Sticky Tools: Full 6DOF Force-Based Interaction for Multi-Touch Tables

Mark Hancock[1], **Thomas ten Cate**[1,2], **Sheelagh Carpendale**[1]

[1]University of Calgary
Department of Computer Science
{msh,sheelagh}@cpsc.ucalgary.ca

[2]University of Groningen
Department of Computer Science
t.ten.cate.1@student.rug.nl

ABSTRACT

Tabletop computing techniques are using physically familiar force-based interactions to enable compelling interfaces that provide a feeling of being embodied with a virtual object. We introduce an interaction paradigm that has the benefits of force-based interaction complete with full 6DOF manipulation. Only multi-touch input, such as that provided by the Microsoft Surface and the SMART Table, is necessary to achieve this interaction freedom. This paradigm is realized through *sticky tools*: a combination of *sticky fingers*, a physically familiar technique for moving, spinning, and lifting virtual objects; *opposable thumbs*, a method for flipping objects over; and *virtual tools*, a method for propagating behaviour to other virtual objects in the scene. We show how sticky tools can introduce richer meaning to tabletop computing by drawing a parallel between sticky tools and the discussion in Urp [20] around the meaning of tangible devices in terms of nouns, verbs, reconfigurable tools, attributes, and pure objects. We then relate this discussion to other force-based interaction techniques by describing how a designer can introduce complexity in how people can control both physical and virtual objects, how physical objects can control both physical and virtual objects, and how virtual objects can control virtual objects.

INTRODUCTION

In the physical world, an object reacts to a person's actions depending on its physical properties and the forces applied to it. For example, a book can be stacked on top of another because it has two flat sides or a pencil can be rolled along a desk because it is cylindrical. People often make use of the unique properties of objects to make them affect other objects in different ways. People use pencils to write, hammers to insert nails, and utensils to cook food. In the virtual world, how objects react to human intervention depends on a particular mapping of human movement to computer feedback. For example, pressing a button with a mouse cursor can cause a variety of behaviour, including opening a menu, advancing to the next page of a document, or invoking a new window to appear. There are benefits to both worlds; in the physical world, people become familiar with the capabilities of the tools they use regularly; in a virtual world the result of a person's actions can be made to either use or extend physical limits.

ITS '09, November 23–25, 2009, Banff, Alberta, Canada.

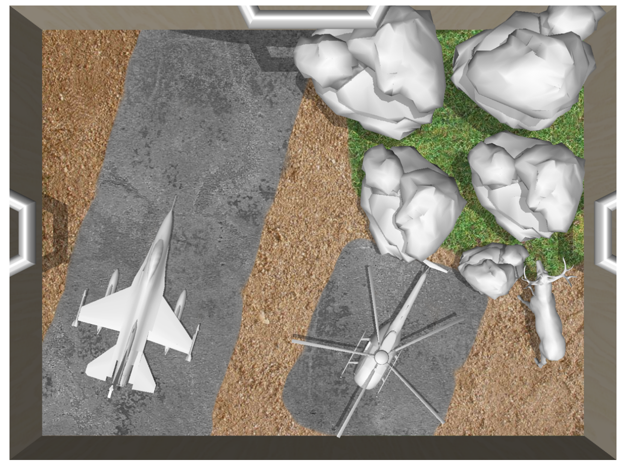


Figure 1: A screenshot of a 3D virtual scene.

Since tabletop displays afford direct touches for interaction, the techniques have a feeling of being more physical than, for example, mouse or keyboard interaction. This directness of interaction with virtual objects opens up the potential for interactive tables to simultaneously leverage the benefits of both the physical and the virtual. The research question is: how does one maintain the feeling of physical interaction with the full capabilities to manipulate a 3D scene such as in Figure 1? Many of the techniques that have been designed specifically for digital tables are based (either explicitly or implicitly) on how objects react in the physical world. However, these techniques typically resort to techniques such as gestures [1, 22] or menus [18] to provide full functionality.

We introduce sticky tools—virtual 3D tools that can be manipulated in the full six degrees of freedom (DOF) of translation and rotation—to allow force-based interaction to provide full control of a system, without the need for gestures or menus. We first describe the historical progress of force-based interaction, we then introduce sticky tools, and then we demonstrate how sticky tools can be used to assign richer meanings to virtual objects. We end with a discussion of how sticky tools leverage the largely unexplored research area of how virtual objects interact with other virtual objects and describe how this research direction can overcome existing limitations in tabletop interaction.

RELATED WORK

Digital tables have used force-based interaction since they were introduced, both explicitly through metaphor and implicitly through 2D or 3D manipulation.

