Dropping the Ball: Releasing a Virtual Grasp

Mores Prachyabrued†
University of Louisiana at Lafayette

Christoph W. Borst†
University of Louisiana at Lafayette

ABSTRACT

We present a method for improved release of whole-hand virtual grasps. It addresses the problem of objects “sticking” during release after the user’s (real) fingers interpenetrate virtual objects due to the lack of physical motion constraints. This problem may be especially distracting for grasp techniques that introduce mismatches between tracked and visual hand configurations to prevent visual interpenetration. Our method includes heuristic analysis of finger motion and a transient incremental motion metaphor to manage a virtual hand during grasp release. We incorporate the method into a spring model for whole-hand virtual grasping. We show that the new spring model improves speed and accuracy for a targeted ball-drop task, and users report a subjective preference for the new behavior. In contrast to a standard spring-based grasping method, measured release quality does not depend notably on object size.

KEYWORDS: Grasp release, virtual grasping, whole-hand interaction, grasp heuristic.


1 INTRODUCTION

We present a method for improved virtual grasping, particularly the release of grasps. Whole-hand grasping is intended to allow the use of hands to interact with objects. A virtual environment that supports such grasps successfully has a low learning curve for users, if they can interact with the environment naturally. This form of interaction is also a necessity in some applications. For example, a virtual reality training system for a manual assembly task may need to provide natural whole-hand interaction for appropriate trainee experience.

A “sticky object” grasp release problem occurs when a user’s fingers (real, not rendered) can sink into a virtual object, and the effect may be especially unpleasant when there is a mismatch between tracked and visual hand configurations. For example, there is a mismatch in the spring-based model of Borst and Indugula [1] to prevent visual interpenetration artifacts. Without physical constraints from a real object, users tend to close their (real) fingers into virtual objects. Since the visual/spring model no longer matches the real hand, an object can appear to stick to the hand (exaggerated finger motions are needed to release the object), and a user can not know precisely when a grasp will release. This led Borst and Indugula to suggest a “light touch” with their approach, and its performance requires practice for some users. The problem may be reduced by force feedback, considering such feedback has been shown to reduce hand closing [2]. However, force feedback does not completely prevent fingers from penetrating virtual objects due to device limitations that are not likely to be overcome soon (e.g., limited degrees of freedom). We would also like to support grasping in environments without force feedback, for example, in systems where the hand is optically tracked and worn or complex devices are not desired.

Burns et al. [3] proposed a virtual-hand management method (MACBETH) that attempted to correct sticking. However, it manages only virtual hand base position (no orientation or articulation management defined), so it could not be applied to whole-hand grasping without extensions. Their work also suggested that preventing visual interpenetration artifacts is important.

We propose a new spring model based on Borst and Indugula's spring model (original) [1]. Our model improves grasp release while retaining the characteristics of physically-based grasping, the prevention of visual interpenetration artifacts, and compatibility with the force rendering method of the original approach. The original spring model couples a simulation-controlled articulated hand model (called the virtual hand or spring hand) to tracked (real) hand configuration using a system of linear and torsional virtual spring-dampers. This resembles the rubber band metaphor [4] to manage a virtual hand during release of grasps, which was shown to cause problems [5]. Instead, our new spring model incorporates a heuristic analysis of finger motion to detect a user's intent to release the grasped object, and it uses a transient incremental motion metaphor to manage a virtual hand during a release period.

The contributions described in this paper are:

- We present a new spring model for whole-hand virtual grasping that includes a method for improved release.
- We present heuristic analysis of finger motions to detect a user's intent to release the grasped object.
- We present an experimental evaluation of our method. It shows that our method improves speed, accuracy, and subjective experience during grasp release.
- Our experiment also demonstrates that the release performance of a standard (original) grasping approach decreases substantially with increasing object size.

