8, the contact forces produced by the hand will adjust toward the actual physical properties of the object. The programmer can vary the rate of decreasing weight and increasing slack to control the responsiveness of the hand. This same scheme can be generalized to model muscle response to any unexpected events (Figure 6). When there is a change in the external force in the dynamic system, we add an objective that maintains the current sum of the contact forces for a short period of time with a drop-off rate specified by the programmer. This simple heuristic is not anatomically realistic but serves the purpose of computer animation applications.

6. Reaching

During the reaching stage, the motion synthesizer produces a hand motion that moves from its arbitrary initial configuration to an optimal initial grasp configuration based on the input grip style and the object description. We represent a grip style as a desired default hand pose $\bar{\mathbf{q}}$ and a set of manipulation handles $\bar{\mathbf{h}}$. The main challenges in the reaching stage are computing the optimal contact points of the initial grasp and solving for a collision-free path to reach this initial grasp configuration.

The initial grasp comprises the contact points in the hand coordinates and in the object coordinates at the initiation of contact. The former is provided by the input grip style as the manipulation handles $\bar{\mathbf{h}}$, while the latter is optimized based on the object's shape, hand joint limitations, preferred pose of the hand, and the object shape, mass and the inertia.

Given a set of manipulation handles $\bar{\mathbf{h}}$ in the hand coordinates, we formulate a constrained optimization (distinct from the motion synthesizer) that solves for optimal hand configuration and contact forces at the initial grasp, subject

© The Eurographics Association 2008.

to maintaining the equilibrium against the gravity.

$$\underset{\mathbf{q},\mathbf{f}}{\text{argmin}}\ E = \|\mathbf{q}-\bar{\mathbf{q}}\| \ \text{ subject to } \begin{cases} f_{sur}(\mathbf{s}) = \mathbf{W}(\mathbf{q})\bar{\mathbf{h}} \\ \sum \mathbf{f} = m\mathbf{g} \\ \sum \mathbf{f} \times (\mathbf{W}(\mathbf{q})\bar{\mathbf{h}} - \mathbf{x}) = \mathbf{0} \end{cases} \tag{9}$$

The first constraint enforces the manipulation handles residing on the surface of the object, while the other constraints ensure that the contact forces counteract the effect of gravity without causing extra torque. The solution is biased toward a configuration similar to the desired pose $\bar{\mathbf{q}}$ described in the input grip style. The constraints and objectives in Equation 9 do not guarantee the solution configuration $\mathbf{q}$ can successfully manipulate the object according to the input task description. We rely on the programmer to provide a feasible grip style for a specific task.

Once we determine the contact locations on the object, we need to compute a set of collision-free trajectories that guide the manipulation handles toward desired contact locations. Many sophisticated path planning algorithms have successfully been applied to human arm manipulation in robotics [KKKL94, Kog94]. We propose a simple but general heuristic to produce collision-free trajectories for the hand when approaching the object.

Our key observation is that when approaching the object, the thumb and the four fingers must conform to the shape of the object while opening wide enough to clear the object before the contacts are met. In other words, the hand attempts to grasp a bigger, closer version of the same object. Therefore, instead of targeting the actual object during the reaching stage, our algorithm makes the hand grab a virtual object that is bigger and closer than the actual one. We then gradually shrink the virtual object to its actual size and move it to its actual location. Attempting to grab this shrinking and moving virtual object, the hand conforms to the object's shape while being guided toward the optimal contact locations without colliding with the object. We apply Equation 9 to find the optimal contact points on the virtual object of each size. To maintain the continuity of the contact points across sizes, we apply a sequence of optimizations in the order of size of the virtual objects (from the smallest to the largest), with each subsequent optimization biasing its solution toward the previous solution.

The advantage of this simple method is that it does not require any path planning algorithms to compute a collision-free path for each object with a specific shape. However, this algorithm does not handle objects requiring a tight "hook grip", or situations when obstacles occlude a clear path between the hand and the object.

7. Results

Depending on the complexity of the manipulation task, our algorithm generates 5-10 simulation frames per second on a single core of 2.8 GHz Intel Core 2 Duo processor. We solve the constrained optimizations using SNOPT [GSM96]. At each time step, the initial state of the optimization is set to the solution of the previous optimization. The simulation time step is $\frac{1}{30}$ second.