Force-Based Metaphors

Many tabletop display interfaces use force-based metaphors to create compelling new interactions. The Personal Digital Historian [17] uses the idea of a "Lazy Susan" to invoke the metaphor of spinning virtual objects to another side of the table. The Pond [19] uses the metaphor of a body of water where virtual objects can sink to the bottom over time. Interface currents [9] demonstrate how the idea of flow can be applied to virtual objects; virtual objects can be placed in a dedicated area on the table that acts like a river, carrying the virtual objects to another part of the screen.

A more abstract property of force-based interaction is that local actions only cause local behaviour, though this behaviour can then propagate to have a larger area of influence. For example, dropping a stone in water initially affects a small area, and over time its ripples eventually affect the entire body of water. Isenberg et al. [10] integrated this locality property into a framework for building tabletop display interfaces. With this framework, tabletop interfaces can be created where virtual objects adhere to this property.

2D Force-Based Interaction

A significant body of tabletop literature focuses on how to move and rotate virtual objects on a digital surface. One of the overarching results of studies [13, 14, 21] involving movement and rotation is that simulating (at least to some degree) how movement and rotation happen with physical forces typically results in both improved performance and a compelling feeling of embodiment with the virtual objects.

The rotate n' translate (RNT) technique [12] for moving and rotating objects uses the metaphor of an opposing force acting on a virtual object to make it rotate while moving. This technique has also been extended so that, when let go, an object will continue along its trajectory according to the current speed of movement. This extension produces the ability to "flick" or "toss" objects across the screen [9, 10]. The TNT techniques [14] use 3DOF to more directly simulate the movement observed in studies of moving and rotating paper on physical tables. With this method, a person can place their hand or a physical block on a virtual object and the position and orientation of the hand or block controls the movement and rotation of the virtual object. On multi-touch tables, two fingers are typically used for a combined movement, rotation and scaling of a virtual object. The position of the first touch is used to determine the movement of the object and the position of the second touch relative to the first is used to determine the rotation and scale. This technique simulates how movement and rotation can occur with physical objects if frictional force between the fingers and objects is considered. The scaling aspect is an example of how this familiar force-based behaviour can invoke virtual behaviour not possible in the physical world (i.e., magically growing or shrinking objects).

ShapeTouch [4] provides force-based interactions on 2D virtual objects, such as pushing objects from the side, tossing them across the screen, peeling them back to place other objects underneath, and more. These techniques use the sensory data to invoke complex but physically familiar behaviour on the objects that are in direct contact with a person's hands and arms.

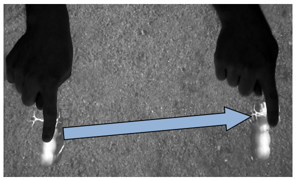
3D Force-Based Interaction

Hancock et al. [7] extended the idea of moving and rotating objects to three-dimensional virtual objects on a table. They used the same metaphor as RNT of an opposing force with a single touch point on an object. Their studies show, however, that the feeling of "picking up" an object is better approximated by using more fingers (up to three). With three fingers, people can have full control of the 6DOF of movement and rotation in a 3D environment [7]. Force-based effects such as collisions, gravity, mass, and inertia can also be integrated into 3D environments through the use of a physics engine (e.g., BumpTop [1]). The image data provided through many multi-touch input devices (FTIR [6], Microsoft Surface[1], SMART Table[2]) can be more directly integrated into such physics engines by creating physical bodies (either through proxies or particle proxies) that then can interact with the virtual objects through the physics engine [21]. Because a person's hands and fingers (or even other physical objects) have a virtual representation in the physics engine, these can be used to push other virtual objects around.

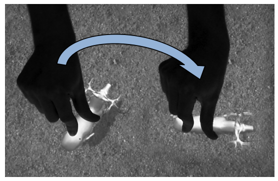
The use of forces in general and the use of physics-based forces in 3D virtual worlds in particular, have immense appeal as a basis for interaction on multi-touch tables. However, while many appealing interactions have emerged, they fall short of the full functionality required for practical applications. For instance, BumpTop [1] resorts to a symbolic gestural language, which has an associated learning curve and the need for memory retention. Wilson et al. [21] point the way to interactions that extend physical real-world responses into the virtual world, but fall short in that the realized virtual interactions provide only the ability to move invisible proxies, and not to spin, flip, or lift the virtual objects. In essence, this work provides no equivalent to an opposable thumb and has made a direct call for the ability to pick objects up and place them inside others – capabilities offered by our sticky tools approach. Another approach to manipulating 2D and 3D objects is to use the space in front of the display [11,15] to extend interaction capabilities, however, this has been only accomplished through additional hardware such as markers and vision-based systems. Sticky tools achieves all 6DOF without additional hardware.