2 PREVIOUS WORK

2.1 Physically-Based Grasping

Physically-based grasping models, such as the one we build on, aim to provide realistic interaction by simulating object motion according to laws of physics. Bergamasco, Degl’Innocenti, and Bucciarelli [6] introduced the use of physically-based object response to achieve whole-hand interaction. They defined grids of control points on a virtual hand to detect contacts between a virtual hand and a virtual object and to compute force vectors acting on the object, including normal contact forces, dynamic frictions, and static frictions. Manipulation was limited to objects with simple shapes.

Using a similar idea, Hirota and Hirose [7] demonstrated dexterous manipulation of objects with complex shapes in a
manipulation system. They used a much larger number of points and a fast collision response computation method.

Borst and Indugula [1] extended the concept of virtual coupling to the whole hand. A virtual hand model was coupled to the tracked hand using a system of linear and torsional spring-dampers. These created forces necessary to simulate physically-based grasping using a widely-available simulation tool. Their technique prevented hand-object interpenetration not accounted for in the two previous works.

Allard et al. [8] used images of a real-world object, captured from different viewpoints, to construct a 3D model representation and inject it into a physically-simulated virtual environment in real-time. The object could be a human hand. These made physically-based hand interaction with virtual objects possible.

Wilson et al. [9] presented physically-based grasping on an interactive surface. They modeled surface contacts as rigid bodies that interacted with virtual objects using physical simulation. This was a limited form of whole-hand grasping near a surface, not a general approach for 3D space.

### 2.2 Heuristics-Based Grasping

Heuristics-based grasping refers to grasping approaches that use heuristics (instead of laws of physics) to determine grasp state (grasp/release) and object motion during grasp. We studied these approaches for our heuristic analysis of finger motions.

Iwata [10] tested 16 control points on a virtual hand for contact with a virtual object. The object was grasped when it was touched by the thumb and one of the other fingers. A grasped object’s coordinate frame was then attached to the hand coordinate frame so that the object moved with the hand. A similar idea using two fingers was presented by Maekawa and Hollerbach [11].

In their virtual assembly environment, Wan, Gao, and Peng [12] abstracted mechanical components into simple primitives (cube, sphere, and cylinder). Possible grasping postures were predefined for each pair of primitive type and size. An object was grasped if collision detection indicated that user hand posture matched one of the previously defined grasping patterns for the object. The object was manipulated by considering its coordinate frame as a child node of hand.

Hilliges et al. [13] allowed pick-up on an interactive surface by detecting a pinch gesture. An object would be under grasp control (with limited rotation) if a ray, projected downward from the center of mass of a hole formed by the gesture, intersected the object's coordinate frame. Possible grasping postures were predefined for each pair of primitive type and size. An object was grasped if collision detection indicated that user hand posture matched one of the previously defined grasping patterns for the object. The object was manipulated by considering its coordinate frame as a child node of hand.

Ullmann and Sauer [14] presented heuristics, based on contact geometry, for establishing one-hand and two-hand grasps. They presented a fine object manipulation method for computing object motion (not just attaching an object's frame to the hand frame) after grasp had been established.

Holz et al. [15] and Moehringer and Froehlich [16] presented grasping heuristics and object manipulation methods that are more general. They supported multi-user, multi-hand, multi-finger, and multi-object interactions. They both used the concepts of grasping pairs and friction cones. Heuristic approaches are not as general as physically-based grasping, but they may perform well for their intended tasks.

While many heuristic approaches used violation of a grasp condition to determine release state, [16] presented explicit release heuristics based on distances of involved grasping pairs. In contrast, our release heuristics consider finger motions.

