

Eden: A Professional Multitouch Tool for Constructing Virtual Organic Environments

Kenrick Kin^{1,2}Tom Miller¹Björn Bollensdorff³Tony DeRose¹Björn Hartmann²Maneesh Agrawala²¹Pixar Animation Studios
Emeryville, CA, USA²University of California, Berkeley
Berkeley, CA, USA³Technische Universität Berlin
Berlin, Germany

ABSTRACT

Set construction is the process of selecting and positioning virtual geometric objects to create a virtual environment used in a computer-animated film. Set construction artists often have a clear mental image of the set composition, but find it tedious to build their intended sets with current mouse and keyboard interfaces. We investigate whether multitouch input can ease the process of set construction. Working with a professional set construction artist at Pixar Animation Studios, we designed and developed *Eden*, a fully functional multitouch set construction application. In this paper, we describe our design process and how we balanced the advantages and disadvantages of multitouch input to develop usable gestures for set construction. Based on our design process and the user experiences of two set construction artists, we present a general set of lessons we learned regarding the design of a multitouch interface.

Author Keywords

eden, multitouch, object manipulation, camera control, gestures, gesture design, set construction

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces—*Graphical user interfaces, Interaction styles*

General Terms

Design, Human Factors

INTRODUCTION

The production of computer-animated feature-length films, such as Pixar's *Toy Story* and DreamWorks' *How to Train Your Dragon*, consists of many distinct stages, commonly referred to as the production pipeline. One of these stages is the construction of virtual sets. Similar to a physical set for live-action films, a virtual set is the environment in which animated films are shot. Set construction artists select and position geometric models of objects, such as furniture and props to build manmade environments, and vegetation to build organic environments.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.

Copyright 2011 ACM 978-1-4503-0267-8/11/05...\$10.00.

Today, many animation studios use off-the-shelf modeling and animation packages (e.g. Maya, 3ds Max) for set construction. Despite more than a decade of interface refinement, the process required to build a set using these mouse and keyboard interfaces is long and tedious. An artist commonly places hundreds if not thousands of 3D objects in the set, but is usually limited to placing one object at a time. Moreover, to properly place a single object in 3D space, the artist often performs several individual 3D manipulations, such as translation, rotation, and scale. However, the mouse only has two degrees of freedom, so the artist cannot manipulate more than two spatial parameters of the object at a time. In addition, existing interfaces introduce significant overhead: the artist must manage modes, select small manipulators, and traverse long distances with the mouse.

In this work we investigate whether *Eden*, a new organic set construction application that leverages multitouch input, can address these concerns. We focus on direct-touch multitouch workstations, which support the use of two hands on a screen where display and input are co-located. With two hands, the artist can work in two different parts of the screen at the same time, thereby reducing the need to travel back and forth between spatially distant screen regions. The artist may also become more efficient by performing simultaneous operations, one with each hand. Furthermore, multitouch workstations can sense the position of each finger and thus two hands provide many degrees of freedom of input. Multitouch interfaces can use these many degrees of freedom to allow users to specify both target object and operation, while manipulating more than just two of the object's spatial parameters at a time. As a result, the application can reduce the number of modes and the number of individual steps needed to complete the placement of a single object.

Despite these advantages, building a multitouch application presents design challenges, such as choosing gestures that are efficient, memorable, and comfortable to perform. There are many different ways to map multitouch sensor data to operations, and the best gesture for a given task is often not obvious. An application might also require a large set of operations, and in order for the application to unambiguously interpret the user's actions, no two gestures can be the same. Finally, touch input has several drawbacks that can reduce the user's efficiency, including imprecision due to the fat finger problem [25] and occlusion of content by the hands [30].

To address these challenges, we built *Eden* while working in close collaboration with our co-author (TM), a veteran

set construction artist at Pixar Animation Studios. We relied on his feedback and experience to create a set construction application suitable for professional-level use. From our design process, we found that restricting Eden to support one operation at a time allowed us to design simple, easy to learn gestures that split the workload across two hands. Using Eden, TM has built a set for an upcoming feature film, and found the system to be more efficient and more pleasant than his current toolset. We believe the general lessons we learned from our design process and the evaluations by TM and a novice user will be informative to researchers and application developers designing multitouch applications for professional users.

RELATED WORK

Prior research has examined new interface techniques for professional tasks. Such case studies include: *Proteus* – a personal electronic journal [9], *ButterflyNet* – a digital field journal for biologists [41], and *I Love Sketch* – a sketching interface for creating 3D curve models [2]. Similarly, we investigate a new multitouch interface technique for professional set construction. We focus on two main areas of related work.

Multitouch Interactions

Researchers have done extensive work on the use of multitouch devices in recent years. They have explored the utility of different attributes of touch, such as shape [7] and orientation [32], as well as simulating physics with touch [35].

Other researchers such as Wu et al. [38, 39] have developed task-oriented multitouch applications including a room planning application to investigate multi-user interaction and an annotation application to investigate gesture design. Brandl et al. [5] developed a sketching application to investigate touch and pen interaction. However, these applications primarily served as testbeds for multitouch interaction design. Researchers have also deployed applications designed for casual users outside of a lab setting, including a senior citizens center [12] and a public city center [24].

Few researchers have explored the use of multitouch for producing professional work. One notable exception is the work of Wigdor et al. [34], which investigated the long term use of a multitouch workstation for office-related tasks. However, the authors used multitouch as a mouse emulation device for pre-existing mouse and keyboard interfaces, whereas we designed and implemented a professional-level set construction application specifically for a multitouch workstation.

Researchers have also recently examined user-defined gestures. Wobbrock et al. [36] combined gestures designed by 20 end-users to create a gesture set for 22 commonly used commands. Follow-up work by Morris et al. [21] found that users preferred gestures designed by end-users and researchers over those designed by researchers alone, seemingly because researchers proposed more physically and conceptually complex gestures than end-users. Thus, we designed our gestures with the help of a veteran set construction artist, one of our target users.



Figure 1. An organic set in Pixar's *Up*. Copyright Disney/Pixar.

3D Object Manipulation

Research in object manipulation and set construction has a long history, but most prior work has developed techniques to improve the mouse and keyboard interface for object manipulation by using constraints, pseudo-physics, and semantic information [6, 16, 23, 29, 40, 42]. In contrast, we examine whether a completely new multitouch interface is a better alternative to the mouse and keyboard interface.