STICKY TOOLS

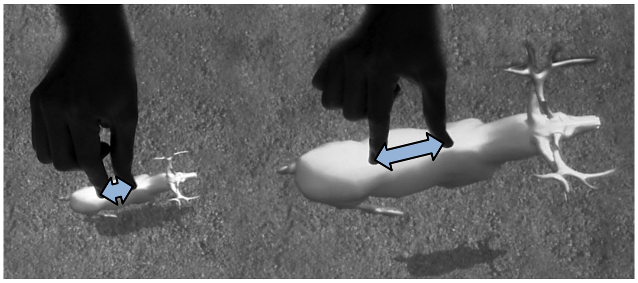
[1]Microsoft Surface. http://www.microsoft.com/surface
[2]SMART Table. http://www.smarttech.com/table



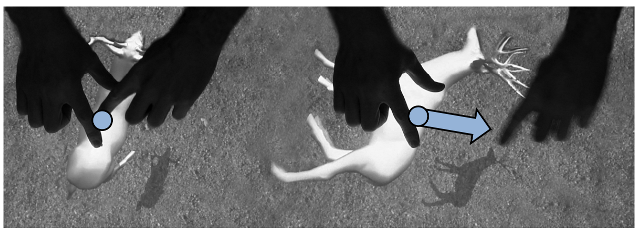
(a) Move



(b) Rotate 2D



(c) Lift



(d) Rotate 3D

Figure 2: Sticky fingers and opposable thumbs.

In this section, we introduce *sticky tools*, a combination of three concepts: *sticky fingers*, *opposable thumbs*, and *virtual tools*. We use sticky fingers and opposable thumbs to enable full control of a single object in a 3D virtual scene. We then use the virtual tools concept to take this full control of a single object and use it to enable full functionality within that system. Thus, sticky tools are a mechanism to improve upon existing force-based interaction techniques so that they can provide full functionality to a multi-touch table, without the need for symbolic gestures or menus.

Of the three significant aspects of controlling single 3D virtual objects that are discussed in the 3D interaction literature (selection, navigation, and manipulation [3]) we focus on selection and manipulation. Selection and some manipulation are available via sticky fingers, full manipulation of single objects requires the addition of the opposable thumb. The possibility of navigation is realizable with virtual tools and is discussed as future work.

Sticky Fingers

In 2D multi-touch interfaces, the two-finger move / rotate / scale interaction technique has become ubiquitous. Because one's fingers stay in touch with the virtual object in the location they initially contact, this can be referred to as a sticky-finger interaction. This perception of touching the virtual object persists through the interaction, providing some of the feedback one might expect in the physical world. The scaling action of spreading one's fingers also maintains stickiness, still providing a person with the feeling that they are controlling two points on the virtual object. However, this scale aspect would be impossible in the physical world (at least for rigid bodies), thus it combines the partial physicality of the sticky fingers with the potential for magic that the computer offers.

Sticky fingers works well in 2D, providing move (in x and y), spin (rotate about z) and scale. In 3D the first two of these capabilities can be directly mapped giving move and spin, however in 3D two additional factors are missing: lift and flip. To address this we extend the 2D sticky-fingers technique together with ideas from the three-finger technique described by Hancock et al. [7] to create a technique to manipulate 3D virtual objects rendered in perspective.

When only two fingers are used, the points of contact remain under one's fingers. Similarly to the 2D technique, as the fingers move about the display, the virtual object moves with them (Figure 2a), and as the fingers rotate relative to one another, so does the virtual object (Figure 2b). In 3D, as the distance between the fingers gets larger, the virtual object moves towards the perspective viewpoint causing the object to appear larger (Figure 2c). Thus sticky fingers in 3D provides lift. Note that the virtual object's size in the 3D model will not change, only its distance to the viewpoint.

Sticky Fingers & Opposable Thumbs

With two sticky fingers alone, one can not flip a virtual object over while maintaining the stickiness property, since the initial contact points are likely to become hidden. To flip the object about x and y, the third finger is used as relative input, providing rotation about the axis orthogonal to the direction of movement (Figure 2d). The third finger is the *opposable thumb*. Unlike actual thumbs, one can use any finger to provide the virtual flipping functionality that our opposable thumbs provide in the real world. Instead of mapping the first two fingers to move (in x and y), rotate (about z), and scale, we map them to move (in x, y, and z) and rotate (about z). The third finger is then used for rotation about x and y. This technique provides *full control* of all 6DOF, enabling behaviour such as lifting objects and flipping them over.

It is possible to maintain the stickiness property of the first two fingers when the third finger is active by using the axis defined by these two fingers as the axis of rotation. The disadvantage, however, is that movement along this axis with the third finger would not have any effect on the virtual object, and achieving the desired rotation may require defining a new axis of rotation (by lifting one's fingers and reselecting the axis with the first two fingers). This disadvantage led to the design decision to use relative interaction for the third touch for 3D rotations.