### 2.3 Virtual Hand Management

A virtual hand that simply follows a tracked hand configuration typically penetrates virtual objects during interaction (due to lack of motion constraints). There are techniques that prevent the visual interpenetration artifacts with resulting discrepancy between virtual and real hands (which complicates the release of grasps, as pointed out in the introduction). Work by Burns et al. [5], suggesting that users are more sensitive to visual interpenetration than to visual-proprioceptive discrepancy, motivates the prevention of visual interpenetration. Zachmann and Rettig [4] discussed two metaphors that can be used to manage a virtual hand after the virtual and real hands separate:

1. **The rubber band metaphor**: the virtual hand maintains its configuration as close as possible to the real hand.
2. **The incremental motion metaphor**: the virtual hand moves by the same amount as the real hand.

Each metaphor has a drawback. The rubber band metaphor causes the virtual hand to stick to a virtual object's surface upon release [5]. The phenomenon was similarly observed in other systems using this type of metaphor [1, 17]. The incremental motion metaphor does not have the sticking problem, but it maintains an offset between the virtual and real hands. It was reported in [5] that maintaining offset between virtual and real hands reduced user performance.

Burns et al. [3] proposed a third metaphor - MACBETH. It involves incremental motion, but it removes position discrepancy by introducing velocity discrepancy that is similarly detectable. Based on their user study comparing MACBETH to the previous metaphors, MACBETH improved user-rated naturalness and user preference while no loss in user performance was detected. However, MACBETH, in its current form, can only manage virtual hand base position. Additional work is needed to extend MACBETH to manage hand orientation and finger joint angles. In contrast, rubber-band and incremental motion metaphors are applicable to both.

A simpler technique to reduce offset between virtual and real hands was used in Immersion's VirtualHand Toolkit [18]. An offset is gradually reduced to zero when the real hand moves in a direction away from contact.

We propose a virtual hand management technique (for whole-hand configuration) for improved release of grasped objects. In particular, it involves the use of incremental motion with offset reduction to manage finger joint angles during grasp release.

![Figure 1. Borst and Indugula's spring model [1] showing tracked hand (left), virtual hand (right), and some of the virtual spring-dampers.](image-url)

### 3 Borst and Indugula’s Spring Model

#### 3.1 Description of the Spring Model

Borst and Indugula [1] proposed a physically-based grasping approach that extended the virtual coupling concept to an
articulated hand. The approach couples a spring (virtual) hand to a tracked (real) hand using a system of virtual linear and torsional spring-dampers. This produced forces and torques necessary for virtual hand motion using dynamic simulation and for physically-based response of grasped objects via collision response.

Figure 1 illustrates their spring model. They used 21 torsional and 6 linear virtual spring-dampers. There was one torsional element for each of 20 finger joint degrees of freedom (illustrated only for the index finger), one torsional element and one linear element for the base of the hand (illustrated), and one linear element for each of the five digit tips (not illustrated).

In addition to supporting grasping and manipulation, this spring model addressed the problem of visual interpenetration and included force rendering for force-feedback gloves.

3.2 Grasp Release Problem and the Spring Model

The spring model can be considered a rubber hand metaphor to manage a virtual hand: the virtual hand maintains a configuration (palm pose and finger joint angles) pulled toward the tracked hand configuration but subject to constraints. This can cause the virtual hand to stick to a virtual object upon grasp release, as mentioned in [5] and indicated as a motivation for using a “light touch” in [1]. Figure 2 illustrates the problem. A user closed the fingers further than necessary, and, when the user opens them to release, they may remain inside the object, causing the object to appear stuck (or the hand model to appear unresponsive). The user can exaggerate finger motions to release, but this reduces naturalness and interferes with precision tasks (our experiment will demonstrate reduced accuracy).

Notably, the problem also occurs to an extent even if the visual hand model is allowed to penetrate objects to match tracked hand configuration. The real fingers still sink into objects due to lack of real motion constraints and small motions may not be sufficient to release grasp. However, the visual hand pose would not be misleading and users might better anticipate release behavior and not interpret it as a sticking object.

Figure 2. Grasp showing tracked hand (mesh) that sank into the virtual object and virtual hand (solid) that remained at the object’s surface.