The mouse's limited degrees of freedom (DOF) has motivated research on input devices with more DOFs. Higher DOF input devices such as the *Data Glove* [43], the *Bat* [33], *GlobeFish* and *GlobeMouse* [11], and the commercially available *SpaceBall* were developed for object manipulation. Multitouch workstations also provide many input DOFs, and we designed our application specifically for this input.

In more recent work, Hancock et al. [15] and Reisman et al. [27] investigated using multitouch for object manipulation. These researchers leveraged direct-touch and the many DOF input of multitouch for this isolated task. In addition, Cardinaels et al. [8] developed a multitouch application for conceptualizing scenes for television productions. Their application, however, was designed for pre-visualization, while our application is designed for final production.

ORGANIC SET CONSTRUCTION

Set construction is the process of selecting and positioning virtual objects to build a virtual environment inhabited by characters of a computer-animated movie. Before building a set, the set construction artist first works with the story and art departments to determine the aesthetics and rough layout of the set. Then the set construction artist works with the layout department, which is responsible for placing the foundation of the set by positioning the terrain and any key architectural elements that help dictate the action in the scene. The set construction artist then populates the sets with the geometric objects built by the modeling department to flesh out the world. The layout department also provides the set construction artist with shot cameras, which are used to make the final renders. Using the shot cameras, the set construction artist "constructs to camera," to avoid building sections of the set that will not be seen in the final film. Throughout this process, the set construction artist continues to work iteratively with the story, art, and layout departments to fi-

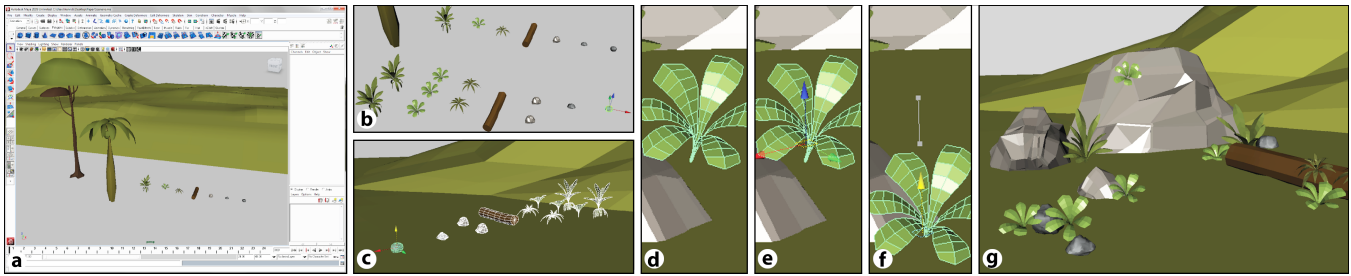


Figure 2. Constructing a set with Maya: a) The set artist creates a model catalog by lining up the models he plans on using away from the terrain. b-c) He then makes duplicates of the objects and translates them to the region of the terrain where he is constructing the set. d-f) To translate an object he first selects the object, then switches to translation mode with a hotkey, and finally picks and drags the arrow manipulator. g) He translates, rotates, and scales objects one by one until he completes the set.

nalize the set. Once the director approves the set, it is then sent to the animation department.

To gain a better understanding of the set construction process, we observed TM, who has over 10 years of set construction experience at Pixar Animation Studios. TM specializes in building organic sets, such as forests and parks, and other outdoor environments consisting primarily of vegetation (Figure 1).

To build a set, TM traditionally uses Autodesk Maya [1], a 3D modeling and animation package. His workflow, whether for manmade or organic sets, typically proceeds as follows (Figure 2). First, TM loads the objects he plans to use for a set and lines them up in a location away from the terrain. These objects serve as his *model catalog*. To add objects to the set he duplicates them in the model catalog area and moves them into the region of the set he is working on. Then using the Maya translation, rotation, and scale manipulators, he positions and orients each object into place. To translate an object, for example, he selects the object, hits the ‘W’ hotkey to enter translation mode, and picks the appropriate arrows on the translation manipulator (Figure 2e,f) to drag the object into position. He can inspect the set by using the default Maya camera controls: while holding the ‘alt’ key, a left mouse button drag performs arcball rotation, a middle mouse button drag translates the camera along the view plane (truck and pedestal), and a right mouse button drag moves the camera forward and back (dolly). He also uses the shot cameras to construct to camera. He repeats this process, working region by region, until he completes the set.

Our original intent was to build a multitouch application for general set construction. However, we found that the imprecision of touch makes the construction of manmade sets particularly difficult. Manmade environments are often structured and rigid. They contain highly regularized elements like furniture arrangements, books on a shelf, or city streets. The positions and orientations of objects often depend precisely on the positions and orientations of other objects. Placing these objects requires precision and fine-tuning, which is problematic as touch is imprecise and the artist’s hands can obscure the content being manipulated.

Instead we chose to first target organic set construction, since it is less affected by precision issues. According to TM, he

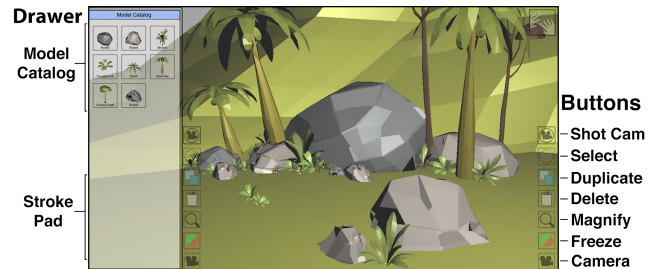


Figure 3. The interface of Eden consists of the main content view, a drawer containing the model catalog and stroke pad overlaid on the left, and two matching columns of buttons.

is less concerned about precision when constructing organic sets because he places vegetation coarsely compared to manmade objects. In addition, he often places a large amount of vegetation in an organic set, so he can frequently make use of the fast coarse targeting of direct-touch [10, 18, 28] to indicate the positions of vegetation. The experience we gain from designing a multitouch application for organic set construction might help us with the more involved task of designing a multitouch application for building general sets.