To explore the applicability of the proposed animating technique, we generated all our examples by only controlling the object without any specification on the hand. Our framework also allows the user to control the kinematic states of the hand although it is often unnecessary. We refer the reader to the accompanying video for a full demonstration.

We used a consistent set of parameters to generate the following examples. Table 7 shows the weights of the kinematic objectives. In all the examples, we set the weight of Q_M, G_F, and the initial weight of G_L to be 0.05, 10.0, and 100.0, respectively.

Different object shapes and grip styles. The same manipulation task and grip style can be applied to a wide range of objects. We specified a simple kinematic goal to describe a lifting task:$G_U(\mathbf{x}) = \mathbf{x}[1] - 0.2$. This objective directly controls the object's vertical position to be at 0.2 meter. We demonstrated that lifting a book and lifting a coffee cup could be handled by the same kinematic goal with the same default hand pose $\bar{\mathbf{q}}$. Since the object orientation was not a kinematic objective in this experiment, the book and cup ended up with different orientations due to the different contacts between the hand and the object.

External disturbances. The user can interact with the scene by applying forces on the hand or the object. By controlling the contact forces between the hand and the object, we can easily adjust the muscle responsiveness of the hand when an unexpected perturbation occurs (Figure 6). We set the attenuation rate of the weight of G_L to 75% for the fast muscle response and 50% for the slow muscle response. Depending on the goals of the manipulation, additional kinematic objectives might be needed. For example, when using lifting manipulation to handle a cup full of coffee, we needed to add one objective that kept the cup in the upright position to reduce spilling in the presence of unexpected disturbances.

Anticipation of the object. The programmer can create animation based on the expected mass and the surface friction of the object. This feature can potentially create many interesting and unpredictable scenarios for video games or other interactive applications. We applied the same lifting manipulation to three scenarios where the expected mass of the object was 50%, 100%, or 150% of the actual mass. When the hand underestimated the mass, the motion appeared wobbly and slow initially. When overestimating the mass, the hand applied lifting force larger than necessary to overcome the gravity, causing an abrupt initial acceleration followed by a sudden deceleration. Similarly, we can model the anticipation of the surface materials. We showed that when the hand tried to grab a wet bar of soap with an overestimated friction coefficient of the surface, the soap slipped out of the hand.

© The Eurographics Association 2008.

Table 1: *Kinematic objectives*

Example	Objectives	Weights
Book	$\mathbf{x}^{t+1}[1] - 0.2$	10
Cup	$\mathbf{x}^{t+1}[1] - 0.2$	10
	$\|\mathbf{R}^{t+1}(0\ 1\ 0)^T - (0\ 1\ 0)^T\|$	10
Gift box	$\mathbf{x}^{t+1} - (0\ 0.15\ 0)^T$	(1 10 10)
Soap	$\mathbf{x}^{t+1}[1] - 0.1$	10
Hammer	six objectives on $\mathbf{x}^{t+1}$ and $\mathbf{R}^{t+1}$	all 10
	$(\dot{\mathbf{x}}^{t+1} + \dot{\mathbf{R}}^{t+1}\mathbf{h}) - (0.2\ -0.3\ 0)$	$\frac{1}{30}$
Nail	six objectives on $\mathbf{x}^{t+1}$ and $\mathbf{R}^{t+1}$	all 10

With the correct estimation, the hand used initial grip forces that prevent slipping.

Two-hand manipulation. In many situations, a hand manipulation can be easily described in terms of the end goal of the object, rather than the configurations of the hand. For example, a hammering manipulation procedure (Figure 6) only requires an objective that specifies the position and the orientation of the hammer at the apex of the motion and another objective that controls the velocity of the hammer head when it hits the nail. We also set an objective to maintain the position and the orientation of the nail. Although the programmer did not specify anything about the hand, the hand behaved actively to secure the nail while responding passively to the impact of the hammer. Finally, we demonstrated wine pouring manipulation that consisted of three keyframes on the bottle and two keyframes on the glass (Figure 6).

8. Conclusion

We describe an interactive physics-based algorithm for synthesizing hand manipulation across a wide variety of tasks, objects, and stylistic preferences. Our framework decouples manipulation into a task description, an object description, and a grip style. Consequently, a small number of these basic components can be combined to create a variety of hand manipulations.

