Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2012)
P. Kry and J. Lee (Editors)

Contact-Invariant Optimization for Hand Manipulation

Igor Mordatch Zoran Popović Emanuel Todorov

University of Washington

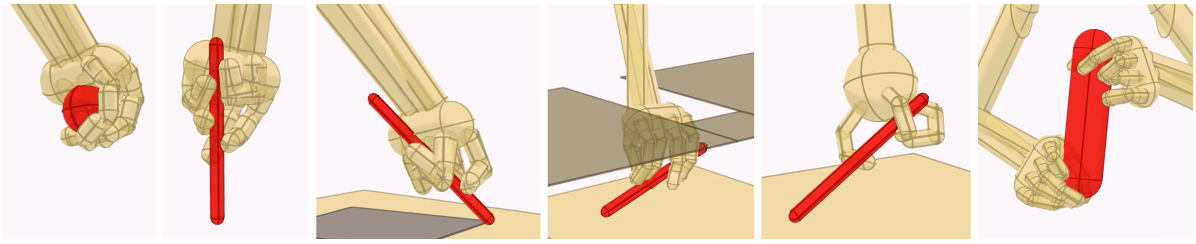

Figure 1: *A selection of grasps and motions synthesized by our method.*

Abstract

We present a method for automatic synthesis of dexterous hand movements, given only high-level goals specifying what should happen to the object being manipulated. Results are presented on a wide range of tasks including grasping and picking up objects, twirling them between the fingers, tossing and catching, drawing. This work is an extension of the recent contact-invariant optimization (CIO) method, which introduced auxiliary decision variables directly specifying when and where contacts should occur, and optimized these variables together with the movement trajectory. Our contribution here is extending the unique contact model used in CIO which was specific to locomotion tasks, as well as applying the extended method systematically to hand manipulation.

Categories and Subject Descriptors (according to ACM CCS): I.3.1 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Hand manipulation involves some of the most complex and fascinating biological movements, and is of interest in graphics and animation as well as robotics, biomechanics and neuroscience. Despite its importance in multiple fields, current understanding of hand manipulation lags behind other behaviors such as locomotion. This likely reflects the intrinsic difficulty of the problem. Indeed it is notable that most animal species are capable of impressive locomotion, while only humans can perform full-blown manipulation, and appear to do so using specialized neural pathways that partially bypass the spinal circuits [RS09]. From a computational perspective hand manipulation is a difficult problem not only because of the large number of degrees of freedom (DOFs) of the hand and under-actuation of the object being manipulated, but perhaps more importantly, because of the contact interactions involved. Planning and control in the presence of contact discontinuities are of course always challenging. Nevertheless hand manipulation is unique in that a large number of contacts can appear and disappear in rapid succession, and dynamic phenomena such as rolling and sliding can be actively used. This makes it virtually impossible to synthesize realistic hand manipulation by prespecifying contact events – as is often done in locomotion for example.

We aim to automate the synthesis of hand manipulation. While approaches based on motion capture or scripting have played an important role and will continue to do so, it is obviously beneficial to have methods that are more automated in the sense that they require less detailed input. Such automation is particularly important in hand manipulation where motion capture is technically challenging: optical sys-

tems suffer from occlusions and marker mis-identifications when tracking a large number of markers in a small volume, while data gloves impede natural movements and also limit tactile sensation – which in turn is essential for dexterous manipulation in humans [JF09].

Here we develop a synthesis method whose input is at the highest level of specification possible. In the task of picking up an object for example, we merely specify the desired final position of the object. The hand and finger movements and finger-object interactions are then computed automatically. Similarly, twirling a pen between the fingers is specified as a desired angular velocity for the pen; drawing is specified as a desired trajectory for the tip of the pen, etc. This high-level specification is not treated as a hard constraint (since it may be infeasible) but as a cost which is mixed with other costs enforcing physical realism and shaping the optimization landscape.

The present work is an extension of the recent contact-invariant optimization (CIO) method which was able to synthesize full-body motions including getting up, walking, climbing, kicking, carrying objects, hand stands and cooperative actions [MTP12]. Our original contribution in this paper is twofold. First, we modify and at the same time simplify the unique contact model used in CIO – which turned out to be inadequate for hand manipulation because it assumed contacts between parallel surface patches and discouraged rolling and sliding. Second, we apply the modified CIO method to a wide range of hand manipulation tasks as well as different hand morphologies, and obtain motions whose complexity exceeds prior results obtained automatically. Key to the success of our method is the special handling of contact dynamics.