EDEN

The interface of Eden (Figure 3), our multitouch set construction application, is composed of a main view, a virtual drawer, and two columns of buttons. The main view presents the scene through a perspective camera and the artist can directly manipulate objects through this view. We designed the view to take up virtually the entire screen to help keep the artist’s focus on the content. On the left side of the interface is the *drawer*, which houses the model catalog and the stroke pad. The model catalog consists of objects available to the artist for a given session. On the stroke pad, the artist can draw single-stroke symbols that execute infrequent commands. If the artist wants to maximize the content area, he can slide the drawer closed. In addition, we provide two matching columns of buttons that map to additional set construction commands. We repeat the buttons on both sides of the interface to allow either hand to invoke them.

TM’s process for building a set with Eden typically proceeds as follows (Figure 4): TM starts a new session by loading the terrain and key architectural elements provided by the layout department into the set. He then creates a model catalog

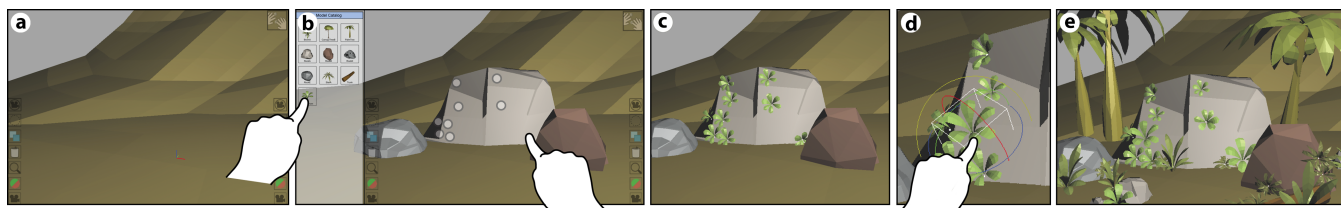


Figure 4. Constructing a set with Eden. a) The set construction artist starts with the empty terrain. b-c) Using the model catalog in the drawer, the artist can touch one finger on the model, and with a second hand touch the locations for where to place copies of the model. He taps several times on the boulder to quickly add nine bromeliads. d) He makes additional adjustments to each bromeliad by performing an arcball rotation for example. e) He continues adding and manipulating objects until the set is complete.

by drawing an ‘L’ in the stroke pad to open a panel, from which he chooses the geometric objects he wants to add to the model catalog. After building the catalog, he adds objects into the set. He might touch a tree in the model catalog and make multiple taps on the terrain to indicate the locations at which to plant each tree. If he is dissatisfied with how a tree looks he can translate, rotate, or scale the tree by performing the corresponding gesture, which we describe in the object manipulation section. In addition to using the default camera to inspect the quality of the set, TM also loads in shot cameras via a stroke command so he can construct to camera by checking the quality of the set through the shot cameras’ views. TM continues to place objects and adjust them until he is satisfied with the set.

Design Principles

Our multitouch workstation can sense the positions of the artist’s ten fingers, providing many degrees of freedom of input. Our challenge is to design gestures that map these degrees of freedom to operations and their parameters. To help us design gestures for object manipulation and camera control, we developed several design principles:

Use simple gestures for frequently used operations

Gestures that require fewer touches and fewer movements require less coordination and are faster to perform. We bind such simple gestures to the more frequently used operations to increase overall efficiency.

Conjoined touch as a modifier

To increase the size of the gesture space while keeping gestures simple, we introduce the *conjoined touch* into our gestures. A one-touch is a standard touch where a single finger touches the screen and yields a single 2D contact point. We detect a conjoined touch whenever two touches are adjacent to each other. Specifically, the two touches are combined into a single instance of a conjoined touch where the centroid of the two touches serves as the 2D contact point for the conjoined touch. Thus, two fingers on the same hand can represent three static states: one-touch, a pair of one-touches, and a conjoined touch (Figure 5). We can use a conjoined touch instead of a one-touch to differentiate two operations similar in function, while maintaining the same underlying motion of the hands.

One operation at a time

We initially designed one-handed gestures for object manipulation so the artist could perform two operations simulta-

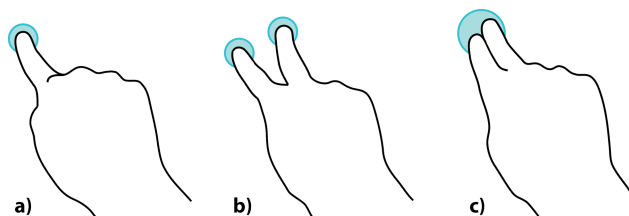


Figure 5. a) One-touch using a single finger. b) Two one-touches using two fingers. c) Conjoined touch using two fingers next to each other.

neously, one with each hand. However, we found that TM concentrates on manipulating a single object at a time and seldom requires the ability to manipulate two objects at a time. According to Raskin [26], a person only has a single locus of attention, and thus can only focus on the position of one object at time, making the simultaneous manipulation of two objects mentally difficult. Moreover, allowing only one operation at a time reduces the ambiguity of interpreting touch input. For instance, if we had permitted simultaneous gestures, then the application could interpret two touches as either two simultaneous one-touch gestures or a single gesture that uses two touches.

Split touches across both hands

Since we only support one manipulation at a time, we split the touches of a single gesture across both hands for two reasons. First, fingers on separate hands are not constrained by the palm, which makes them more mobile than fingers on the same hand. This increased mobility makes performing complex motions easier and more comfortable. Second, assigning touches to a second hand can reduce the amount of occlusion of the object being manipulated as the second hand can perform movements in an indirect fashion away from the object.

Use at most two fingers from each hand

Although a single hand supports up to five touches, anatomical constraints of the hand limits the flexibility of each touch. For example, the middle and index fingers on the same hand cannot move arbitrarily far apart. The more fingers a gesture requires, the more complicated and uncomfortable the gesture can become. Therefore, we designed gestures that limited the number of fingers used to at most two per hand.

Interchangeability of hands

For bimanual interaction, Guiard assigns fixed roles to the hands in his Kinematic Chain Model [13]: the non-dominant

hand sets the frame of reference while the dominant hand performs the primary action. We, however, permit the artist to begin an operation with either hand. Since an object can be located anywhere on the screen, interchangeability of the hands allows the artist to choose the most convenient hand to manipulate an object.

Motion of gesture reflects the operation

If the motion of the gesture is similar to the effect of the operation, then the artist can more easily guess how the gesture will affect the target object. Also, the association between motion and operation can help the artist recall gestures.

