Extended 3D Object Manipulation: Gestures on Mobile Devices with Double-sided Multi-touch Input

|  |  |
| --- | --- |
| First Author Name (Blank if Blind Review)  Affiliation (Blank if Blind Review)  Address (Blank if Blind Review)  e-mail address (Blank if Blind Review)  Optional phone number (Blank if Blind Review) | Second Author Name (Blank if Blind Review)  Affiliation (Blank if Blind Review)  Address (Blank if Blind Review)  e-mail address (Blank if Blind Review)  Optional phone number (Blank if Blind Review) |

# ABSTRACT

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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*CHI 2009,* April 4–9, 2009, Boston, Massachusetts, USA.

Copyright 2009 ACM 978-1-60558-246-7/09/04...$5.00.

The paper presents a new interaction model for mobile devices, using both the front screen and the back-side surface as touch input regions. We define several double-sided finger gestures which enable users to interact with digital objects on the screen in a way which is more analogous to interaction in the real world. With simultaneous input from both the front and back surfaces of a mobile device, users are able to manipulate on-screen content similar to the way they manipulate small tangible objects with their hands.

## Author Keywords

finger gestures, double-side, multi-touch, mobile device.

## ACM Classification Keywords

H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

# INTRODUCTION

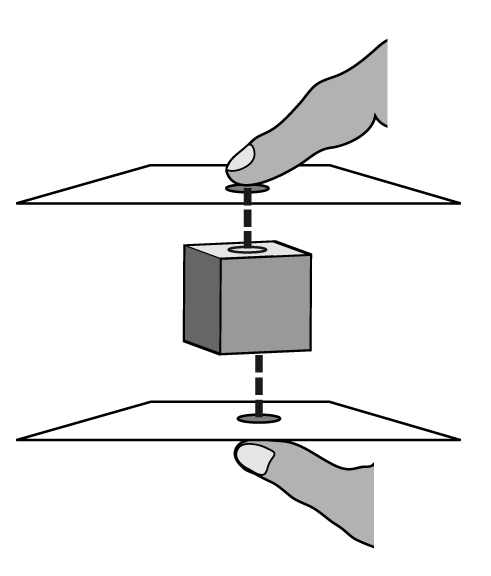
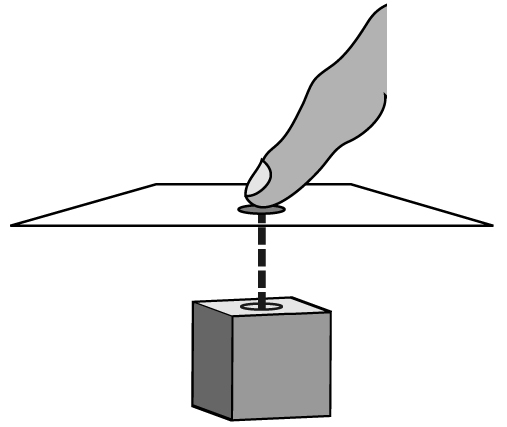


Figure 1: Virtual space between fingers: the object in virtual 3D space is projected to both the front and back surfaces of the device.

For early hand-held or mobile devices, keyboards and buttons were the major input interface. As mobile devices have shrunk and their displays grown, the screen-to-device ratio has increased. To fully utilize the available surface area, mobile devices such as the Apple iPhone and the HTC Touch Diamond have even replaced physical keyboards with stylus- or finger-controlled touch screens.

In conventional single touch-screen input, users hold the device with one hand from the device’s backside while manipulating its screen with the other hand using either a stylus or a finger. Since fingers are in contact with both the screen and the device’s backside, exploring the device’s backside as a touch input region [12, 15] makes a lot of sense, especially considering that the size of device’s backside surface is substantial and not used. This work further extends the use of device’s backside touch input, as an additional degree of freedom, for 3D interface manipulation on mobile devices. The scale of mobile devices also renders them suitable for 3D interface interaction, simulating real world scenarios: interacting with small tangible objects held in the hands.