2. Related Work

2.1. Planning through contacts

The power of the CIO method (both original and modified) lies in its ability to optimize movement trajectories through multiple contact events, and do so fully automatically, without using motion capture data. This type of optimization is desirable, but apparently is so hard to achieve in practice that results along these lines have appeared only recently. Note that the present discussion is specific to trajectory optimization – as opposed to optimization of feedback controllers which have dealt with contacts automatically [Sim94, Ijs01] but have other limitations including computationally-intensive optimization, poor local minima, and the need to guess a suitable controller parameterization. Focusing on trajectory optimization, [TT10] handled contacts automatically by introducing a stochastic complementarity formulation that yields smooth yet phenomenologically hard contacts. These authors then optimized the sequence of controls in forward dynamics for an object spinning task. An alternative method for contact smoothing applicable to inverse dynamics (and thus space-time optimization) was developed in [Tod11]. It is also possible (and actually easier) to define inverse contact dynamics with respect to spring-damper models [BSF09], but so far this has been applied to motion capture data and not to planning through contacts.

The original CIO method also used an inverse dynamics approach: it optimized the trajectory (represented with splines), while the contact forces along with the controls were defined implicitly via inverse dynamics. The modified CIO method developed here takes a different approach. Instead of obtaining the contact forces from either forward or inverse dynamics, these forces are now treated as independent variables and are optimized jointly with the trajectory. Their physical realism (i.e. complementarity and friction-cone constraints) is enforced using soft costs. The idea of treating contact forces as independent optimization variables seems to have originated in the context of tracking recorded movements [MLPP09] and single timestep optimization [SJL09]. It was later extended to automatic synthesis of entire trajectories [PT12].

Another approach that synthesized finger-object contacts automatically, given only the desired motion of the object and an initial grasping pose, is [Liu08, Liu09]. This is a hierarchical scheme where contact forces are computed on the high level while corresponding joint torques are computed on the low level. The high level did not anticipate the making and breaking of contacts, but instead planned forces at existing contacts so as to achieve the desired object accelerations. Rich yet unanticipated contact dynamics were then obtained in forward simulation. The same group later developed a more systematic method for generating contact events, based on careful sampling [YL12]. The object and wrist trajectories were measured using motion capture, and finger trajectories compatible with the measurements were then computed automatically. The results were quite remarkable. However this method is sensitive to the specifics of the motion capture data, e.g. having irregular object velocity consistent with the underlying finger gait. Unlike methods using soft costs (which can smooth out measurement imperfections), the sampling method uses a hard criterion for compatibility with the recorded motion and makes binary decisions based on this criterion, thus it is not clear how its sensitivity can be reduced. Work in robotics [BELK09, JXL10] has also used sampling-based methods to plan trajectories in the presence of contacts, albeit in a quasi-static setting which does not give rise to the rich dynamic interactions observed in [YL12].

2.2. Motion capture

In graphics and animation, the most common approach to hand manipulation has been to rely on motion capture data to various degrees. Collecting such data is not easy as we already mentioned, but the lack of automated methods has

left animators with no other choice. Among the more popular methods in this class are [YKH04, PZ05, SH07]. We will not discuss the details here (except for the issue of dimensionality reduction – see below) since our work is quite different. An up-to-date discussion of hand manipulation based on motion capture can be found in [YL12].

While the key advantage of our method is that it does not need motion capture data, it can nevertheless be adapted to take advantage of available data. All we have to do is augment or replace the abstract task specification with one that encourages staying close to the data.

2.3. Dimensionality reduction

While biological systems have many DOFs, the movement behaviors observed in daily life span a small fraction of the space that is potentially accessible. This is due to a combination of biomechanical and neural factors which have been debated for decades [TJ09, KVC12]. Every researcher who has looked for dimensionality reduction in movement data appears to have found it - at the level of kinematics [SFS98], instantaneous muscle activity [TSB99], spatio-temporal patterns of muscle activity [dSB03], organization of feedback control laws [LT07]. This may be why it is possible to interpolate in motion capture databases and obtain plausible movements. Without reduced dimensionality, the volume of space that needs to be filled with data would be so large that interpolation would be unlikely to work.

With regard to hand manipulation, it was shown [SFS98] that the hand poses used to grasp arbitrary objects lie in a surprisingly low-dimensional space: the first 4 principal components accounted for over 95% of the variance in the data. In more complex manipulation tasks the effective dimensionality can increase from 4 to 10, but is still half the number of recorded DOFs [TG04]. This low-dimensional structure has been exploited in robotics to speed up the search for viable grasping poses [CA09]. In its present form our method does not exploit low-dimensional structure, and we believe this is one of its limitations – see Future Work.

2.4. Automated grasping

Grasping has received more attention than any other type of hand movement, and has been automated through a variety of methods (mostly in robotics) that do not rely on motion capture. These methods are not applicable to dexterous object manipulation, but nevertheless are based on important ideas. One such idea is force closure – which is in some sense the equivalent of locomotion stability criteria such as ZMP. More generally, researchers have considered a variety of grasp quality metrics, and optimized the grasp pose with respect to them [MA99, MM04]. See also the review papers [OSC00, BK00].