Combine direct and indirect manipulation

An attractive quality of performing direct manipulation with direct-touch is the sensation of moving a virtual object as one would in the physical world [27]. However, including indirect manipulation can improve efficiency. Using indirect manipulation, the artist can perform movements away from the target object. As a result, the artist does not need to select the object with the manipulating hand and thus the hands occlude less of the object.

Control at most two spatial parameters at a time

We had intended to design gestures that allow an artist to manipulate more than two spatial parameters of an object at a time. However, TM prefers having more individual control of these spatial parameters, so each of our gestures controls just one or two spatial parameters of an object. Research has also shown that even with a six degree of freedom input device, users perform translation and rotation separately [20].

Object Manipulation

Eden supports eight operations for object manipulation that utilize two types of touches: one-touch and conjoined touch. In Eden, the world is oriented such that the x and y axes correspond to the horizontal ground plane, and the z-axis corresponds to the up direction. As shown in Figure 6, the object manipulation operations and gestures consist of:

- x-y translation – a conjoined touch on the object, then drag
- z translation – a conjoined touch on the object, together with a one-touch drag up and down
- arcball rotation – a one-touch on the object, then drag
- local z rotation – a one-touch on the object, together with a second one-touch drag left and right
- world z rotation – a one-touch on the object, together with a conjoined touch drag left and right
- uniform scale – a one-touch on the object, together with a two-touch pinch
- one-dimensional scale – a one-touch on the object, together with a conjoined touch drag on the bounding box face perpendicular to the local scaling axis
- throw-and-catch – a one-touch on the object, and a second one-touch tap at another location

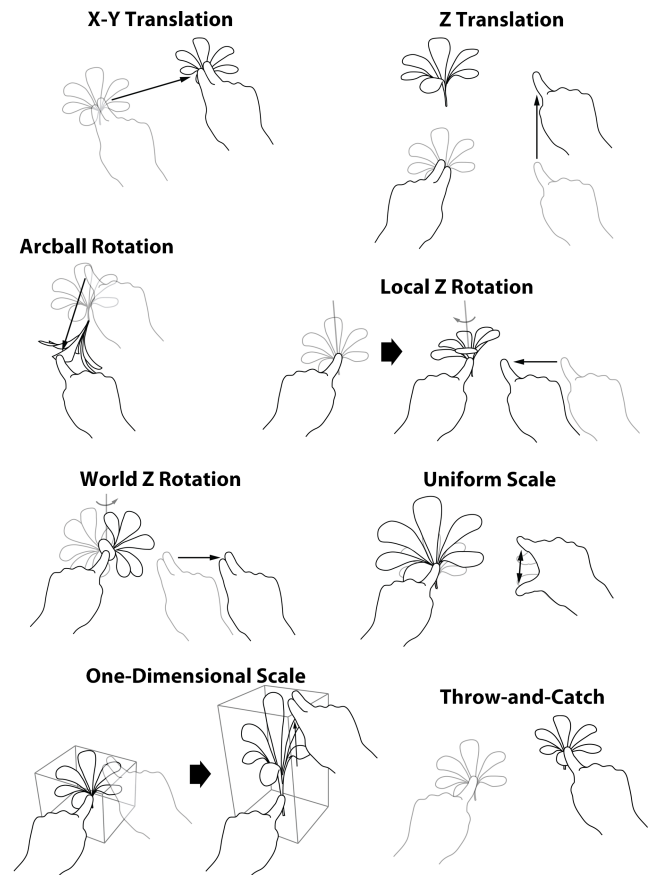


Figure 6. Set of object manipulation gestures.

We tailored the set of operations for organic set construction to support the operations most useful to TM. A set construction artist typically uses separate x and y translations to carefully align manmade objects with each other. For organic objects, however, TM finds controlling both of these translational degrees of freedom at the same time to be more efficient. Thus, we support simultaneous x-y translation, instead of separate x and y translations. We also provide a separate z translation to give TM full 3D positional control.

In addition to positioning each object, TM also adds variation to each object by rotating and scaling it. For example, he can build a grove of oak trees replicating just one oak tree model, and rotate and scale each copy to make it appear different from the other trees. TM needs just enough rotational control to tilt each object off the world z-axis and spin it about its local z-axis to make the object appear unique. Therefore, arcball rotation and z rotation are sufficient for specifying the orientation of an organic object. For some objects such as rocks that do not have a natural orientation, we provide world z rotation. We also include both uniform and one-dimensional scaling along the object's local axes, to provide additional methods to add variation to an object.

To help TM transport objects across long distances, we provide the *throw-and-catch* operation. Mouse-based interfaces often require dragging to transport an object from one loca-

tion to another, as typically exhibited in the *drag-and-drop* technique. The *Boomerang* [19] technique for use with a mouse allows the user to suspend the dragging component by using a mouse flick gesture to *throw* the object off the screen. The user can later *catch* the object to resume dragging. With multitouch throw-and-catch, TM teleports an object by specifying the source and target locations simultaneously, thus eliminating the time needed to drag the object.

The gestures bound to object manipulation operations all require the artist to first select an object for manipulation with either a one-touch or a conjoined touch. The most frequently used operations should be the simplest to perform, so arcball rotation and x-y translation only require the first touch and then a drag. For the remaining gestures, the artist uses both hands with no more than two fingers per hand. He selects the object with one hand, and then with the second hand, he adds touches away from the object to perform indirect manipulation. For each object manipulation gesture, the artist needs only to select the object and place any additional touches eyes-free to specify the object, operation, and parameters. In Maya, however, the artist needs to select a mode and sequentially target the object and manipulator.

To help make these gestures easy to remember, we used the first touch to indicate the category of manipulation. A conjoined touch on the object always begins a translation and a one-touch on the object begins either a rotation or a scale. When possible, we designed the motion of a gesture's second or third touch to reflect the motion of the object being manipulated. For example, translation along the z-axis moves an object up and down in screen space, so the second touch of the z translation gesture moves in an up and down motion. The second touch of the z rotation gesture moves side to side, which provides the sensation of spinning the object about a vertical axis. The second hand of the uniform scale gesture performs a pinching motion, which is commonly used for resizing photos on multitouch devices.