4 GRASP RELEASE METHOD AND NEW SPRING MODEL

The two key ideas in our method for improving grasp release are:

1. The use of heuristic analysis of finger motions (release-heuristic function) to detect user’s intent to release grasped objects.
2. The use of a transient incremental motion metaphor with subsequent convergence period to manage the virtual hand during release of grasps.

We illustrate the method with our new spring model for virtual grasping.

4.1 New Spring Model

4.1.1 Three Hand Configurations Concept

Our new spring model behaves similarly to that of [1] except during, and for a short time following, grasp release. To incorporate the incremental motion metaphor for release, the new spring model defines three hand configurations:

1. Tracked hand refers to the real hand configuration as measured by sensing hardware and calibration steps.
2. Spring (virtual, visually-rendered) hand refers to a simulation-controlled virtual hand configuration.
3. Target hand refers to a target configuration for the virtual hand.

The virtual hand is coupled to the target hand (instead of the tracked hand as in [1]) using a system of linear and torsional spring-dampers. Figure 3 illustrates the target hand concept. When a user opens their hand (by a “delta” amount) to release a virtual object, we update the target hand configuration to the “current virtual hand configuration plus the delta”. This has the effect of instructing the virtual hand to open by the same delta (as in the incremental motion metaphor), causing it to release the object more immediately than waiting for fingers of the tracked hand to exit the object’s surface. Subsequently, the target hand is adjusted by a convergence mechanism.

Figure 3. Target-hand outside the object, causing the virtual hand to open more immediately even when the tracked-hand finger is still inside the object.

4.1.2 Target Hand Update Algorithm

We update target-hand configuration (palm pose and finger joint angles) for every new tracked-hand configuration. For grasping of unconstrained objects, which is our focus, the grasp release problem comes mostly from finger motions (finger penetrations) and not from palm motions (palm penetration). Therefore, target-hand palm (the base frame for the hand) simply matches tracked-hand palm. For the target-hand finger joint angles, the equations below describe the main update component. We evaluate a release-heuristic function (Section 4.2) prior to the update. For each joint angle in a hand joint model (Section 4.3):

If the release-heuristic function detected release,
\[ \theta_{tg1} = \theta_{tg0} + (\theta_{tg1} - \theta_{tg0}) \]  
Otherwise,
\[ \theta_{tg1} = \theta_{tg0} + (\theta_{tg1} - \theta_{tg0}) \]

where:

\[ \theta_{tg0} \] is the initial target joint angle,
\[ \theta_{tg1} \] is the new target joint angle,
\[ \theta_{tg} \] is the tracked joint angle.
\(\theta_{\text{tg}}\), \(\theta_{\text{tr}}\) are next (post-update) and current (pre-update) joint angles of the target hand, 
\(\theta_{\text{tg}}, \theta_{\text{tr}}\) are next and current joint angles of the tracked hand, 
\(\theta_{\text{tg}}, \theta_{\text{tr}}\) is the current joint angle of the virtual hand.

\(\theta_{\text{tg}}\) is also subject to an additional update mechanism described at the end of this section.

Initially, target and virtual hands are set to the same configuration as tracked hand. Before release, target-hand finger configuration (finger joint angles) will be equal to tracked-hand finger configuration (they move by the same delta, see Equation 2). This results in the same virtual-hand behavior (w.r.t. finger motions) as the original spring model. The behavior begins to differ at the time of release (when release-heuristic function detects release). Target-hand finger configuration will be set to virtual-hand finger configuration plus the change undergone by the tracked-hand fingers (Equation 1). Later, target-hand finger configuration will be updated using Equation 2 (release-heuristic function no longer detects release). This is similar to the incremental motion metaphor to manage virtual-hand fingers during release. However, this creates and maintains an offset during release. However, this creates and maintains an offset increment.

The virtual-hand finger configuration will be updated using Equation 2 (release-heuristic function no longer detects release). This is similar to the incremental motion metaphor to manage virtual-hand fingers during release. However, this creates and maintains an offset increment.