For selection, we use a combination of crossing [2] and standard picking to select the initial point of contact for each of the three fingers. Thus, the order that the fingers come in contact with a virtual object determine the points used for movement and rotation. By extension, in a multi-touch environment, where for instance the flat of one's hand or one's forearm could be interpreted as a series of touches, all objects crossed would be moved with one's hand. As a result, a person can use their fingers and arms to perform actions on multiple objects simultaneously (e.g., sweeping all objects to the side). This sweeping action relates to the sweeping actions in Wilson et al. [21] without requiring the use of the physics engine.

Virtual Tools

While together sticky fingers and opposable thumbs provide a way to select and fully manipulate a single 2D or 3D virtual object, more complex interactions, such as using an object to push another object around or changing an object's properties (e.g., density, colour) are not possible. We introduce virtual tools to enable more complex interactions on virtual objects. A virtual tool is a virtual object that can act on other virtual objects and is able to cause changes to the receiving object. Any virtual object that is controlled with sticky fingers and opposable thumbs becomes a *sticky tool*.

While virtual tools can exist in any virtual environment, we realized our virtual tools within a simulated real world by using a physics engine[3]. Thus, when a person interacts with a virtual object, it is placed under kinematic control so that other virtual objects will react physically to its movement, but the contact with the sticky fingers gives control of the object to the fingers. Thus, the object can now be used to hit other objects, but will not be knocked from the sticky contact. When the sticky tool makes contact with another object, it can cause physically familiar behaviour but these contacts can also be detected and made to invoke abstract actions, such as re-colouring the object.

The concept of sticky tools is useful in explaining previous work. The technique introduced by Wilson et al. [21] can be thought of as an example of a very simple virtual tool. Their interaction technique can be described as controlling the 2D position of many invisible virtual objects and these invisible objects interact with other objects in the scene through the use of a physics engine. In this framework the proxies can be considered to be a virtual tool whose behaviour is always to invoke frictional and opposing forces on other virtual objects. Similarly, the joint technique used in BumpTop [1] allows 3D icons to act as virtual tools that cause collisions that invoke behaviour on other 3D icons.

Table 1 shows a comparison of the features of joints (J), proxies (P), sticky fingers (SF), and sticky fingers with opposable thumbs (SF+OT). They are compared on many commonly provided multi-touch interactions. Sticky fingers and opposable thumbs offer a more complete set of these interactions than any other; however, all have some gaps and this is not a complete list of all possible functionality. For any of these approaches the gaps can be addressed by virtual tools. That is, with virtual tools the functionality of any of the unchecked cells in Table 1 can be enabled. For example, sticky fingers and opposable thumbs can use a virtual tool to push or surround other objects. This is also true for the joints technique or sticky fingers alone (without opposable thumbs). A virtual tool could be used in combination with either the joints technique or the proxies technique to lift objects in the third dimension. For example, a platform could be introduced that objects could be moved onto. The platform could then be used to lift the objects through use of a dial, a slider, or elevator virtual object. Similar virtual objects could also be imagined that could enable flipping and spinning of virtual objects. Virtual tools also offer new potential for additional functionality not possible with any previous single technique.

[3]NVIDIA Corporation. http://nvidia.com/physx

Feature	J	P	SF	SF+OT	ST
Lift (move in z)			✓	✓	✓
Drag (move in x & y)	✓	✓	✓	✓	✓
Spin (rotate about z)	✓		✓	✓	✓
Flip (rotate about x / y)				✓	✓
Push		✓			✓
Toss	✓		✓	✓	✓
Surround (contour)		✓			✓
Additional Points		✓			✓
Usable with Fingers	✓	✓	✓	✓	✓
Usable with Objects		✓			✓

Table 1: Comparison of different techniques for interacting with 3D virtual objects on a table.

UNDERSTANDING VIRTUAL OBJECTS

In essence, the difference between the use of virtual tools and previous techniques comes down to the ability to assign richer meaning to virtual objects. This assignment of meaning is analogous to a similar discussion introduced by Underkoffler and Ishii [20] for their luminous tangible system. They showed how tangible objects could be assigned richer meaning to expand interaction possibilities. We parallel their discussion on luminous tangible object meanings with a discussion on virtual tool object meanings. The discussion on virtual tool meanings is followed by a generalized model of how force-based interaction can be used to provide all this functionality by changing the complexity of how people control both physical and virtual objects, as well as how those physical and virtual objects can control each other.

Virtual Object Meanings

In this section we provide examples of how using virtual objects to control other virtual objects can enrich interaction. We demonstrate this richness by mirroring Underkoffler and Ishii's [20] description of how tangible objects can take on different object meanings along the spectrum:

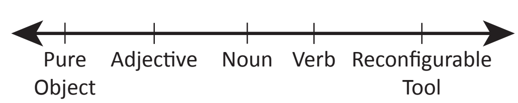


Figure 3: The spectrum of object meanings used in Urp [20] to describe tangible devices.