Camera Control

Camera control is an important component to set construction as the artist must be able to inspect the scene from different angles. To control the camera, the artist first holds down the camera button, which invokes a *quasimode* [26] in which Eden interprets any additional touches as a camera control gesture. This technique is analogous to holding down the 'alt' key in Maya to invoke camera control. We designed our camera control gestures to be similar to object manipulation gestures so they would be easier to remember. A one-touch drag rotates the camera in an arcball fashion, as it does for object manipulation. A conjoined touch drag translates the camera along the view plane (truck and pedestal), which is the same gesture for the planar translation in object manipulation. Lastly, we used the two-touch pinch gesture to move the camera forward and back (dolly), which is similar to the pinch used for scaling an object. We also included view direction rotation (roll) using the same two touches as dolly, as the orientation of the two fingers maps well to the camera's orientation. While holding the camera button, the artist can also choose a custom camera pivot by tapping on the scene

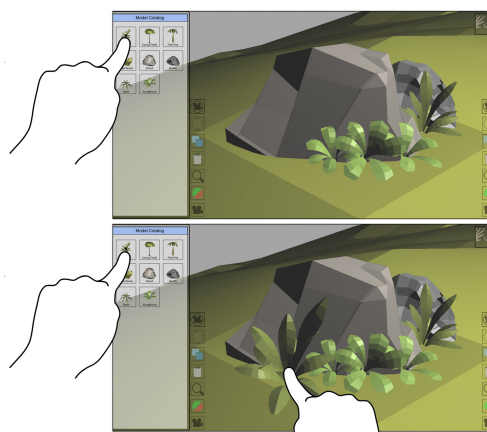


Figure 7. To add an object using throw-and-catch, the first finger selects the model and the second finger taps the position to place it.

or the artist can *frame* on an object (i.e. position the camera to provide a close-up view of the object) by tapping the object with a conjoined touch.

In an early iteration of Eden, we distinguished camera control from object manipulation not by a quasimode, but by the touch locations. If the touches did not hit an object, then the system interpreted the touches as a camera control gesture, otherwise it interpreted the touches as manipulating the touched object. However, this method had a major flaw as objects could easily fill the entire view, making camera control impossible.

Adding Objects

The artist can add an object to the set using throw-and-catch. Specifically, he selects and holds the object in the model catalog to throw with one finger and specifies the destination to catch the new object instance by tapping with a second finger (Figure 7). The base of the new object rests directly on the terrain or the closest object underneath the touch. This technique allows the artist to quickly drop a pile of shrubs onto the terrain, for example. The artist can even use all five fingers to place five new objects with one action, although in practice it could be difficult to position all five fingers in the desired configuration. Since no two objects are identical in nature, if the user selects an object in the model catalog with a conjoined touch, we add a small amount of randomness in scale and orientation to the placed object.

In addition to adding objects from the model catalog to the set, the artist can throw a copy of an object from the set into the model catalog. To store a new object, the artist holds an object in the scene with one finger and then taps inside the drawer with a second finger. Adding objects into the model catalog allows the artist to set the size and other parameters of the object and save it for future use. For example, he can scale up a rock object to the size of a boulder and then save it to the model catalog using this throw-and-catch technique.

Additional Commands

We incorporate quasimodes and stroke-recognition to support additional set construction commands.

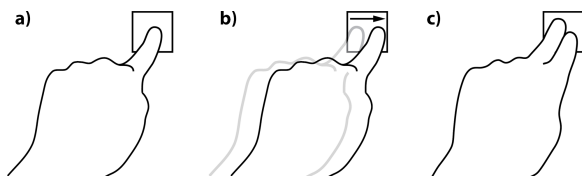


Figure 8. a) One-touch to invoke quasimode. b) Swipe on button triggers secondary action. c) Conjoined touch to make mode explicit.

Quasimodes and Buttons

Quasimodes in our application have the general advantage of keyboard-based quasimodes: the muscle tension needed to hold a key or button down reminds the user that a mode is currently invoked. In addition to camera control, we use quasimodes for various secondary operations that TM finds useful for organic set construction. Although we intended to avoid modes, quasimodes allow us to reuse simple gestures thereby keeping gestures easy to perform. The simplest gesture is a tap, and touch-based interfaces are particularly good for tapping on objects [10, 18, 28]. By holding down one of the quasimode buttons (Figure 3), the artist can simply use another finger to tap on objects to freeze/unfreeze, delete, duplicate, or group select them.

We augment our buttons in a number of ways. We place descriptive icons on the buttons so the artist can recognize the icon, whereas with a keyboard the artist would need to memorize key bindings. More importantly, a user can perform gestures directly on the icon. For example, if we have saved camera positions, a swipe through the icon (Figure 8b) can cycle back and forth between the saved cameras in a manner similar to Moscovich’s *Sliding Widgets* [22]. In addition, a conjoined touch tap on the camera icon (Figure 8c) can activate persistent camera mode, where the application only recognizes camera control gestures even if the camera button is not held down. Although we avoided regular modes, we provide camera mode so the artist can keep a hand free when only inspecting a set.

To make the buttons easy to access, we carefully considered their layout. Our multitouch screen sits almost horizontally, so in order to minimize the reach needed to hit buttons, we placed the buttons towards the bottom of the screen. Moreover, we put the same set of buttons on both sides of the screen to allow either hand to initiate a quasimode. We also made our buttons larger than the width of a finger to provide easy targeting.

Stroke Commands

In mouse and keyboard interfaces, commands are typically executed with hotkeys and menus. To keep the artist’s focus on the content, we avoided cluttering the interface with buttons or requiring the artist to navigate through menu hierarchies. Instead, the artist can execute commands by drawing single-stroke symbols in the stroke pad of the drawer (Figure 9 left). For example, drawing an ‘L’ opens a load model panel, whereas drawing a left arrow performs undo. The stroke pad interprets any touch as a potential stroke command, which allows the artist to execute single-stroke com-

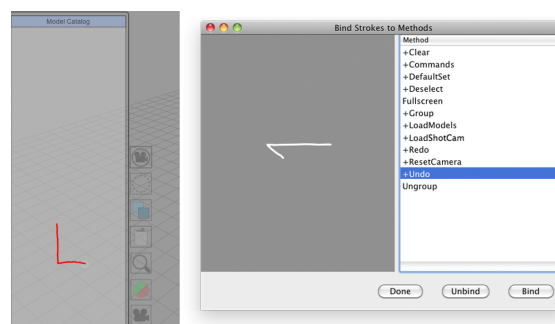


Figure 9. Left: Stroke pad. Drawing a stroke executes the corresponding command. Right: Stroke binding panel. The left panel displays the stroke bound to the highlighted command in the right panel. The artist can choose his own stroke by drawing a new stroke in the left panel.

mands that do not conflict with default object manipulation operations. Since the stroke pad is large and always in the same location, the artist can easily target the pad and draw a stroke with the left hand. Strokes can be difficult to remember, so the artist can define his own strokes for the supported commands, using a stroke binding panel (Figure 9 right). We use the dollar gesture recognizer [37] for stroke recognition.