4.3 Implementation Details

We use a standard hand joint model similar to a CyberGlove joint model [19]. Each of four fingers has a 2-dof metacarpophalangeal joint (MPJ) for abduction and flexion at the first knuckle and a 1-dof interphalangeal joint (IJ) at each of the remaining two knuckles for flexion (PJ for the second knuckle and DIJ for the third knuckle). The thumb has a 2-dof trapeziometacarpal joint (TMJ) in the palm for roll and abduction, a 1-dof MPJ for flexion at the first knuckle, and a 1-dof IJ for flexion at the second knuckle.

We use the following values to implement our new spring model and release-heuristic function: 
\(c = 0.035^{\circ}\) degrees, \(L = 3\), 
\(F_t = \{\text{TMJ-roll, MPJ-flexion}\}, F_{ij} = \{\text{MPJ-flexion, PJJ-flexion}\}, F_{am} = \{\text{MPJ-flexion}\}, F_i = \{\text{MPJ-flexion}\}, F_p = \{\text{MPJ-flexion}\}\). We set the thresholds parameters to integer multiples of calibrated angular resolutions of finger sensors at the corresponding joints (e.g., sensor gains for CyberGlove). The multipliers for thumb
joints provided torsional springs for finger joint angles. We used simulation with collision detection and response. PhysX revolute QuadroFX 5800 graphics card.

Xeon E5450 3.00GHz processors, 8GB RAM, and an NVIDIA synchronized with the monitor refresh. The head (viewpoint) was computed as two thirds of the middle knuckle angles. Palm base doesn't have sensors at distal finger joints, so their angles are sensed by an 18-sensor right-handed CyberGlove (this glove viewing via CrystalEyes LCD shutter glasses. Joint angles were and refresh rate was 100 Hz, for time-multiplexed stereoscopic real and virtual workspaces. Monitor resolution was 1024 x 768 inch monitor placed at a 45° angle above a mirror) to co-locate.

Figure 4 illustrates the grasping system hardware for the main study.

5 EXPERIMENT
We conducted within-subjects experiments to compare our approach to the standard spring model from [1] (we implement the standard spring model by simply setting target hand to tracked hand, disabling new mechanisms). The main study was a targeted ball-drop task and had the following independent variables:

1. Grasping Technique – new and old spring models.
2. Object Size – small (radius = 3.0 cm), medium (4.5 cm), and large (6.0 cm).

The dependent variables were:
1. Release Time – amount of time required to release a grasped ball.
2. Translational Error – horizontal movement of a ball resulting from grasp release.

We hypothesized that the new spring model improves speed and accuracy of the ball-drop release.

We also conducted a subjective comparison experiment in which a virtual environment contained two objects, using the two different grasp techniques, and users indicated which was easier to release and which was easier to pick up. This allowed us to determine whether or not users could detect quality differences (they were not informed which object used which technique). Object size was not varied for this study, but we included three different medium-sized shapes: ball (5.25 cm), cube, and bunny.

We hypothesized that the new technique provides easier release based on subjective comparison.

We note that two other experiments were also conducted. For clarity and due to space constraints, we focus on the two above and later mention the other studies only briefly (Section 6.3). The findings of other studies were consistent with the findings of the main study.

5.1 Apparatus
Figure 4 illustrates the grasping system hardware for the experiment. We used a mirror-based “fish tank” VR display (21-inch monitor placed at a 45° angle above a mirror) to co-locate real and virtual workspaces. Monitor resolution was 1024 x 768 and refresh rate was 100 Hz, for time-multiplexed stereoscopic viewing via CrystalEyes LCD shutter glasses. Joint angles were sensed by an 18-sensor right-handed CyberGlove (this glove doesn’t have sensors at distal finger joints, so their angles are computed as two thirds of the middle knuckle angles). Palm base pose was tracked by an Ascension miniBird system that was synchronized with the monitor refresh. The head (viewport) was not tracked. Audio output was via ordinary stereo speakers. All software ran on a Dell Precision T5400 with two Intel quad-core Xeon E5450 3.00GHz processors, 8GB RAM, and an NVIDIA QuadroFX 5800 graphics card.