A more intuitive yet robust approach to grasp synthesis is to position and pre-shape the hand near the object, and then simply close the fingers, relying on their compliance to curl around the object [CH90]. In this approach one does not need to model the hand-object interactions in detail. The quality of an initial pose can be defined as the grasp quality of the final pose resulting from this procedure. Even if such heuristics can work, it is not clear how often humans actually use them. Indeed it was recently shown that the nervous system predicts finger-object contacts around 100 msec before they occur, and begins to adjust muscle activity so as to transition from motion to force [MF08].

In terms of numerical optimization, it is notable that most methods for automated grasp synthesis have relied on extensive sampling or restarts – presumably because the search space has many poor local minima. There are exceptions such as [CG96] where grasps were planned using residual minimization. One surprising feature of our method is that local minima are not a problem (see below), most likely because we have formulated cost functions that result in easier-to-navigate landscapes.

3. Review of Contact-Invariant Optimization (CIO)

CIO is a trajectory optimization method used for character animation that jointly synthesizes motion as well as contact events from only few high-level goals. The motion is broken up into a sequence of motion phases with each phase having a constant contact state. The key idea of CIO is to introduce real-valued auxiliary variables c_j in the optimization that control whether or not a body region j is making contact with the environment. In the case of humanoid character models, a contact variable was associated with a surface region of each hand and foot. These variables enforce a trade-off as follows. When c_j for limb j has high value (active), the limb can generate a contact force anywhere in the predefined contact region, but that limb must be touching the ground for the duration of the motion phase. When a contact variable has low value (inactive), the limb is free to move in space, but it cannot generate any contact forces. An inverse dynamics quadratic program (QP) was used to recover forces at various points along the trajectory, and the trajectory was optimized such that positions, forces, controls, and contact variables were all consistent with each other.

3.1. CIO for Hand Manipulation

The original CIO method assumes that contact forces originate in rectangular patches on the body (such as soles of feet and palms of hand). This is necessary in order for contact force regions to be expressed as linear constraints in an inverse dynamics QP, but also means that curved contact region geometry cannot be supported. While not too restrictive for humanoid motion, this assumption is problematic when applying CIO to hand manipulation, because the surfaces of the fingers and palm are curved and hand motion often takes

advantage of this (e.g. when rolling objects on the side of the finger).

To remove this limitation, in this work we include contact forces and their origin points as extra variables in the trajectory optimization problem. We can then impose force origins to lie on arbitrarily-shaped contact regions through soft constraints. Friction cone force constraints can also be enforced on exact cone geometry, rather than pyramidal approximations of the cone that are typically used. This reformulation removes the need to set up and use a quadratic programming solver to calculate the forces, which makes individual objective function evaluations much faster and makes the method easier to implement. The downside is that the size of the optimization problem becomes significantly larger. Nonetheless, we have found this approach to have optimization times similar to those reported in the original CIO work. Additionally, the gradient of the objective function (described in section 5) wrt the contact forces and their origins can be calculated analytically and does not require expensive finite difference calculation (although we do not take advantage of this in our present implementation). Note that our modified approach is still different from previous work that optimizes only contact forces, because we also optimize auxiliary contact variables. We believe these scalar contact variables provide compact coordination between higher-dimensional position and force variables.

We now describe our CIO-based method specific to hand manipulation.

4. State and Trajectory

Given an initial state $\mathbf{s}_0$ of the entire system and a high-level goal (such as terminal object location), our method computes a state trajectory $\mathbf{s}(t)$ for a fixed period of time $[0,T]$. For simplicity, consider a system with one hand H and one object O to be manipulated. The state of the hand at any point in time is described as:

$$\mathbf{s}_H = \left[\begin{array}{cccc} \mathbf{x}_H & \dot{\mathbf{x}}_H & \mathbf{p}_i & \dot{\mathbf{p}}_i \end{array} \right] \quad (1)$$

Where $\mathbf{x}_H \in R^6$ are the position and orientation of the palm, and $\mathbf{p}_i$ is Cartesian fingertip location of finger $i \in \{1...N_{\text{fingers}}\}$. Orientation is expressed with the exponential map [Gra98] because of its suitability in trajectory optimization. From fingertip and palm locations, we can recover finger and arm joint angles and transforms using inverse kinematics (which in our case can be solved analytically). We assume the shoulder is fixed. The state of the object O at any point in time is described as:

$$\mathbf{s}_O = \left[\begin{array}{ccccc} \mathbf{x}_O & \dot{\mathbf{x}}_O & \mathbf{f}_j & \mathbf{r}_j^O & c_j \end{array} \right] \quad (2)$$

Where $\mathbf{x}_O$ are the position and orientation of the object. $\mathbf{f}_j$ and $\mathbf{r}_j^O$ and the force direction and origin point for contact $j \in \{1,...,N_{\text{contacts}}\}$. $\mathbf{r}_j^O$ is expressed in the object's local coordinate system because simple trajectories in object space might trace complex trajectories in world space. Let $\mathbf{r}_j$ be the corresponding derived world space origin point. $c_j \in [0,1]$ is an auxiliary variable that specifies if contact j is active. In contrast with the original CIO formulation where c_j were unbounded, here we treat c_j informally as the probability of being in contact.