# Double-side Multi-touch Interaction

In a conventional single-sided touch-screen device, a 3D scene is projected onto a 2D plane. When the finger touches the surface of device, the virtual contact position on the digital object inside the 3D space is at the end of an arrow perpendicular to the screen and which originates from the contact point on the device, as shown in Figure 1(a). This kind of manipulation allows users to access those surfaces of a digital object which face the user. This, however, constrains the interaction to two dimensions over the horizontal and vertical axes parallel to the screen.

By adding touch inputs to the back side surface, the degree of freedom for manipulation can be extended to three dimensions, and by using the same method described in the previous context we can access the other side of the digital object by touching the back of the hand-held device. Figure 1(b) illustrates an object in 3D space between the front- and back-side touch surfaces. This approach allows for the control of digital objects in virtual space in a way analogous to the manipulation of small objects in the hands.

Based on this double-side multi-touch interaction model, this study designed and implemented several 3D finger gestures for manipulate 3D interface on mobile devices. Furthermore, this study explores empirically how well the double-sided multi-touch input interaction compares with existing single-sided multi-touch input interaction for users to manipulate 3D interface on mobile devices.

The rest of the paper is organized as follows. We first review the related work. Next, we present design concept behind double-side multi-touch input and explain each of its finger gestures for manipulation 3D interface. We then describe a user study that compares the speed of the double-side and single-side multi-touch inputs, as well as the users’ subjective perceptions of them. We discuss the implications and suggest how to improve the multi-touch input interaction. Finally, we draw conclusion and future work.

# Related work

## Unimanual and Bimanual

Forlines et al. [6] conducted an experiment to compare the difference between unimanual and bimanual, direct-touch and mouse input. Their results show that users benefit from direct-touch input in bimanual tasks. A study by Tomer et al. [10] reveals that two-handed multi-touch manipulation is better than one-handed multi-touch in object manipulation tasks, but only when there is a clear correspondence between fingers and control points. Leganchuk et al. [8] conducted experiments to validate the advantage of bimanual techniques over unimanual ones. Kabbash et al. [7] studied the “asymmetric dependent” bimanual manipulation technique, in which the task of one hand depends on the task of the other hand, and showed that if designed appropriately, two-handed interaction is better than one-handed interaction.

## Precision pointing using touch input

In addition to two well-known techniques, Zoom-Pointing and Take-Off, the high precision touch screen interaction project [1] proposes two complementary methods: Cross-Keys and Precision-Handle. The former uses virtual keys to move a cursor with crosshairs, while the latter amplifies finger movement with an analog handle. Their work improves pointing interaction at the pixel level but encounters difficulties when targets are near the edge of screen. Benko et al. [5] developed a technique called dual-finger selection that enhances selection precision on a multi-touch screen. The technique achieves pixel-level targeting by dynamically adjusting the control-display ratio with a secondary finger while the primary finger moves the cursor.

## Back side input

*Under the table interaction* [16] combines two touch surfaces in one table. Since users cannot see the touch points on the underside of the table, the authors propose using visual feedback on the topside screen to show the touch points on the underside of the table, which improves the touch point precision on the underside. *HybridTouch* [12] expands the interaction surface of a mobile device to its back side. This is done by attaching a single-touch touch pad to the back-side of the device, enabling the non-dominate hand to perform document scrolling on the backside touch pad. *LucidTouch* [15] develops pseudo-transparency for a mobile device’s back-side interaction: by using a camera extended from the device’s back side to capture the locations, fingers operating on the device’s back-side are shown on the front screen. Wobbrock et al. [18] also conducted a series of experiments to compare the performance of the index finger and thumb on the front and the rear sides of a mobile device. Baudisch et al. [4] created a device called *nanoTouch* which enables direct-touch input on the back side of very small devices, with the *shift* technique to solve the “fat finger” problem.