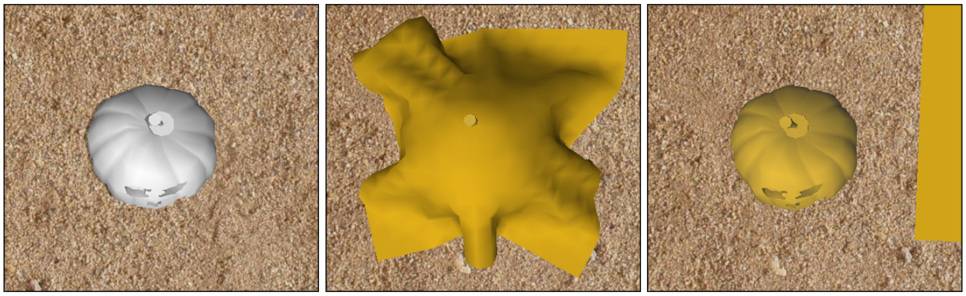


Figure 4: Virtual objects as verbs.

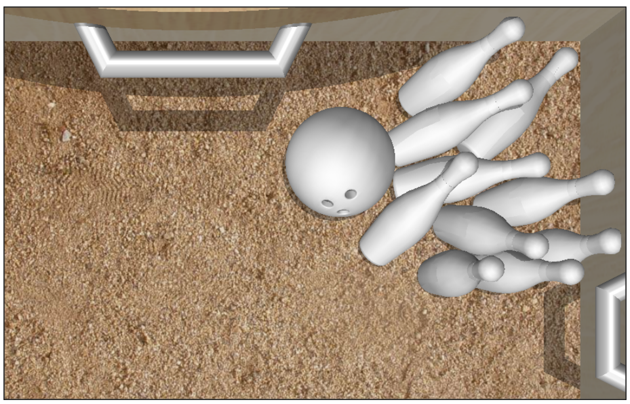


Figure 5: Virtual objects as nouns.

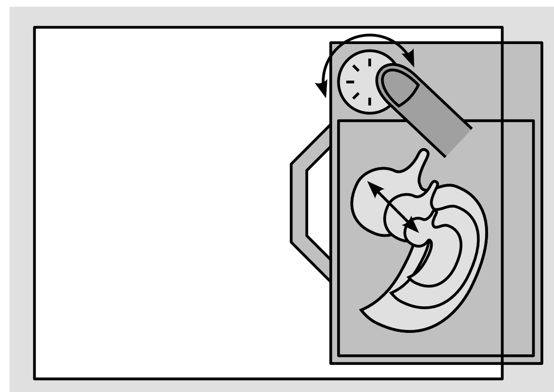


Figure 6: Virtual objects as reconfigurable tools.

In each of the following subsections, we first state the definition used by Underkoffler and Ishii to describe the different object meanings (with modifications so that they describe a virtual environment, instead of a tangible system) and then describe an example of a sticky tool whose meaning can be interpreted using this definition. We thus show that sticky tools enable virtual objects to take on all of the possible meanings of tangible luminous objects.

Virtual Objects as Nouns

"These objects occupy the center of the axis and are likely the most obvious in their behavior. They are fully literal, in the sense that they work in their [virtual] context very much the way objects 'operate' in the real world—an Object As Noun exists in our applications simply as a representation of itself: an immutable thing, a stand-in for some extant or imaginable part of the real-world." [20, p. 392]

A virtual object as a noun stands for itself—in the virtual world, that is, for what it appears to be. Thus, if it looks like a ball it should behave like a ball. In a virtual 3D environment, we can render any mesh of triangles that has been modeled. Thus, rigid bodies of virtually any shape can be added to the environment and made to interact with other rigid bodies using the physics engine. Thus, these virtual objects can operate in the virtual world in a way that is similar to how they behave in the real world. For example, a set of bowling pins in the environment can be knocked over using a virtual bowling ball (Figure 5).

Virtual Objects as Verbs

"As we move to the right along the continuum, away from Object As Noun, inherent object meaning is progressively abstracted in favor of further—and more general—functionality ... It is not understood as 'present' in the [virtual] world ... but exists to act on other components that are, or on the environment as a whole." [20, p. 392]

A virtual object as verb exists as a virtual object but embodies actions. That is the appearance of the object symbolizes the possibility of an action. In our virtual environment, we include a cloth that embodies 'wrapping' (Figure 4). Dropping a cloth on another object wraps that object. We leave evidence of this wrapping by changing the affected object's colour, providing a way to colour objects. The act of covering another virtual object with a cloth can be further abstracted to provide a variety of different functions. We also provide a lamp sticky tool that embodies the actions of shedding light, casting shadows, and can be used as the sundial in Urp to simulate changing the time of day. This sticky tool differs from the tangible device in Urp in that the lamp can be made to disobey the law of gravity and to pass through other objects in the environment.