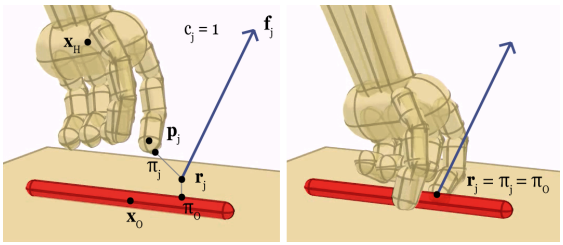

Figure 2: Scene Model. *The quantities used in hand and object state. The left state is infeasible because contact c_j is active, but force origin $\mathbf{r}_j$ do not coincide with the hand and object. The state on the right is feasible.*

N_{contacts} is a fixed number of possible contacting surfaces that can affect the object. The surfaces we consider are:

- N_{fingers} contacts with the fingers, anywhere on the surface of the distal segments;
- one contact anywhere on the palm surface for each hand;
- two contacts anywhere on the ground surface.

Thus, $N_{\text{contacts}} = N_{\text{hands}}(N_{\text{fingers}}+1)+2$. See figure 2 for a visual description of the relevant quantities.

The entire system state is $\mathbf{s} = \left[\begin{array}{cc} \mathbf{s}_H & \mathbf{s}_O \end{array} \right]$ and the optimization problem variables are:

$$\mathbf{S} = \left[\begin{array}{c} \mathbf{s}_k \end{array} \right] \quad (3)$$

Where $k \in \{1,...,N_{\text{keyframes}}\}$ are the discrete indices for the sequence of key states being optimized. The initial state $\mathbf{s}_0$ is specified by the user and is not part of the optimization variables. A motion phase is a segment between any two key states $\mathbf{s}_k$ and $\mathbf{s}_{k+1}$. Motion trajectory $\mathbf{s}(t)$ is formed by interpolating quantities of these key states, spaced apart every 0.5sec in time. We use cubic spline interpolation for quantities $\mathbf{x}$, $\mathbf{p}$ (using $\dot{\mathbf{x}}$ and $\dot{\mathbf{p}}$ as spline tangents), linear interpolation for $\mathbf{f}$, $\mathbf{r}^O$, and constant interpolation for c. These choices of interpolation are motivated by how much complexity we expect in the trajectories for these quantities. Kinematic positions and orientations need complex spline trajectories (and thus need to include their velocities as optimization variables). Forces are interpolated linearly, while the auxiliary contact variables c are constant within each phase, by their original definition.

5. Objective Cost Terms

5.1. Contact-Invariant Cost

When contact j is active, the object and contact surface j must be coincident with each other, and the contact force origin $\mathbf{r}_j$ must lie on this coincident surface. In the pesent verion of the method we also restrict ourselves to motions with no slipping, so relative velocity at $\mathbf{r}_j$ must be zero. These constraints are satisfied when the following residual is minimized:

$$L_{\text{CI}}(\mathbf{s}) = \sum_j c_j(|\mathbf{e}_j^O|^2 + |\dot{\mathbf{e}}_j^O|^2 + |\mathbf{e}_j^H|^2 + |\dot{\mathbf{e}}_j^H|^2) \quad (4)$$

$$\mathbf{e}_j^O = \pi_O(\mathbf{r}_j) - \mathbf{r}_j \quad (5)$$

$$\mathbf{e}_j^H = \pi_j(\mathbf{r}_j) - \mathbf{r}_j \quad (6)$$

Where $\pi_O(\mathbf{r})$ and $\pi_j(\mathbf{r})$ are projections of $\mathbf{r}$ on the surface of the object and of contact body j, respectively. We use capsules for all our contact bodies, and as a result the projections are continuous functions of $\mathbf{r}$ and are fast to compute.

Note that this formulation is able to support arbitrary contact region geometry. For any point on the contact region, the above constraint will hold. For example, in the case of rectangular foot contacting the ground, any $\mathbf{r}_j$ inside the contact rectangle will set (4) to zero. However the present approach is not limited to rectangular patches.

5.2. Physics Violation Cost

The hand H is connected to a fixed shoulder base, which makes the hand system fully actuated. Thus we assume that all kinematically feasible poses are also dynamically feasible, and place no dynamics constraints on the motion of the hand.