The NVIDIA PhysX SDK (www.nvidia.com) provided physical simulation with collision detection and response. PhysX revolute joints provided torsional springs for finger joint angles. We used the equations from [1] for the springs at the base of the hand (palm). We omitted the linear fingertip springs from [1].

Our visual hand model consisted of 16 segments and resembled the model provided with CyberGlove devices. Our OpenGL-based visual rendering system included shadow-mapped shadows. Our application was separated into two main threads: a graphics thread for graphics rendering and an interaction thread for hand data processing and simulation.

5.2 Subjects
28 subjects participated in the experiment: 25 males and 3 females, aged 20 to 33 years (average = 25), 23 right-handed and 5 left-handed. Almost all subjects (27) were students, mostly from computer science and computer engineering programs. Experience levels were mixed: 5 reported previous exposure to virtual grasping (presumably from demos in our lab), 9 others reported exposure to VR systems, and all of the remaining 14 took a graphics class, played video games, or watched 3D movies.

5.3 Design
Considering the two experiments detailed here and additional experiments, subjects performed five total tasks: a learning task, a pick-and-drop experiment (not detailed), the targeted ball-drop experiment, a cube-alignment experiment (not detailed), and the subjective-comparison experiment. To reduce possible effects of fatigue and short-term learning, we split experiments into two days, with a different technique presented per day (order randomized per subject such that half of the subjects experienced the new approach on their first day, and half experienced the other approach first). On both days, subjects completed tasks in this order: learning task, pick-and-drop, targeted ball-drop, and cube-alignment. Additionally, subjects completed the subjective-comparison task only at the end of the second day, because it involved exposure to both techniques in each of its trials. We calibrated the CyberGlove for each subject before they started per day. Experiment duration was typically 30 to 45 minutes per day.

We split experiments into sessions conducted in the following order:

1. A demo session with on-screen instruction to introduce subjects to the task. It demonstrated one trial.
2. A practice session that allowed subjects to practice the task without instruction. It consisted of three trials. As an exception, the subjective-comparison experiment had no practice session.
3. The actual experiment session for measuring subject performance. It contained no instructions.
5.3.1 Procedure for Learning Task
During the learning task (Figure 5), subjects practiced virtual grasping with ball, cube, and bunny objects. They were required to lift and drop each object at least 5 times to practice grasping and releasing interactions.

5.3.2 Procedure for Targeted Ball-Drop Experiment
In the ball-drop experiment (Figure 6), subjects picked up a ball from the virtual “floor” and dropped it from above a red target mark on the floor. In the demo session, subjects were told that a floating wireframe cube above the target was the best place to drop the ball (the cube center was aligned with the center of the red mark). The components of the trial are explained by the demo session instructions:

1. Pick up the ball and move it inside the cube. The cube will turn green and the (2-second) countdown sound will begin.
2. Wait for the countdown sound to end.
3. Release the ball immediately at the end of the countdown sound using normal finger motion.

There were 9 trials in the experiment session: three per ball size. Condition order was randomized per subject.

5.3.3 Procedure for Subjective Comparison Experiment
The subjective comparison experiment (Figure 7) had subjects compare the two grasping techniques directly. In each trial, there were two similar objects at the left and right sides of the scene, separated by an invisible wall at the center (objects could not cross the wall). The left object was manipulated using one grasping technique (randomized per trial) while the right object was manipulated using the other grasping technique. A question displayed at the top of the scene asked subjects to choose the object that was easier to release. After free exploration, subjects pressed a CyberGlove-mounted switch when they were ready and indicated an object by touching it with the index fingertip for 2 seconds. A second question then asked them to indicate the object that was easier to pick up, and they answered using a similar procedure.