QUALITATIVE EVALUATION

We asked TM to evaluate his experience using Eden to build a set for an upcoming feature film. We also asked a second set construction artist who has not previously used Eden to evaluate the system from the perspective of a novice user.

Apparatus

Eden runs on a multitouch workstation that we built using the frustrated total internal reflection technique of Han [14]. The size of the screen is 72.5 cm x 43.5 cm with a resolution of 1280 x 768 pixels. The device is patterned after a drafting table and is capable of detecting an arbitrary number of simultaneous touches. The artist interacts with the table by standing in front of the screen, which is mounted at a 23 degree incline. For text entry the artist uses a keyboard connected to a terminal next to the multitouch workstation. Text entry is reserved for infrequent actions such as naming a new set before saving it.

Veteran User Experience

Over the course of two 30-minute sessions, TM used Eden to build a set consisting of 136 trees for an upcoming feature film. He had built the same set previously in Maya, but he and his supervisor found no difference in quality between the two sets. We summarize his experience and evaluation of the system.

Object manipulation

According to TM, the rotation and scaling gestures on Eden are particularly effective because he does not need to first select the object to manipulate and then carefully pick a small manipulator to adjust the object as he does with Maya. In Eden, both the object and the operation are specified by the gesture. For rough placement, x-y translation in Eden is faster than in Maya. However, TM needs more preci-

sion when fine-tuning object positions, and x-y translation is cumbersome on a small object because the conjoined touch obscures the position of the object. Also, TM occasionally needs to dolly close to an object in order to select it, because distant or partially occluded objects have small target areas making them difficult to select. In working with Eden, TM did discover an unintended but positive side effect: in certain situations our implementation permits him to switch between operations without lifting the finger selecting the object. For example, if TM first performs an x-y translation, he can then fluidly transition to z translation by adding a one-touch with a second hand, without lifting the conjoined touch used for x-y translation.

Camera control

For TM, the Eden camera controls have slight usability advantages over Maya. Clutching a mouse is a physical annoyance for TM as he sometimes inadvertently slides the mouse off the working surface, which is not an issue with direct-touch. However, TM finds framing on an object difficult with Eden, because it often requires tapping on a small object, which is imprecise with the conjoined touch.

Adding objects

TM finds adding objects to a set with Eden is more efficient than with Maya. Using the throw-and-catch technique he can tap directly where a new object should roughly be positioned. The visual icons in the model catalog also help remind him what each model looks like. Maya does not provide preview icons.

Additional commands

TM considers quasimodes to be effective for accessing additional commands. Quasimodes permit the reuse of simple gestures, which makes the corresponding commands easy to invoke. The icons on the buttons help him remember which quasimodes are available. TM also finds strokes are as effective as keyboard shortcuts for executing simple commands such as undo and redo.

Repetitive Stress Injury

Over the years building sets, TM has developed repetitive stress injury (RSI) and currently wears a wrist protector on the hand he uses to control the mouse. To prevent his RSI from worsening, he takes regular breaks and finds other ways to exercise his wrist. TM finds that using two hands with Eden better balances the load between both hands. However, we do not have enough experience to know if different RSI problems will arise from multitouch interaction.

TM estimates that he is 20% faster building a set with Eden than with Maya. These results suggest that we have succeeded in providing an expert set construction artist a fully functioning multitouch application that is more efficient than an industry-approved application that has been refined over many years.

Nevertheless there is still room for improvement in both the interface and hardware of our system. According to TM, coarse placement is sufficient for the majority of the organic

set construction task. But, if we can address the occlusion problem for x-y translation and the precision problem for selecting small objects with techniques such as Shift [31] or FingerGlass [17], then we can provide a better overall experience for TM. Our hardware also limits the effectiveness of Eden. Our multitouch sensor only runs at 30 Hz and our touch detection system has a small delay when responding to input, which makes Eden less responsive than Maya. Also, detection for conjoined touch is not 100% robust, so the application may at times interpret TM's intentions incorrectly.

New User Experience

We designed Eden using the input from one set construction artist. To gain a better understanding of Eden's potential, we asked TP, a set construction artist with two years of experience, to use Eden for three, 45-minute sessions.

In the first session, we introduced Eden to TP, explaining its operations and features. He spent the second half of the session exploring and familiarizing himself with the interface by constructing a few small sets. His biggest early frustration was camera control, as the sensitivity did not match the Maya controls he was used to.

At the start of the second session we asked TP to recall the object manipulation gestures and the camera control gestures. He was able to perform each one without help, with the exception of world z rotation and one-dimensional scale. These two operations tend to be the least frequently used for object manipulation. After spending 20 minutes warming up and refamiliarizing himself with the interface, he was ready to construct a set. In 15 minutes he was able to build the set shown in Figure 3. At this stage, TP claimed he was "having fun" and building organic sets with his hands "feels like gardening." By the end of session two TP felt he was over the initial hump of learning the gestures.

TP returned for the third session three days after session two. Despite the break, TP was able to recall all the object manipulation and camera control gestures. He remembered the quasimode functions as well as the stroke commands for loading models, performing undo, and resetting the camera position. After ten minutes of practicing the various gestures, he spent the remaining time constructing a set.

Overall, TP found that Eden provided a more immersive experience than Maya, because he felt like he was "sculpting a space" with his hands and could "forget about the technology," which made him feel like he was sketching. In addition to enjoying the tactile quality of interacting with the objects, he found that using both hands to quickly transport objects in and out of the drawer was effective and efficient. We are encouraged that TP was able to learn and remember all of the object manipulation and camera control gestures after just two sessions, suggesting that our gestures are easy to learn and recall. Like TM, TP also discovered that he could perform fluid transitions between operations without lifting the selecting finger. He used fluid transitions frequently.