Since the object O is unactuated, the only way it can move is through application of contact forces. At any point in time, we place traditional Newton-Euler constraints on the motion of the object with the following residual:

$$L_{\text{physics}}(\mathbf{s}) = ||\mathbf{f}_{\text{tot}} - \dot{\mathbf{P}}||^2 + ||\mathbf{m}_{\text{tot}} - \dot{\mathbf{L}}||^2 + \lambda \sum_j ||\mathbf{f}_j||^2 \quad (7)$$

$$\mathbf{f}_{\text{tot}} = \sum_j c_j \mathbf{f}_j + \mathbf{f}_{\text{ext}} \quad (8)$$

$$\mathbf{m}_{\text{tot}} = \sum_j c_j \mathbf{f}_j \times (\mathbf{r}_j - \mathbf{x}_O) + \mathbf{m}_{\text{ext}} \quad (9)$$

Where $\dot{\mathbf{P}}(\mathbf{x}_O, \ddot{\mathbf{x}}_O)$ and $\dot{\mathbf{L}}(\mathbf{x}_O, \ddot{\mathbf{x}}_O)$ are change in linear and angular momentum of the object. $\mathbf{f}_{\text{ext}}$ and $\mathbf{m}_{\text{ext}}$ are any external forces and moments applied to the object including gravity. λ is a force regularization constant that discourages the use of very large individual forces and in turn encourages distributing the total force across multiple contacts.

Note that while $\mathbf{f}_j$ follows a continuous trajectory, we are able to model contact force discontinuities using the product $c_j\mathbf{f}_j$ because c_j is piecewise constant.

Additionally, we must constrain $\mathbf{f}_j$ to lie in the friction cone of the contact surface j. We use the body's contact surface normal at $\pi_j(\mathbf{r}_j)$ to define the cone normal $\mathbf{n}_j$. Then the angle between $\mathbf{f}_j$ and $\mathbf{n}_j$ must be less than the coefficient of friction μ. This is imposed as a half-quadratic soft cost.

$$L_{\text{cone}}(\mathbf{s}) = \sum_j \max(\text{acos}(\mathbf{f}_j \cdot \mathbf{n}_j) - \tan^{-1}\mu, 0)^2 \quad (10)$$

5.3. Kinematic Violation and Task Cost

We place several kinematic constraints on our hand configuration. First, realistic limits on the finger and arm joint angles are imposed (recall that joint angles are available from analytical inverse kinematics). Second, because fingertip locations are independent optimization variables, their distance from the finger base must be restricted not to exceed the maximum length of the finger. Lastly, all collisions are penalized. Because all geometry consists of capsules, we can efficiently calculate and enforce a minimum distance between capsules. The three above constraints are imposed as soft half-quadratic costs and added up to form the cost term $L_{\text{kinematic}}$.

Humans prefer to use the bottom pads of their fingers for grasping objects. This can be due to the higher friction on the skin of that area, as well as softer tissue which can squash to create larger contact regions. Since we currently do not model these factors, we simply add an additional cost term L_{pad} which encourages $\mathbf{r}_j$ for a finger contact to be close to the distal bottom pad of the finger j (with derived transform matrix $\mathbf{x}_j$).

$$L_{\text{pad}}(\mathbf{s}) = \sum_{j \in \{1,\ldots,N_{\text{fingers}}\}} ||(\mathbf{x}_j^{-1}(\mathbf{s})\mathbf{r}_j)^{(y)}||^2 \quad (11)$$

The task cost is a set of simple high-level goals ℓ_b that are behavior-specific and are provided by the user of our method. Each task b is constructed by selecting features of interest $\mathbf{h}_b$ and providing their target values $\mathbf{h}_{b,t}^\star$ at certain times $t \in T_b$. The task cost then penalizes the difference between the actual and target feature values. Additionally, we place a small preference on smooth motion by penalizing hand and object accelerations.

$$L_{\text{task}}(\mathbf{s}, t) = \sum_b \ell_b(\mathbf{s}, t) + \lambda(||\ddot{\mathbf{x}}_O||^2 + ||\ddot{\mathbf{x}}_H||^2) \quad (12)$$

$$\ell_b(\mathbf{s}, t) = I[t \in T_b] \left\|\mathbf{h}_b(\mathbf{s}) - \mathbf{h}_{b,t}^\star\right\|^2 \quad (13)$$

Where $I[\cdot]$ is an indicator function for t matching one of the times when the goal is specified. For example, to create a task ℓ_{location} that commands the object location at the end of the movement, set $T_b = \{T\}$, $\mathbf{h}_b(\mathbf{s}) = \mathbf{x}_O$ and $\mathbf{h}_{b,T}^\star$ to the desired object position and orientation.

The typical weights on the costs we used are summarized in table 1. We have not found the algorithm to be particularly sensitive to the choice of these weights.

Table 1: *Cost Term Weights*

Cost Term	Weight	Cost Term	Weight
CI	10^1	task	$10^{-1 \text{ to } +1}$
physics/cone	10^0	pad	10^{-3}
kinematic	10^0	λ	10^{-3}

6. Numerical Optimization

We solve a box-constrained optimization problem to obtain $\mathbf{S}^\star$ described in (3), from which we can then obtain a continuous motion trajectory as described in section 4. The optimization problem is:

$$\mathbf{S}^\star = \begin{array}{l} \arg\min_{\mathbf{S}} \sum_t \sum_i L_i(\mathbf{s}(t)) \\ c_j \in [0,1] \end{array} \quad (14)$$

Where i refers to the cost terms described in the previous section and t is a time index that subsamples the entire trajectory (in our case, we place t every 0.05sec of the trajectory). Thus, even though we optimize the motion at a very coarse time scale (key states every 0.5sec), we check for dynamic and kinematic validity at a fine scale.