## Gesture recognition

Wu et al. [20] articulate a set of design principles for constructing multi-hand gestures on direct-touch surfaces in a systematic and extensible manner, allowing gestures with the same interaction vocabulary to be constructed using different semantic definitions of the same touch data. Wobbrock et al. [19] present an approach to designing tabletop gestures which relies on eliciting gestures from users instead of employing gestures created by system developers. Oka et al. [11] proposed a method to track users’ finger tip trajectories while presenting an algorithm to recognize multi-finger symbolic gestures.

## 3D space interaction

Wilson et al. [17] proposed the idea of proxy objects to enable the physically realistic reaction of digital objects in the 3D space inside a table surface. Hancock et al. [9] explored 3D interaction involving the rotation and translation of digital objects on a tabletop surface with limited depth, and compared user performance in tasks using one-, two- and three-touch respectively.

Our work emphasizes direct-touch finger gestures that involve simultaneously touching both sides of a mobile device. While these gestures may applied to various applications on mobile devices, we here focus on 3D object interaction through double-sided input manipulation and are interested in how well the double-side input compares with single-side input for manipulating 3D interface on mobile devices.

# DOUBLE-SIDE MULTI-TOUCH INTERACTION

Although multi-touch input enables scrolling and the zooming of a document, its manipulation is constrained to two dimensions over the horizontal or vertical planes of a mobile device’s 2D touch screen. By adding touch inputs from the device’s back side, the degree of freedom for manipulation can be extended to a pseudo three dimensions. This 3D space concept is as described in the under-table interaction work [9], in which the virtual 3D space shown in the device’s display is a “fixed volume, sandwiched” between the front and the back surfaces of the device. This table-based 3D concept was extended in our double-side multi-touch interaction model on a mobile device, and then a set of touch gestures was created for manipulating 3D objects.

## Hardware Prototype

A double-side multi-touch device prototype was created as shown in Figure 2. This was accomplished with two iPod-touch devices attached back to back. In this case the back-side iPod-touch device became the back-side multi-touch pad for the device. Each iPod-touch had a 3.5”, 320x480px capacitive touch screen, and supported up to five simultaneous touch points. The width, height, depth, and weight of each device are 62mm, 110mm, 20mm, and 230g respectively. The touch input signals of the back-side iPod-touch device, including locations of the touch points, were then transmitted to the front side iPod-touch device through an ad-hoc Wi-Fi connection. Note that the current iPod-touch device supports at most five touch points at the same time. When the 6th finger touches the screen, all touch events are cancelled. We have found that five touch points are sufficient to implement our double-side, multi-finger touch gestures.