Virtual Objects as Reconfigurable Tools

"This variety of object-function is fully abstracted away from 'objecthood', in a way perhaps loosely analogous to a GUI's mouse-plus-pointer." [20, p. 392]

A virtual object as a reconfigurable tool is an object that can be manipulated to affect other objects. It does not stand for

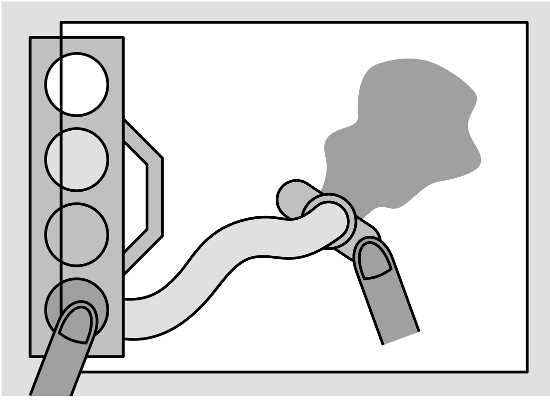


Figure 7: Virtual objects as attributes.

itself as a noun, or imply an action as a verb, but instead symbolizes a functionality. We create a compound sticky tool consisting of a drawer object and a dial (Figure 6). When a figurine is placed inside this drawer, the dial can be rotated to grow or shrink the figurine. This compound sticky tool could be reconfigured to perform any action that involves changing a one-dimensional property of another virtual object along a continuous axis. For example, it could be used to change an object's density or elasticity.

Virtual Objects as Attributes

"As we move to the left away from the center of the axis, an object is stripped of all but one of its properties, and it is this single remaining attribute that is alone considered by the system." [20, p. 392]

A virtual object as attribute represents one and only one of its attributes. For an example, we create another compound sticky tool for painting the background of the environment (Figure 7). This sticky tool includes a group of four buckets that each contains a different texture and a hose that extends from below the buckets. In the case of the bucket, the only attribute that matters is its texture. The shape, size, density, location and all other attributes are abstracted from this virtual object. To paint the background a person selects a bucket with one finger to activate the hose and then, with the other hand, can move the hose's nozzle to indicate the area of the background to paint. Movement in the z-direction affects the area of influence of the hose (the farther from the background, the larger the radius of influence). Touching the texture bucket activates the texture that flows along the hose into the environment.

Virtual Objects as Pure Objects

"This last category is the most extreme, and represents the final step in the process of stripping an object of more and more of its intrinsic meanings. In this case, all that matters to a [virtual] system is that the object is knowable as an object (as distinct from nothing)." [20, p. 392]

A virtual object as pure object is a symbol and stands for something other than itself. We create a sticky tool that allows the storage of the locations of all of the figures in a scene to be symbolized by a pure virtual object. Which virtual object will perform this symbolic function is established by placing an object in a "save" drawer. Thereafter, the scene is essentially stored in this virtual object and can be reloaded by placing that same figure in the empty environment. Thus, any virtual object can be stripped completely of its intrinsic meaning, and the locations of the remaining virtual objects can be (e.g.) "put in the dinosaur". That is, the dinosaur now stands for the scene.

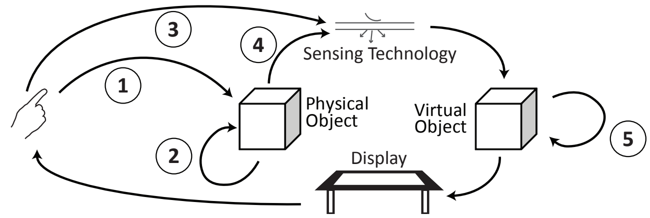


Figure 8: A diagram of the five major components of force-based interaction.

Force-Based Interaction

We have introduced sticky fingers, opposable thumbs, and virtual tools as 3D tabletop interaction concepts and discussed them in relation to joints, proxies, particle proxies, and tangible devices. In this section, we generalize from these approaches to provide a framework that encompasses these techniques and indicates how existing functionalities in force-based interactions can be expanded.

Using physical forces to control virtual objects has the appeal of being easy to understand and learn due to our ability to transfer knowledge from our experience in the physical world. However, in order to simulate physical behaviour in the digital world, two primary components are required: a sensing technology, and a display technology. The sensing technology takes actions from the physical world and translates them into messages that can be understood by the computer, and the computer can then translate those messages into something virtual that can be understood as a physical reaction to the initial action.