Although TP had a positive experience overall, he found certain operations difficult to perform with Eden. While he

could control the camera, he was uncomfortable with the gestures. He found that camera roll mapped to two fingers was confusing as he would inadvertently roll the camera when he wanted to only perform a dolly. Although the dolly gesture has enough degrees of freedom to also specify roll, we could separate the two operations or remove roll entirely. Also, his interpretation for the pinch motion to perform dolly was inverted from its intended use. When he spread two fingers apart he thought he was pushing the set away, so he expected the camera to dolly away from the set; instead, the camera dollyed towards the set. We could resolve this difference in interpretation by giving TP a method to customize gestures. For the majority of the organic set construction process, TP did not find precision to be an issue. However, like TM, when TP wanted to fine-tune the positions of a few objects, he had to dolly in close, otherwise he found selecting and manipulating small or distant objects difficult.

As we observed with TM, our hardware has room for improvement. TP felt he had to apply heavy pressure on the screen when performing gestures, making them slow and possibly straining on the hand. If we improve the hardware to recognize lighter touches and be more responsive, then we can provide a more comfortable and seamless experience.

LESSONS LEARNED

Based on our experiences designing a complete multitouch application and our interviews with professional set construction artists who used it, we summarize the following lessons:

- **Justify simultaneous interactions** – Determine how often users will use simultaneous interactions, if at all. If the benefits of simultaneous interactions do not outweigh the complexity of handling simultaneous interactions and the cognitive difficulty for a user to perform them, then support just one interaction at a time.
- **Balance gestures across both hands** – Split the touches across both hands in order to reduce the number of touches per hand and increase mobility. Fewer touches per hand makes gestures faster and more comfortable to perform.
- **Reuse gestures via modes** – As the number of operations increases, the more complicated the gestures generally become. Although we sought to reduce modes, quasimodes allow reusable gestures, which keep gestures simple.
- **Interpret gestures based on location** – Reduce conflicts by interpreting gestures made in one location (e.g. stroke pad) differently than gestures made in other locations.
- **Identify low precision tasks** – Evaluate the proposed application and consider whether precision will be a major factor. Techniques that compensate for touch imprecision [3], may slow the user's performance and limit the effectiveness of a multitouch interface.
- **Factor in occlusion** – Consider designing gestures that use indirect manipulation, so the user can perform manipulations away from the object and reduce hand occlusion, or allowing the user to release static touches once a gesture is recognized [39]. In addition, consider augmenting the interface to be occlusion-aware [30].

- **Throw objects** – A mouse cursor cannot be in two places at once, whereas a user's hands can. Pass objects between the hands to reduce travel times. Consider integrating a flick gesture to indicate a throw.

- **Design fluid transitions between gestures** – If two operations are often performed in sequence, design corresponding gestures that smoothly transition between the two. For example, a one-touch gesture can transition to a two-touch gesture with the application of a second touch.

EXTENSIONS

Eden primarily supports the rough placement of objects for organic set construction. For general set construction, we need to augment Eden with more precise interaction techniques. An artist should be able to adjust a single spatial parameter of an object without affecting the others, so we need additional gestures that control each spatial parameter separately. We could also incorporate existing techniques such as snap-dragging [4] to help the artist precisely align and position manmade objects found in general sets. In addition, a better hardware setup could improve precision by increasing touch resolution and reducing latency. Aside from object manipulation, we expect Eden's basic interface for camera control, adding objects, and setting modes to be sufficient for general set construction.

In addition to general set construction, our design decisions should transfer well to other single-user multitouch applications. By restricting applications to support only one operation at a time, developers can design simple, two-handed gestures that are easy to remember and comfortable to perform. Quasimodes allow the reuse of simple one-handed gestures, and when applicable, throw-and-catch eliminates the need for dragging.

CONCLUSION

We have developed and presented Eden, a multitouch application for organic set construction. A veteran set construction artist has used Eden to construct a scene for an upcoming feature film at Pixar Animation Studios. He found the tool to be more efficient than Maya, which demonstrates that multitouch is a viable option for producing professional level work for at least one workflow. From our design process, we found that focusing on supporting one operation at a time allows us to design simple gestures that split the workload across two hands. These gestures are easy to learn and remember as demonstrated by the experience of a set construction artist new to Eden. Despite focusing on organic set construction, our artists have some trouble with precision and occlusion issues. We believe that with further development we can address these issues to provide not only a better interface for organic set construction, but to begin supporting general set construction as well.

FUTURE WORK

When building Eden, we quickly discovered the difficulty in managing the variety of different gestures. Adding a gesture would require careful design and implementation to avoid conflict with preexisting gestures. A practical avenue of future work is to develop a framework that helps programmers

and interaction designers manage large gesture sets. Another area of future work that is important to the investigation of multitouch for professional use is to understand the long-term physical effects and potential RSI issues. Other areas include exploring more aspects of multitouch input, such as using finger identification to reduce the number of modes, and investigating how multitouch widgets should differ from mouse-based widgets, such as Sliding Widgets [22] and our buttons that interpret multitouch input.

ACKNOWLEDGMENTS

We would like to thank Kurt Fleischer, Dominik Käser, Tony Piedra, Craig Schroeder, and Allison Styer for their invaluable input. This work was partially supported by NSF grant IIS-0812562.