Our formulation allows the programmer to synthesize a manipulation task by describing simple kinematic goals in the domain of the object configuration. The animator only needs to specify the motion of the object and the algorithm will automatically produce the hand motion via coupled dynamic equations of motion. In all our examples, each task descriptions comprised at most five kinematic goals.

Our algorithm model dynamic response of the hand by controlling the grip force and the lift force at the contact point. Without modeling the anatomical details, many subtle movements of the hand cannot be captured by our algorithm. We suspect the motion realism can be improved by incorporating the joint compliances in our model. We are also interested in modeling the joint interdependency of the human hand. Such a hand representation can increase the physical realism and reduce the dimensionality of the optimization space.

We also describe a simple algorithm for generating the initial grasp and a collision-free reaching motion. Although our algorithm is generic across a wide range of objects, we significantly simplify the problem of grasp planning addressed in the robotics literature. The algorithm focuses on synthesizing aesthetically pleasing grasp motion, rather than real-world considerations such as robustness and stability. One major limitation of our algorithm is that the initial grasp optimization is sensitive to the initial relative position of the hand and the object. Consequently, this naive optimization only yields a local optimal initial grasp.

Our algorithm does not handle rolling contacts that occur frequently in everyday hand manipulation. The current geometric representation of the hand is overly simplified to model realistic rolling contact. One possible avenue of future work is to build a more sophisticated model that includes muscles and skin.

Acknowledgments The author would like to thank Clint Hidinger and Michael Su for help with the modeling and rendering. This work was supported by NSF grant CCF-0742302 and Georgia Tech GVU center.

References

[ABM82] ABEND W., BIZZI E., MORASSO P.: Human arm trajectory formation. *Brain 105* (1982), 331–348.

[AH89] ATKESON C., HOLLERBACH J.: Kinematic features of unrestrained vertical arm movements. *Journal of Neuroscience 5* (Sept. 1989), 2318–2330.

[AHS03] ALBRECHT I., HABER J., SEIDEL H.-P.: Construction and animation of anatomically based human hand models. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA-03)* (Aire-la-Ville, July 2003), Breen D., Lin M., (Eds.), Eurographics Association, pp. 98–109.

[AN99] AYDIN Y., NAKAJIMA M.: Database guided computer animation of human grasping using forward and inverse kinematics. *Computers and Graphics 23*, 1 (1999), 145–154.

[BHM96] BUSS M., HASHIMOTO H., MOORE J. B.: Dextrous hand grasping force optimization. *IEEE Transations on Robotics and Automation 12* (June 1996), 406–418.

[Bir94] BIRYUKOVA E.: *A Model of Hand Dynamics. In Advances in the Biomechanics of Hand and Wrist*. Plenum Press, 1994.

[BK00] BICCHI A., KUMAR V.: Robotic grasping and contact: A review. In *ICRA* (2000), vol. 1, pp. 348–353.

[BLTK93] BEKEY G. A., LIU H., TOMOVIC R., KARPLUS W. J.: Knowledge-based control of grasping

© The Eurographics Association 2008.

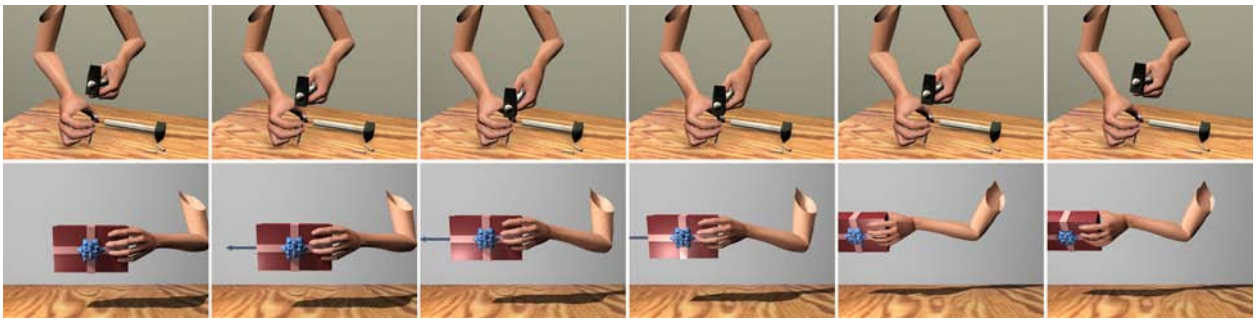

Figure 6: *Top: Wine pouring manipulation. Created by three kinematic goals on the bottle and two on the glass. Middle: Hammering manipulation. Created by an objective that specifies the position and the orientation of the hammer at apex of the motion and another objective that controls the velocity of the hammer head when it hits the nail. Bottom: User intervention.*

in robot hands using heuristics from human motor skills. *IEEE Transactions on Robotics and Automation 9*, 6 (Dec. 1993), 709–721.