In sensing and translating this information, there are several places that the complexity of the force-based action-reaction can vary. First, new sensing technologies can be invented to be able to identify more and more complex physical forces. Essentially, the computer can become better at understanding how people control physical objects (in multi-touch, through a person's fingers or in tangible, through a person's use of a physical object). Second, as seen in our sticky tools, Wilson et al. [21] and Agrawala et al. [1] the mapping from what is sensed to the system response can be made to include complex physics algorithms that better simulate real-world responses. Third a largely unexplored possibility is the introduction of complexity through how the system's response propagates in the virtual environment. That is, virtual objects can control other virtual objects.

These real and virtual interaction possibilities can be summarized by: (1) *people controlling physical objects*, (2) *physi-*

cal objects controlling physical objects, (3) *people controlling virtual objects*, (4) *physical objects controlling virtual objects*, and (5) *virtual objects controlling virtual objects*. Figure 8 provides a diagram of this space. The first four aspects have been well researched; our introduction of virtual objects, that we have called sticky tools, controlling virtual objects is one of the contributions of this paper.

People Controlling Physical Objects

We are accustomed to interacting in the physical world and are adept at controlling physical objects with fine motor movements. It can be advantageous to leverage these natural abilities when creating computer interfaces. Although the interactions that are available in the physical world are arguably highly complex, they are familiar. From the interaction design perspective, this familiarity makes it easier to predict what a person might expect. For example, the designer might predict that people will expect large objects (i.e., objects with more mass) to require more force to push than smaller objects.

Physical Objects Controlling Physical Objects

Some tasks require more precision or more power than most people's physical abilities. For these tasks, we use tools (hammers, levers, needles). Good design can make a tool that can be both useful and usable [16], making it possible for people to extend their physical capabilities.

People Controlling Virtual Objects

To enable a person's actions to cause reactions in the virtual world, human movement must be sensed in some way. This sensed information has typically been mouse movement or a key press, though many more complex devices exist [5, 23]. The sensed information can be then used to cause a virtual action. However, the necessary translation can interfere with familiarity. The interactions we learned from the physical world may not predict virtual interactions. However, reintroducing the familiarity of physicality is considered a positive goal, which with current massively multi-touch capabilities seems increasingly possible. A variety of techniques now approach this goal [4, 8, 12, 14, 17, 18] and recently have been extended to provide more physically realistic interactions with 3D virtual objects [7, 21].

Physical Objects Controlling Virtual Objects

Tangible computing focuses on how physical objects can be used to control virtual objects. This line of research suggests that the richness of interactions with physical objects can be leveraged by using them directly in an interface. The use of physical objects to control virtual objects relates to the human tendency to use physical tools to control other physical objects. Tangible computing devices can be seen as a form of tool that provides a mechanism for designers to introduce complexity. One of the disadvantages of using physical objects to control virtual objects is the need to sense the behaviour of the physical object.

Virtual Objects Controlling Virtual Objects

As noted in our discussion of virtual tools, virtual objects can be used to control other virtual objects. Since they are virtual, they are already fully described computationally. These virtual objects can then be assigned meaning in the same way that tangible devices can; they can be used as pure objects, attributes, nouns, verbs, and reconfigurable tools.

Interface components in general are virtual tools, however, the simulation of physical forces implies that virtual objects could have the capabilities of real physical objects—where objects behave as themselves (nouns) and can be used to act (verbs). The use of virtual objects to control other virtual objects expands the methods for creating create complex interactions. One direction is to leverage people's familiarity with physical objects through use of the software support of a physics engine. Another is to take advantage of the fact that, in a virtual world, physical laws do not have to be obeyed. These two ends of an interaction spectrum of course have rich possibilities of combinations. An important factor in the potential of virtual tools is that since these objects are already virtual there is no need for a sensing interface between the action and reaction. Now that there is a simple interface between a person and a virtual object in a 3D forced-based environment the possibility of exploring the potential of virtual tools is open.

CONCLUSIONS

In this paper, we have presented the first method that provides full 3D interaction that has the benefits of force-based interaction. Sticky tools—complete with full 6DOF manipulation—allow people to pick objects up, place them in other objects, and use these virtual objects as tools. Using virtual objects as tools opens up the possibilities of practical actions, thus providing more complete functionality to tabletop interfaces. These interaction capabilities require only multi-touch input such as that provided by the Microsoft Surface and the SMART Table. That is, they provide full 3D interaction while one's fingers stay in contact with the table's surface and thus with the virtual object.

We have taken this concept of sticky tools and demonstrated its potential for generating whole families of interaction approaches by describing a parallel to Underkoffler and Ishii's [20] luminous tangible system. By showing that sticky tools as virtual objects can also be understood as nouns, verbs, reconfigurable tools, attributes, and pure objects, we show that sticky tools can enable richer meaning in tabletop interfaces.