There were 9 trials in the experiment session: three trials each for ball, cube, and bunny. The size for each object was a middle size between medium and large sizes from the earlier experiment. Object type order was randomized per subject.

6 RESULTS AND DISCUSSION

6.1 Targeted Ball-Drop Experiment Results
We computed release time and translation error values for the ball-drop experiment as follows:

Let:
- \( t_1 \) be the time instant when the countdown sound ends,
- \( t_2 \) be the time instant when no finger phalanges of the virtual hand touch the ball (the experiment software does not allow multiple grasps in a trial, so this is the end of the single grasp),
- \( t_3 \) be the instant when the ball touches the floor,
- \( \mathbf{d} \) be the projected vector of \((\mathbf{p}_{t_3} - \mathbf{p}_{t_1})\) on the floor, where \( \mathbf{p}_{t_1} \) and \( \mathbf{p}_{t_3} \) are positions of the ball origin at times \( t_1 \) and \( t_3 \), respectively.

Then:

- Release time = \( t_2 - t_1 \), and
- Translational error = \( \text{length}(\mathbf{d}) \).

Note that error is defined independently of user targeting error. It is a measure of horizontal motion that results from release.

Figures 8 and 9 summarize these release times and errors. We performed two-way repeated-measures ANOVA per dependent variable. Due to an interaction, we additionally performed one-way repeated measures ANOVA per grasping technique, with object size as the independent variable. Reported post-hoc test p-values include Bonferroni correction.

For release time:

1. There was a significant effect of grasping technique, \( F(1,27) = 18.02, p < .001 \).
2. There was a significant effect of object size, \( F(2,54) = 18.94, p < .001 \).
3. There was a significant technique-size interaction, \( F(2,54) = 9.99, p < .001 \).
Mean release time with the new spring model was 27% shorter than with the old (standard) spring model on average. Mean release time for the large ball was significantly longer than for medium and small balls by 19% (p < .05) and 35% (p < .001), respectively. Mean release time for the medium ball was significantly longer than for the small ball by 14% (p < .001).

**Figure 8. Release time for targeted ball-drop task.**

The per-technique tests revealed a significant effect of object size for the old spring model (F(2,54) = 17.81, p < .001) with pairwise comparisons detecting significance for all pairs. However, no significant effect of object size was detected for the new spring model (F(2,54) = .95, p = .395).

For translational error:

1. There was a significant effect of grasping technique, F(1,27) = 63.32, p < .001.
2. There was a significant effect of object size, F(2,54) = 11.79, p < .001.
3. There was a significant technique-size interaction, F(2,54) = 12.34, p < .001.

Mean translational error for the new spring model was 44% smaller than for the old spring model on average. Mean translational error for large and medium balls was significantly larger than for the small ball by 49% (p < .005) and 24% (p < .01), respectively. Mean translational error for the large ball was near-significantly larger than for the medium object by 20% (p = .095).

The per-technique tests revealed a significant effect of object size for the old spring model (F(2,54) = 18.29, p < .001) with pairwise comparisons detecting significance in all pairs except the medium-small pair (which showed near significance, p = .065). However, no significant effect of object size was detected in the new spring model (F(2,54) = 1.54, p = .22).

### 6.2 Subjective Comparison Experiment Results

For the subjective comparison experiment, we computed a per-subject score as the number of times the subject picked the new spring model over the number of contributing trials (i.e., percentage of trials for which the new technique was chosen as easier). Figure 10 summarizes the results.

Subjects reported that grasp release was easier for the new spring model than for the old model: overall mean score for the release question was significantly above 0.5 (t(27) = 12.06, p < .001; all reported tests are two-tailed). Overall, the object manipulated using the new spring model was picked 61% of the time. Furthermore, the result holds for the ball object (t(27) = 3.67, p < .005) but not for other objects (cube: t(27) = .59, p = 1.00; bunny: t(27) = 1.32, p = .597).