[BW92] BARAFF D., WITKIN A.: Dynamic simulation of non-penetrating flexible bodies. In *SIGGRAPH* (July 1992), vol. 26, pp. 303–308.

[ES03] ELKOURA G., SINGH K.: Handrix: Animating the human hand. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA-03)* (Aire-la-Ville, July 2003), Breen D., Lin M., (Eds.), Eurographics Association, pp. 110–119.

[GSM96] GILL P., SAUNDERS M., MURRAY W.: *SNOPT: An SQP Algorithm for Large-scale Constrained Optimization*. Tech. Rep. NA 96-2, University of California, San Diego, 1996.

[HBT95] HUANG Z., BOULIC R., THALMANN D.: A multi-sensor approach for grasping and 3-D interaction. In *Computer Graphics International '95* (June 1995).

[HH97] HAJIAN A. Z., HOWE R.: Identification of the mechanical impedance at the human finger tip. *Journal of Biomechaniclal Engineering 119*, 1 (1997), 109–114.

[Ibe97] IBERALL T.: Human prehension and dextrous robot hands. *International Journal of Robotics Researh 16*, 3 (June 1997).

[Kaw99] KAWATO M.: Internal models for motor control and trajectory planning. In *Current Opinions in Neurobiology* (1999), vol. 9.

[KCMT00] KIM J., CORDIER F., MAGNENAT-THALMANN N.: Neural network-based violinistt's hand animation. In *Proceedings of the Conference on Computer Graphics International (CGI-00)* (Los Alamitos, CA, June 2000), IEEE, pp. 37–44.

[KKKL94] KOGA Y., KONDO K., KUFFNER J., LATOMBE J.-C.: Planning motions with intentions. In *SIGGRAPH* (July 1994), pp. 395–408.

[KL00] KUFFNER J., LATOMBE J.-C.: Interactive manipulation planning for animated characters. In *Pacific Graphics* (Oct. 2000).

[Kog94] KOGA Y.: *On Computing Multi-arm Manipulation Trajectories*. PhD thesis, Stanford University, Feb. 1994.

[KP06] KRY P. G., PAI D. K.: Interaction capture and synthesis. *ACM Trans. on Graphics (SIGGRAPH) 25* (Aug. 2006), 872–880.

[LHP05] LIU C. K., HERTZMANN A., POPOVIĆ Z.: Learning physics-based motion style with nonlinear inverse optimization. *ACM Trans. on Graphics (SIGGRAPH) 24*, 3 (July 2005), 1071–1081.

[PZ05] POLLARD N. S., ZORDAN V. B.: Physically based grasping control from example. In *ACM SIGGRAPH /Eurographics Symposium on Computer Animation* (2005), Eurographics Association, pp. 311–318.

[Reu63] REULEAUX F.: *Kinematics of Machinery*. Dover, 1963.

[Sta91] STANSFIELD S.: Robotic grasping of unknown objects. *The International Journal of Robotics Research 10*, 4 (Aug. 1991), 314–326.

[TSF05] TSANG W., SINGH K., FIUME E.: Helping hand: An anatomically accurate inverse dynamics solution for unconstrained hand motion. In *Eurographics/SIGGRAPH Symposium on Computer Animation* (2005), pp. 319–328.

[UKS89] UNO Y., KAWATO M., SUZUKI R.: Minimum muscle-tension-change model which reproduces human arm movement. In *Symposium on Biological and Physiological Engineering* (1989), pp. 299–302.

[VI90] VENKARAMAN S. T., IBERALL T.: *Dextrous Robot Hands*. New York: Springer-Veriag, 1990.

[YKH04] YAMANE K., KUFFNER J. J., HODGINS J. K.: Synthesizing animations of human manipulation tasks. *ACM Trans. Graph 23*, 3 (2004), 532–539.

© The Eurographics Association 2008.