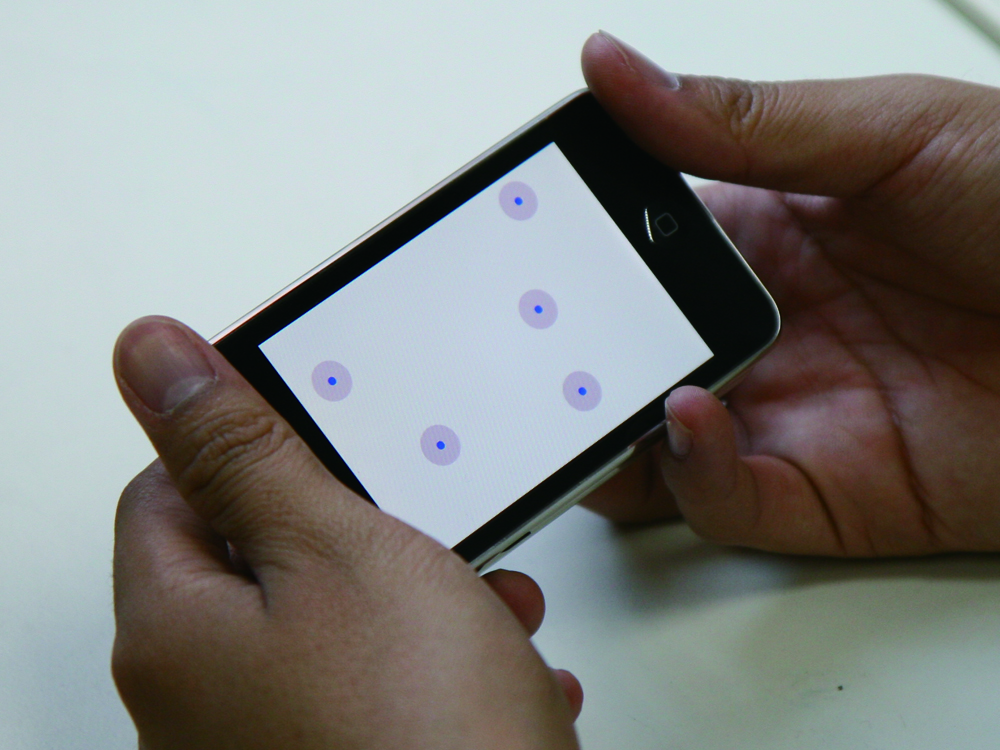


Figure 2: Prototype of our double side multi-touch input mobile device. The location of back side touch points are displayed to provide visual feedback.

## Double-side Multi-finger Touch Gestures

In traditional single-side touch interaction, manipulating a 3D object is done by touching one face of the object. In contrast, manipulating a real-world object in a physical 3D space involves a rich set of multi-finger actions such as grabbing, waving, pushing or flipping the target object. If the object is soft or elastic, it can also be stretched or kneaded into different shapes. Based on the double-side multi-touch interaction model, we designed several double-side multi-finger touch gestures similar to manipulating 3D objects in the physical world.

Figure 3: A sequence of three gesture states. The begin state may correspond to more than one gesture.

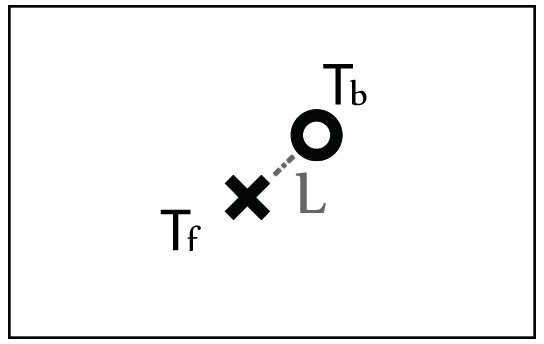
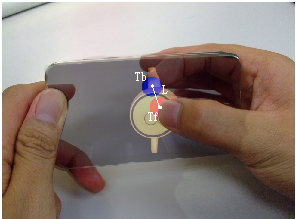


Figure 4: The *Grab* gesture is triggered when the distance between the two touch points on opposing sides is less than a predefined threshold.

A gesture is a sequence of interactions. Baudel and Beaudouin-Lafon [3] proposed a model for designing gestures where each command is divided into phases from initialization to ending phase. In our model, each gesture possesses three states: begin, dynamic, and end. The begin state in some situations is a common state shared by different gestures: the identity of the gesture is determined according to the fingers’ movement sequence. Once a gesture is recognized, it enters the dynamic state, during which the begin states of other gestures are not triggered until the current gesture ends. Figure 3 shows the order of the three gesture states. Here we present five sample double-sided multi-touch gestures used in 3D interface operations: *Grab*, *Drag*, *Flip*, *Push*, and *Stretch*.