**Figure 9. Translational error for targeted ball-drop task.**

**Figure 10. Percentage of trials for which the new spring model was chosen in subjective comparison task (mean and standard error of per-subject scores).**

### 6.3 Discussion

The results from the targeted ball-drop task confirm our hypothesis that the new spring model improves speed and accuracy of grasp release (by 27% and 44%, respectively, on average, for our task). This can be explained by the new spring model requiring less finger extension to release grasped objects due to the use of the incremental motion metaphor during grasp release. Less required finger extension provides faster release and less sticking of grasped objects, which also improves release accuracy.

The results also show that it took significantly longer to release larger objects than smaller ones with the old (standard) spring method, with associated reduced accuracy. This would be explained by larger objects resulting in larger interpenetration, which may simply be due to the larger range of motion available, or to something more complex like tighter grasps learned for larger objects that are expected to be heavier. The new model appears to alleviate the problem, as no significant effect of size was detected in the new spring model, and the resulting means and standard errors suggest any present effect would be relatively small. In the new spring model, if the heuristic analysis detects release motion, the virtual hand will open almost immediately independent of the amount of (real) finger penetrations due to the use of incremental motion metaphor.

The subjective comparison results confirm our hypothesis that the new approach provides easier release subjectively, and this is
consistent with the objective ball-drop results. Furthermore, the results from the pickup question provide some evidence that the new approach does not induce disturbing pickup problems (addressing possible concerns that the release-heuristic function could incorrectly trigger during pickup, which would result in the object slipping out of grasp). Most subjects stated in the learning session of this task that they could not detect any difference for the pick-up question. We expect that there was actually no effect of grasping technique on the pickup action, since new and old spring models behave similarly during pickup (assuming no side-effect from the use of the release-heuristic function). The results (better subjective pickup with new technique, overall and for the ball object) may reflect overall subject experience with the object during the trial, including release, rather than differences during pickup.

Our results demonstrate that maintaining a pose discrepancy (of finger configurations) with subsequent reduction can improve user performance and subjective experience. This augments the results of Burns et al. [3], who introduced discrepancy in hand base position and showed improved user ratings with no loss in performance for a hand navigation task.

Two other experiments that were not detailed here were a pick-and-drop experiment and a cube-alignment experiment. Their results were consistent with the reported experiments (significant effects of technique and size, with the new model outperforming the old model), with no contrary findings of significance. The pick-and-drop task was simpler than the ball-drop task in that no targeting was required (except coarsely crossing a threshold) and no accuracy was measured. However, it included different object types, additionally showing that ball interactions were faster than cube and bunny interactions. The cube-alignment task included no gravity simulation and thereby resembled a task where a user arranges 3D scene or interface components using the hand (with objects sticking in place after release). It required 3D rotation to align a cube to a target cube, so rotation error was measured in addition to translation error, showing rotational error reduction with the new method.

7 CONCLUSION AND FUTURE WORK

We described a grasp release problem that is relevant for any virtual grasping technique that requires fingers to be moved outside an object’s boundaries for release. The problem may have especially unpleasant effects when these approaches prevent visual interpenetration. Our new spring model for virtual grasping successfully addresses this problem, improving grasp release. This was confirmed by a human factors study showing significantly improved speed, accuracy, and subjective experience during targeted ball-drop and subjective comparison tasks.

We want to improve our heuristic analysis of finger motions and formally evaluate specific heuristics. The current hand convergence algorithm manifests a type of visual-proprioceptive motion discrepancy – stationary-self (the virtual hand moves but the real hand does not), which may be detectable [5]. We want to apply a concept similar to MACBETH [3] to tradeoff position discrepancy with motion discrepancy in a more intelligent way, for whole-hand convergence with a fully articulated hand model.

REFERENCES