REFERENCES

- Autodesk Maya. <http://www.autodesk.com/>.
- S. Bae, R. Balakrishnan, and K. Singh. ILOveSketch: As-natural-as-possible sketching system for creating 3D curve models. *Proc. UIST 2008*, pages 151–160, 2008.
- H. Benko, A. Wilson, and P. Baudisch. Precise selection techniques for multi-touch screens. *Proc. CHI 2006*, pages 1263–1272, 2006.
- E. Bier. Snap-dragging in three dimensions. *Symposium on Interactive 3D Graphics*, 24:193–204, 1990.
- P. Brandl, C. Forlines, D. Wigdor, M. Haller, and C. Shen. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. *Proc. AVI 2008*, pages 154–161, 2008.
- R. Bukowski and C. Séquin. Object associations: a simple and practical approach to virtual 3D manipulation. *Symposium of Interactive 3D Graphics*, pages 131–138, 1995.
- X. Cao, A. Wilson, R. Balakrishnan, K. Hinckley, and S. Hudson. ShapeTouch: Leveraging contact shape on interactive surfaces. *Proc. Tabletop 2008*, pages 129–136, 2008.
- M. Cardinaels, K. Frederix, J. Nulens, D. Van Rijsselbergen, M. Verwaest, and P. Bekaert. A multi-touch 3D set modeler for drama production. *Proc. International Broadcasting Convention 2008*, pages 330–335, 2008.
- T. Erickson. The design and long-term use of a personal electronic notebook: a reflective analysis. *Proc. CHI 1996*, pages 11–18, 1996.
- C. Forlines, D. Wigdor, C. Shen, and R. Balakrishnan. Direct-touch vs. mouse input for tabletop displays. *Proc. CHI 2007*, pages 647–656, 2007.
- B. Froehlich, J. Hochstrate, V. Skuk, and A. Huckauf. The GlobeFish and the GlobeMouse: two new six degree of freedom input devices for graphics applications. *Proc. CHI 2006*, pages 191–199, 2006.
- S. Gabrielli, S. Bellutti, A. Jameson, C. Leonardi, and M. Zancanaro. A single-user tabletop card game system for older persons: General lessons learned from an in-situ study. *Proc. Tabletop 2008*, pages 85–88, 2008.
- Y. Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior*, 19(4):486–517, 1987.
- J. Han. Low-cost multi-touch sensing through frustrated total internal reflection. *Proc. UIST 2005*, pages 115–118, 2005.
- M. Hancock, S. Carpendale, and A. Cockburn. Shallow-depth 3D interaction: design and evaluation of one-, two- and three-touch techniques. *Proc. CHI 2007*, pages 1147–1156, 2007.
- S. Houde. Iterative design of an interface for easy 3-D direct manipulation. *Proc. CHI 1992*, pages 135–142, 1992.
- D. Käser, M. Agrawala, and M. Pauly. FingerGlass: Efficient multiscale interaction on multitouch screens. *Proc. CHI 2011*, 2011.
- K. Kin, M. Agrawala, and T. DeRose. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. *Proc. GI 2009*, pages 119–124, 2009.
- M. Kobayashi and T. Igarashi. Boomerang: Suspendable drag-and-drop interactions based on a throw-and-catch metaphor. *Proc. UIST 2007*, pages 187–190, 2007.
- M. Masliah and P. Milgram. Measuring the allocation of control in a 6 degree-of-freedom docking experiment. *Proc. CHI 2000*, pages 25–32, 2000.
- M. Morris, J. Wobbrock, and A. Wilson. Understanding users' preferences for surface gestures. *Proc. GI 2010*, pages 261–268, 2010.
- T. Moscovich. Contact area interaction with sliding widgets. *Proc. UIST 2009*, pages 13–22, 2009.
- J. Oh and W. Stuerzlinger. Moving objects with 2D input devices in cad systems and desktop virtual environments. *Proc. GI 2005*, pages 195–202, 2005.
- P. Peltonen, E. Kurvinen, A. Salovaara, G. Jacucci, T. Ilmonen, J. Evans, A. Oulasvirta, and P. Saarikko. It's mine, don't touch!: interactions at a large multi-touch display in a city centre. *Proc. CHI 2008*, pages 1285–1294, 2008.
- R. L. Potter, L. J. Weldon, and B. Shneiderman. Improving the accuracy of touch screens: an experimental evaluation of three strategies. *Proc. CHI 1988*, pages 27–32, 1988.
- J. Raskin. *The Humane Interface*. Addison Wesley, 2000.
- J. Reisman, P. Davidson, and J. Han. A screen-space formulation for 2D and 3D direct manipulation. *Proc. UIST 2009*, pages 69–78, 2009.
- A. Sears and B. Shneiderman. High precision touchscreens: design strategies and comparisons with a mouse. *International Journal of Man-Machine Studies*, 34(4):593–613, 1991.
- M. Shinya and M. Forgue. Laying out objects with geometric and physical constraints. *The Visual Computer*, 11(4):188–201, 1995.
- D. Vogel and R. Balakrishnan. Occlusion-aware interfaces. *Proc. CHI 2010*, pages 263–272, 2010.
- D. Vogel and P. Baudisch. Shift: A technique for operating pen-based interfaces using touch. *Proc. CHI 2007*, pages 657–666, 2007.
- F. Wang, X. Cao, X. Ren, and P. Irani. Detecting and leveraging finger orientation for interaction with direct-touch surfaces. *Proc. UIST 2009*, pages 23–32, 2009.
- C. Ware and D. Jessom. Using the Bat: a six-dimensional mouse for object placement. *IEEE Computer Graphics & Applications*, 8(6):65–70, 1988.
- D. Wigdor, G. Perm, K. Ryall, A. Esenther, and C. Shen. Living with a tabletop: Analysis and observations of long term office use of a multi-touch table. *Proc. Tabletop 2007*, pages 60–67, 2007.
- A. Wilson, S. Izadi, O. Hilliges, A. Garcia-Mendoza, and D. Kirk. Bringing physics to the surface. *Proc. UIST 2008*, pages 67–76, 2008.
- J. Wobbrock, M. Morris, and A. Wilson. User-defined gestures for surface computing. *Proc. CHI 2009*, pages 1083–1092, 2009.
- J. Wobbrock, A. Wilson, and Y. Li. Gestures without libraries, toolkits or training: a \$1 recognizer for user interface prototypes. *Proc. UIST 2007*, pages 159–168, 2007.
- M. Wu and R. Balakrishnan. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. *Proc. UIST 2003*, pages 193–202, 2003.
- M. Wu, C. Shen, K. Ryall, C. Forlines, and R. Balakrishnan. Gesture registration, relaxation, and reuse for multi-point direct-touch surfaces. *Proc. Tabletop 2006*, pages 185–192, 2006.
- K. Xu, J. Stewart, and E. Flume. Constraint-based automatic placement for scene composition. *Proc. GI 2002*, pages 25–34, 2002.
- R. Yeh, C. Liao, S. Klemmer, F. Guimbretire, B. Lee, B. Kakaradov, J. Stamberger, and A. Paepcke. ButterflyNet: a mobile capture and access system for field biology research. *Proc. CHI 2006*, pages 571–580, 2006.
- R. Zeleznik, K. Herndon, and J. Hughes. Sketch: an interface for sketching 3D scenes. *Proc. SIGGRAPH 1996*, pages 163–170, 1996.
- T. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill. A hand gesture interface device. *Proc. CHI 1987*, pages 189–192, 1987.