We use a standard off-the-shelf box-constrained LBFGS algorithm [BLNZ95] to solve (14). We use between 10 and 20 motion phases, depending on complexity of the task. The dimensionality of the problem is $N_{\text{keyframes}}(N_{\text{hands}}(12+6N_{\text{fingers}})+N_{\text{objects}}(12+7N_{\text{contacts}}))$ which is from 1030 to 2060 in typical scenarios. Gradients of the objective function are computed with finite differencing (we use $\varepsilon = 10^{-3}$). Our implementation of finite differences takes advantage of the fact that variables affect the cost terms only at a few points in time and the costs for other times do not need to be recomputed.

The optimization is initialized with a static trajectory obtained by repeating the initial state for all times. This removes the need for hand-crafted initialization guesses specific to each motion. This initial trajectory is then perturbed with zero-mean Gaussian noise (with variance 10^{-2}) to break any potential symmetries. All contact variables c_j are initialized to 0.5 (which is the middle of their allowed range). For each example shown we run the optimization only once, i.e. there is no need to use a large number of restarts and eliminate all but a few of the solutions. The fact that we obtain good results in all cases suggests that even if there are local minima in this problem (which is very likely), most local minima correspond to successful motions.

We perform numerical optimization using continuation, in two stages. In the first stage, L_{physics} and L_{CI} are downweighted to 0.1 of their typical values, which allows the optimization to explore potentially unphysical trajectories. In the second stage, all weights are at their weights specified in table 1. The solution at the end of the first stage is used to initialize the second stage. Despite the large problem size, we obtain optimization times of 2 to 6 minutes depending on the problem complexity on a quad-core 2.4Ghz machine. The optimizer executed between 500 and 1500 iterations in each stage before reaching a local minimum.

7. Results

By using simple task specifications ℓ_b or changing the environment, we can generate a wide range of hand/object interactions.

Grasping By using only ℓ_{location} (described in section 5.3) which specifies the terminal location of the object, our method automatically synthesizes entire hand and object trajectories. The synthesized grasps vary depending on object shape, e.g. a pen-like object leads to a different grasp compared to a ball-like object.

The type of grasp also varies depending on the specified final orientation. For horizontal orientation, the hand simply grips the object. For vertical orientation, a more complex grasp between fingers occurs in order not to twist the hand too much (and thus violate the joint limits). The final orientation can also be left unspecified. When the environment has obstacles, the hand will manipulate the orientation of the object to avoid the obstacles.

The resulting grasps are robust to external force perturbations. We show this by applying random forces of 0.25 Newtons at the object's tip in the second half of the trajectory.

Cyclic In-Hand Manipulation By connecting the last motion phase to the first phase, we can generate cyclic motions without any pre-specified initial state. By removing the ground environment (and its contacts), the object must be dexterously manipulated using only the hand. To synthesize cyclic object twirling motions, we use a task that commands a target orientation trajectory for the object and constant position for the hand. The target orientation trajectory is simply a full constant-velocity rotation about one of the standard axes. The weight on L_{task} cost is low (10^{-1}) so the hand and object can deviate from this trajectory and form a natural gait.

Common Manipulation Tasks We generated a small sample of common hand manipulation tasks, such as drawing and dialing a cellphone. For the task of drawing a square, we specified that the tip of the pen object must reach each of the four corners of the square at every other motion phase. The need for the hand to pick up the pen was not specified and emerged automatically. Due to interaction of the pen with the the arm, the drawing task formed interesting trajectories that were not simple straight lines.

For the dialing task, constant object position and orientation were specified for the second half of the trajectory, and at every other motion phase the thumb fingertip location was specified to coincide with one of the buttons on the phone object.

Non-Humanoid Hand Morphologies The method is able to generalize to different hand morphologies, such as a three-finger hand similar to the Barrett Hand robot. We demonstrate a range of grasps and cyclic manipulation on this modified model. While the movement strategies are now quite different from a human hand, the high-level tasks are still accomplished.

Two-Handed Interaction The method is also able to generalize to a varying number of hands, with the hands cooperating to achieve the tasks and to evenly distribute the load (which is because of the force penalty term in (9)). We demonstrate grasping motions as well as a two-handed cyclic juggling motion. For juggling, a new ℓ_{contact} task was introduced that allowed only the contact variables c for the first hand to be active in the first half of the trajectory, and only the c for the second hand to be active in the second half of the trajectory. For one motion phase in the middle, the c's for both hands were penalized, restricting the object to be in the air for that phase.