### Grab

In the real world, grabbing an object (for example, a coin or a card) by hand involves using at least two fingers to apply opposing pressure on two or more surfaces of the target. Mapping this physical grabbing action onto our double-sided multi-touch input interface yields the gesture triggered by two fingers from opposite side of the surface touching the same target object. If the distance between the two touch points on opposing sides is less than a predefined threshold, then the *Grab* gesture enters its begin state and the two touch points are recorded for further recognition of other gestures. In our model, the *Grab* gesture includes the begin and end states but not the dynamic state. When we want to rotate or drag a small object, we first grab it; hence the action following a *Grab* depends on the relative movement of the fingers. Since the two sides’ touch points do not always make contact with the device at the same time, the *Grab* gesture enters its begin state if the two side touch points are close enough: the order and delay between the two touch events does not affect the *Grab* begin state. The *Grab* gesture enters the end state if one or two fingers leave the surface of the device. As shown in Figure 4, is the touch point of the front side and is from the back. *L* is the distance between and . When *L* is short enough and when both and are in the display region to which the object is projected, the *Grab* gestured is triggered and object is grabbed by the two fingers.

### Drag

In the real world, dragging a small object by hand involves first grabbing the object with opposing fingers and then moving the object by moving the two fingers together in a single target direction. Mapping the physical dragging action onto our model yields the *Drag* gesture, which starts in the same begin state as the *Grab* gesture. As Figure 5 shows, when the fingers move, represents the vector starting from the initial touch point and ending at the new touch point . is the corresponding back-side vector. Let *F* be the inner product of and ; the sign of *F* determines whether the *Drag* or *Flip* gesture is to be triggered. If *F* is positive, the gesture is recognized as *Drag* because it indicates that the two fingers are sliding in the same direction: the target object thus moves along . If either finger leaves the surface, the *Drag* gesture ends so the system can wait for other events to trigger the next gesture’s begin state.

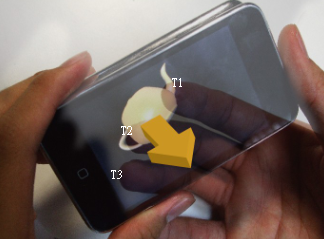


Figure 7: The *Push* gesture. ,, and are touch points from back side. The object moves toward the user when fingers keep touching it; the velocity varies according to number of touching fingers.



### Flip

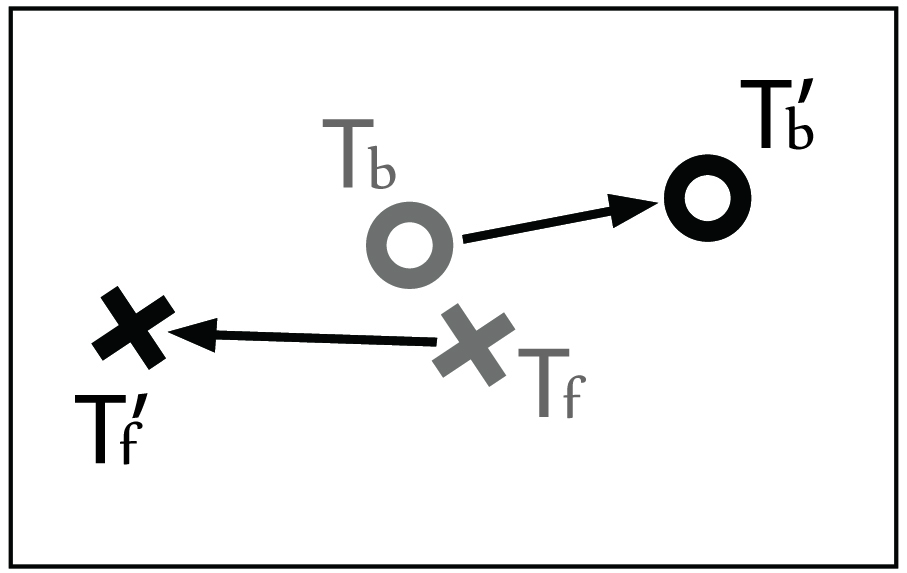


Figure 6: The *Flip* gesture. Let ; if *F* is negative, the *Flip* gesture enters the dynamic state and the object rotates about the axis perpendicular to which passes through the center of the object.