By comparing our sticky fingers and opposable thumbs with joints and proxies, we demonstrate how the addition of virtual tools to any of these approaches can provide full functionality. With the addition of virtual tools, the limitations of these techniques can be overcome; they can be enhanced with the ability to push, surround, toss, flip, lift, and more, where they were otherwise limited. In our discussion of force-based interaction, we provide a framework for introducing rich interactions to tabletop computing via virtual tools. Designers can use this framework to enable rich interactions for applications with more complete functionality.

ACKNOWLEDGEMENTS

We would like to thank Natural Science and Engineering Research Council of Canada, Albertas Informatics Circle of

Research Excellence, Alberta Ingenuity, and the Canadian Foundation of Innovation for research support. We also thank the reviewers and members of the iLab for their helpful comments on this work.

REFERENCES

1. Agarawala, A. and Balakrishnan, R. Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. In *Proc. CHI*, ACM (2006), 1283–1292.
2. Apitz, G. and Guimbretière, F. Crossy: a crossing-based drawing application. In *Proc. UIST*, ACM (2004), 3–12.
3. Bowman, D. A., Kruijff, E., LaViola, J. J., and Poupyrev, I. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004.
4. Cao, X., Wilson, A. D., Balakrishnan, R., Hinckley, K., and Hudson, S. E. Shapetouch: Leveraging contact shape on interactive surfaces. In *Proc. Tabletop*, IEEE Computer Society (2008), 129–136.
5. Fröhlich, B., Hochstrate, J., Skuk, V., and Huckauf, A. The globefish and the globemouse: two new six degree of freedom input devices for graphics applications. In *Proc. CHI*, ACM (2006), 191–199.
6. Han, J. Y. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proc. UIST*, ACM (2005), 115–118.
7. Hancock, M., Carpendale, S., and Cockburn, A. Shallow-depth 3D interaction: Design and evaluation of one-, two- and three-touch techniques. In *Proc. CHI*, ACM Press (2007), 1147–1156.
8. Hancock, M. S., Carpendale, S., Vernier, F. D., Wigdor, D., and Shen, C. Rotation and translation mechanisms for tabletop interaction. In *Proc. Tabletop*, IEEE Computer Society (2006), 79–88.
9. Hinrichs, U., Carpendale, S., and Scott, S. D. Evaluating the effects of fluid interface components on tabletop collaboration. In *Proc. AVI*, ACM (2006), 27–34.
10. Isenberg, T., Miede, A., and Carpendale, S. A buffer framework for supporting responsive interaction in information visualization interfaces. In *Proc. C5*, IEEE Computer Society (2006), 262–269.
11. Keijser, J., Carpendale, S., Hancock, M., and Isenberg, T. Exploring 3D interaction in alternate control-display space mappings. In *Proc. 3DUI* (2007).
12. Kruger, R., Carpendale, M. S. T., Scott, S. D., and Tang, A. Fluid integration of rotation and translation. In *Proc. CHI*, ACM Press (2005), 601–610.
13. Kruger, R., Carpendale, S., Scott, S. D., and Greenberg, S. Roles of orientation in tabletop collaboration: Comprehension, coordination and communication. *J. CSCW 13, 5-6* (2004), 501–537.
14. Liu, J., Pinelle, D., Sallam, S., Subramanian, S., and Gutwin, C. TNT: improved rotation and translation on digital tables. In *Proc. GI*, CIPS (2006), 25–32.
15. Malik, S., Ranjan, A., and Balakrishnan, R. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In *Proc. UIST*, ACM (2005), 43–52.
16. Norman, D. A. *Psychology of everyday things*. Basic Books, New York, NY, 1998.
17. Shen, C., Lesh, N., and Vernier, F. Personal digital historian: story sharing around the table. *Interactions 10, 2* (2003), 15–22.
18. Shen, C., Vernier, F. D., Forlines, C., and Ringel, M. DiamondSpin: an extensible toolkit for around-the-table interaction. In *Proc. CHI*, ACM (2004), 167–174.
19. Ståhl, O., Wallberg, A., Söderberg, J., Humble, J., Fahlén, L. E., Bullock, A., and Lundberg, J. Information exploration using the pond. In *Proc. CVE*, ACM (2002), 72–79.
20. Underkoffler, J. and Ishii, H. Urp: a luminous-tangible workbench for urban planning and design. In *Proc. CHI*, ACM (1999), 386–393.
21. Wilson, A. D., Izadi, S., Hilliges, O., Garcia-Mendoza, A., and Kirk, D. Bringing physics to the surface. In *Proc. UIST*, ACM (2008), 67–76.
22. Wu, M. and Balakrishnan, R. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *Proc. UIST*, ACM (2003), 193–202.
23. Zhai, S. User performance in relation to 3D input device design. *SIGGRAPH Comput. Graph. 32, 4* (1998), 50–54.