8. Conclusions and Future Work

In this paper, we presented a method that can synthesize a wide variety of dexterous hand manipulation motions from only a few high-level goals on the object or hand, without any motion capture or detailed reference trajectory data. Our method is not specific to human hands and can generalize to other morphologies. This was achieved by adapting the Contact-Invariant Optimization (CIO) method to hand manipulation, and exploiting its ability to perform continuous optimization through contact events – which are numerous and exhibit complex structure in hand manipulation.

We were able to lift the previous restriction in the CIO method of contact regions being rectangular patches, and instead allowed contacts on curved surfaces. Another modification we made was to explicitly include contact forces and their origins as optimization variables. Note that these two changes are actually orthogonal. The first one is needed to apply CIO to hand manipulation. The second one was implemented because our brief preliminary testing indicated that it speeds up convergence, however we have not yet performed a systematic comparison. In other words, we do not yet know if it is better to optimize contact forces and their origins directly (while imposing physical realism with soft costs) or define them implicitly via inverse dynamics. In the latter case physical realism is automatically imposed and the number of variables being optimized is smaller, however the amount of computation per iteration increases (because we need an inverse contact solver) and furthermore an opportunity for continuation is lost. This comparison is a topic for future work.

Instead of using inverse contact dynamics, the new formulation only requires a projection function on the contact body. Although we only used single-capsule bodies, one could project onto more complex surfaces, or multi-capsule geometry. This would no longer tie contact to a single body, but instead could provide a floating "pool" of contacts to apply.

However, some restrictions present in the original CIO method still remain. By penalizing any relative velocity between hand and object (which prevents slipping contact), we cannot synthesize motions that purposefully slide the fingers along the object, or exploit slippage. Also, by using a coarse and smooth spline trajectory representation, we sometimes have difficulty with sharply-changing motions such as hard collisions or separation from the ground. We believe this is a large factor of why our object motion may sometimes look "floaty" or exhibit minor penetration. Either allowing discontinuities at spline knots, or having a second optimization at a finer timestep (initialized from the coarse result) may fix these issues. We also use a fixed number of phases and fixed phase durations as in the original CIO method and have not addressed adaptively changing these parameters.

Additionally, the dynamics and inertial effects of the hand are not taken into account in our method. Because the hand was not moving fast in our examples, we have not found this to be a problem, but modelling hand dynamics may be necessary in other cases.

8.1. Biomechanical and neural constraints

Some of the motions we synthesizes were too rich, in the sense that the simulated hand moved in ways that a human hand could not. This is much less of an issue in synthesizing full-body movements, because even though most skeletal muscles act on multiple joints, the number of muscles and their attachment is such that the joints can be controlled largely independently. In contrast, hand muscles act through a complex network of tendons and the muscle-to-DOF ratio is smaller, introducing substantial coupling.

We should be able to increase the realism of our results by incorporating such constraints. This could be achieved using very detailed hand models [SKP08] however it remains to be seen if such models are amenable to trajectory optimization. A different approach is to apply random forces to the tendons of simulated or physical hands – such as the Anatomically Correct Testbed robot hand [WRWM04] or cadaver hands [KVC12] – and measure the correlations in the resulting movements. This statistical information can then be taken into account via additional cost terms. Similar information can of course be obtained from motion capture data, however it is likely to reflect not only biomechanical but also neural constraints – which may or may not be desirable depending on what we aim to accomplish. To avoid going back to motion capture databases that specify the movement details, one could extract the correlations common to a wide range of movement tasks, including basic tasks such as attemtping to move one joint at a time [TG04].

Acknowledgments

We thank Tom Erez and Yuval Tassa for inspiring technical discussions. This research was supported in part by NSERC, NSF, NIH grants.

References

[BELK09] BOUYARMANE K., ESCANDE A., LAMIRAUX F., KHEDDAR A.: Potential field guide for humanoid multicontacts acyclic motion planning. *International Conference on Robotics and Automation.* (2009).

[BK00] BICCHI A., KUMAR V.: Robotic grasping and contact: A review. *International Conference on Robotics and Automation* (2000), 348–353.

[BLNZ95] BYRD R. H., LU P., NOCEDAL J., ZHU C.: A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput. 16*, 5 (1995).

[BSF09] BRUBAKER M. A., SIGAL L., FLEET D. J.: Estimating contact dynamics. In *ICCV* (2009), pp. 2389–2396.

[CA09] CIOCARLIE M. T., ALLEN P. K.: Hand posture subspaces for dexterous robotic grasping. *International Journal Robotics Research 28*, 7 (2009), 851–867.

[CG96] COELHO J., GRUPEN R.: Online grasp synthesis. *International Conference on Robotics and Automation* (1996).

[CH90] CUTKOSKY M. R., HOWE R. D.: Human grasp choice and robotic grasp analysis. In *Dextrous robot hands*, Venkataraman S. T., Iberall T., (Eds.). Springer-Verlag New York, Inc., New York, NY, USA, 1990, pp. 5–31.