In the real world, similar to dragging, when flipping a small object by hand, we first grab the object and then flip it with two fingers sliding in opposite directions. Mapping the physical drag action onto our model yields the *Flip* gesture, which also enters the same begin state as the *Grab* gesture. As shown in Figure 6, the definition of , ,and *F* is the same as that in the *Drag* gesture. The only difference is that during the *Flip* gesture, the opposing directions of and cause *F* to be negative. Once the *Flip* gesture enters its dynamic state, is the torque exerted on the object which forces it to rotate with respect to the axis perpendicular to which passes through the center of the object. As with the *Drag* gesture, when one or two of the fingers leaves the surface, the *Flip* gesture enters the end state and the object stops rotating. As the *Drag* and *Flip* gestures cannot occur at the same time, the rotation and translation of the object are independent of each other.



### Push

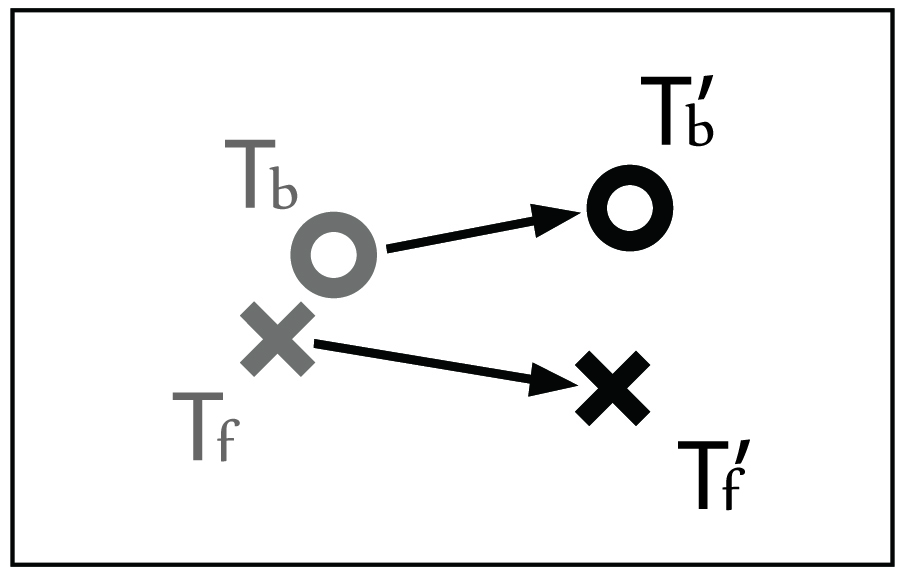
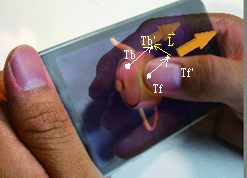


Figure 5: The *Drag* gesture. Touch points and trigger the begin state. and are the touch points after the fingers move. Let ; if *F* is positive, than the *Drag* gesture enters the dynamic state and the object moves along .



In the real world, when we exert force on an object to push it, our hands maintain contact with a certain point on the object. Correspondingly, we define the *Push* gesture as that which moves the digital object along the axis perpendicular to the touch screen. When the user pushes the object by touching the back-side screen, the object moves toward the user. In our model, the begin state of the *Push* gesture is triggered by tapping the surface twice by two or more fingers. The gesture enters the dynamic state when the fingers keep on touching the same position, and the object moves in the direction the fingers are pushing. The more fingers on the object, the faster it moves. When the number of fingers touching the object is less than two, the *Push* gesture ends. Figure 7 indicates how this gesture works.

### Stretch

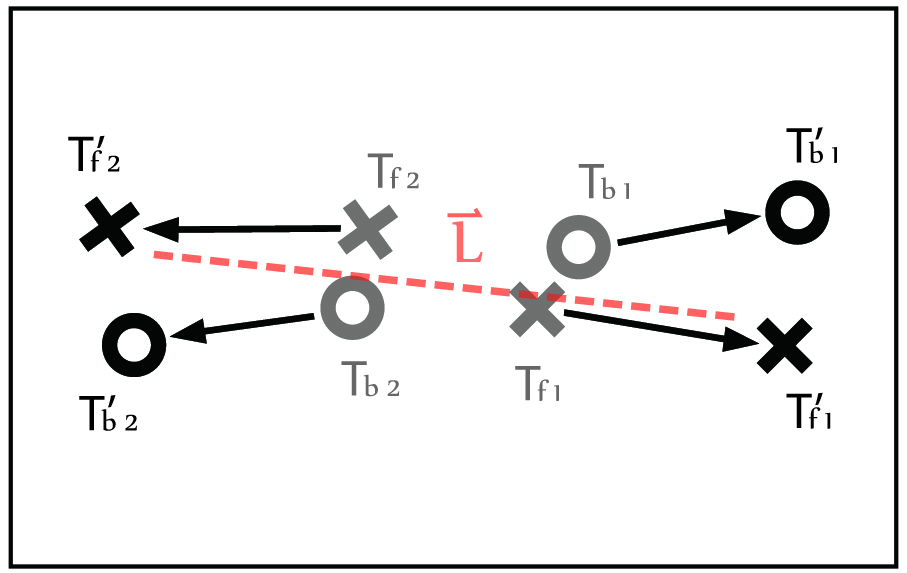
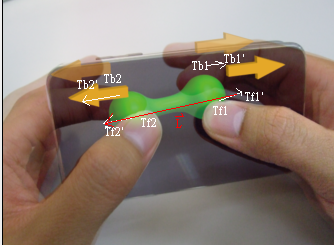


Figure 8: The *Stretch* gesture. Index 1 and 2 represent different hands. Let , the object enlarges/deflates as increases/decreases.



In the real world, stretching an elastic body by hand involves first grabbing the object at several positions and then applying forces at these positions to stretch or knead it. Mapping this physical stretch action onto a double-sided multi-touch input device yields the *Stretch* gesture, which enters the begin state when two sets of the *Grab* gesture begin. When the fingers move, the *Stretch* gesture enters dynamic mode and the size of object varies according to the length change of . The gesture ends if any of the four fingers leaves the surface. Figure 8 illustrates the *Stretch* gesture.

# Pilot STUDY

We conducted a pilot study to evaluate the performance of object manipulation with double-side multi-touch gesture. We designed a series of tasks which consisted of all gestures we mentioned above (*Grab*, *Drag*, *Push*, *Stretch*, and *Rotate*) on our prototype device. Each tasks has a single-sided version to compare with our double-sided design. All tasks were conducted with the device in the horizontal direction (landscape mode). Three participants were invited to this study. We got some feedback about our design from them.

The pilot users could not *Grab* the objects easily in this study. We used the touch point of back-side finger as the index to select objects on the screen in this design. When the objects went small, it was hard for users to *Grab* it successfully at first time. On the other hand, two of our users considered that Both-side selection was similar to Back-side, but it had to use one more finger (the front one) to complete this gesture. Two-finger gesture is instinctively slower than one-finger gesture if the back-side finger touch point does not support the front-side one.

To solve these two problems, we redesign our *Grab* gesture in making it possible to select objects by both fingers and enlarging the selection threshold. We also implemented a visual feedback system which will mention below to help users select objects intuitively. To find the relationship between target size and different gesture technique, we did further research in the formal user study with these improvements.

# VISUAL feedback

Back on previous research on back-of-device interaction, users lack visual feedback indicating the positions of their fingers on the back side of the screen. In our work, we have found it was difficult to touch the target point on the back side without visual feedback. As a visual aid, we added shadows to where the front-side area we touched. These shadows let us know where is the exact area we can select objects when our thumbs touch front side screen. We also added light dots to indicate where the backside fingers are touching. These dots let us know whether the backside fingers are in the right location to perform features such as grabbing or dragging. They correspond to the cursor on personal computers; targets are not occluded because each dot is only one pixel. Moreover when our fingers touch the object on the backside, the object would be enlarged. The metaphor of this design is that when we tapping the object toward us, the object seems to be larger and be easier to be selected by the front finger.