[dSB03] DAVELLA A., SALTIEL P., BIZZI E.: Combinations of muscle synergies in the construction of a natural motor behavior. *Nature Neuroscience 6* (2003), 300–308.

[Gra98] GRASSIA F. S.: Practical parameterization of rotations using the exponential map. *Journal Graphics Tools 3*, 3 (1998), 29–48.

[Ijs01] IJSPEERT A.: A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander. *Biological Cynernetics 84* (2001), 331Ű348.

[JF09] JOHANSSON R., FLANAGAN J.: Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nature Reviews Neuroscience 10* (2009), 345Ű359.

[JXL10] J. XU T. K., LI Z.: Sampling-based finger gaits planning for multifingered robotic hand. *Autonomous Robots 28*, 4 (2010).

[KVC12] KUTCH J., VALERO-CUEVAS F.: Challenges and new approaches to proving the existence of muscle synergies of neural origin. *PLoS Computational Biology 8*, 5 (2012).

[Liu08] LIU C. K.: Synthesis of interactive hand manipulation. *Symposium on Computer Animation* (2008), 163–171.

[Liu09] LIU C. K.: Dextrous manipulation from a grasping pose. *ACM Trans. Graph. 28*, 3 (2009).

[LT07] LOCKHART D., TING L.: Optimal feedback transformations for balance. *Nature Neuroscience 10* (2007), 1329–1336.

[MA99] MILLER A. T., ALLEN P. K.: Examples of 3d grasp quality computations. *International Conference on Robotics and Automation* (1999), 1240–1246.

[MF08] M V., F. V.-C.: Neural control of motion-to-force transitions with the fingertip. *Journal of Neuroscience 28* (2008), 1366–1373.

[MLPP09] MUICO U., LEE Y., POPOVIC J., POPOVIC Z.: Contact-aware nonlinear control of dynamic characters. *ACM Trans. Graph. 28*, 3 (2009).

[MM04] MILLER A. T., MILLER A. T.: Graspit!: A versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine 11* (2004), 110–122.

[MTP12] MORDATCH I., TODOROV E., POPOVIC Z.: Discovery of complex behaviors through contact-invariant optimization. *ACM Trans. Graph. 31*, 4 (2012).

[OSC00] OKAMURA A. M., SMABY N., CUTKOSKY M. R.: An overview of dexterous manipulation. *International Conference on Robotics and Automation* (2000), 255–262.

[PT12] POSA M., TEDRAKE R.: Direct trajectory optimization of rigid body dynamical systems through contact. *Workshop on the Algorithmic Foundations of Robotics* (2012).

[PZ05] POLLARD N. S., ZORDAN V. B.: Physically based grasping control from example. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2005), SCA '05, ACM, pp. 311–318.

[RS09] RATHELOT J., STRICK P.: Subdivisions of primary motor cortex based on cortico-motoneuronal cells. *Proceedings of the National Academy of Sciences 106*, 3 (2009), 918–923.

[SFS98] SANTELLO M., FL M., SOECHTING J. F.: F.: Postural hand synergies for tool use. *The Journal of Neuroscience 18*, 3 (1998), 10105–10115.

[SH07] SAFONOVA A., HODGINS J.: Construction and optimal search of interpolated motion graphs. *ACM Trans. Graph.* (2007).

[Sim94] SIMS K.: Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1994), SIGGRAPH '94, ACM, pp. 15–22.

[SJL09] S. JAIN Y. Y., LIU K.: Optimization-based interactive motion synthesis. *ACM Trans. Graph.* (2009).

[SKP08] SUEDA S., KAUFMAN A., PAI D. K.: Musculotendon simulation for hand animation. *ACM Trans. Graph. 27*, 3 (2008).

[TG04] TODOROV E., GHAHRAMANI Z.: Analysis of the synergies underlying complex hand manipulation. *IEEE Engineering in Medicine and Biology* (2004), 4637–4640.

[TJ09] TRESCH M., JARC A.: The case for and against muscle synergies. *Current Opinion in Neurobiology 19* (2009), 601–607.

[Tod11] TODOROV E.: A convex, smooth and invertible contact model for trajectory optimization. *International Conference on Robotics and Automation* (2011), 1071–1076.

[TSB99] TRESCH M., SALTIEL P., BIZZI E.: The construction of movement by the spinal cord. *Nature Neuroscience 2* (1999), 162–167.

[TT10] TASSA Y., TODOROV E.: Stochastic complementarity for local control of discontinuous dynamics. *Robotics: Science and Systems* (2010).

[WRWM04] WEGHE M. V., ROGERS M., WEISSERT M., MATSUOKA Y.: The act hand: Design of the skeletal structure. *International Conference on Robotics and Automation* (2004), 3375–3379.

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

[YL12] YE Y., LIU K.: Synthesis of detailed hand manipulations using contact sampling. *ACM Trans. Graph. 31*, 4 (2012).