# USER STUDY

To test the performance of double-side multi-touch, we conducted a formal user study that consists of three kinds of object interaction, selection, moving, and scaling. The primary aim of this study is to test which technique has better performance when user try to operate objects in different ways in double-sided environment.

## Participants

Ten participants (5 males, 5 females) from our university were recruited as participants in the study. Half of them are major in electronic engineering/computer engineering and the others major in marketing, management, and literature. Five of them have experience of using multi-touch input on mobile devices. The other 5 participants have never used any multi-touch input interface before. All of them are right-handed. Ages ranged from 19 to 22.

We conducted our user study in our research lab. During the test, participants were given with an office chair and desk. They could choose to sit in any comfortable way they like. It cost around 45 minute for each user.

## Apparatus

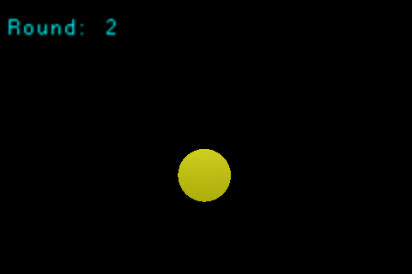


Figure 9: The scene of the object selection experiment

The experiment was performed on a double-side multi-touch input device (i.e., two Apple iPod-touch devices attached back-to-back shown in Figure 2). The iPod touch has a 480-by-320-pixel display, and its resolution is 163 pixels per inch (6.42 pixels/mm).

## Methodology

In each experiment, there were three phases: Demonstration phase, Familiarization Phase, and Experimental Phase. In the first phase, we explained the operation of the devices and showed to the participants how to perform different gestures in this experiment. In the familiarization phase, participants performed a practice set that used the same tasks as the experimental phase. They could continuously practice until they were comfortable.

During the experimental phase, the participants were asked to follow the task procedure, which will describe in the following sections. Test programs automatically log task completion times, numbers of error, and finger-on-device time of each user.

In the first and second phases, participants were allowed to ask questions about performing gestures. We helped them to perform gestures correctly in these two phases; however, we gave them no any help in the final phase.

## Measurement

In these three experiments, we log the following data:

Figure 10:

### Completion Time

The completion time is the time which user cost to complete the task. We can say one is faster than the other if its completion time is fewer.

### Extra Movement

The extra finger actions users do in that trial. We did not use *Error Rate* here because some of our gestures are continuous. For a gesture which needs only one finger, it generates *Extra Movement* if a user does this gesture with two or more fingers.

# EXPERIMENT 1: SELECT OBJECT

## Gestures

To select an object, we use our *Grab* gesture as Both-side technique. In Back-side input version, user taps once on the object with his/her index finger on the back of the device. In Front-side input version, user taps once on the object with his/her thumb on the front of the device.

## Procedure

During each round in this experiment, our testing program generated one yellow circular plate with random position. When user selects the plate, the task in a round is completed. Figure 9 shows the testing scene.

The independent variables in this experiment were *Technique* (Both-side, Back-side, or Front-side), and *TargetSize* (10, 8, 6, 5, 4, 3, 2 mm). For each size of target, there were 4 repeat rounds. Totally a participant had to do 3 \* 7 \* 4 = 84 trials in this experiment.

## Hypotheses

Those hypotheses below are base on the operation of object selection on the double-sided device.

(H1) Front-side will be slower than Back-side, and Back-side will be slower than Both-side. Users will generate the most *Extra Movement* when they use Front-side but the least one when they use Both-side.

(H2) *TargetSize* will influence the speed and *Extra Movement* of those three techniques, but in different ways. If the targets are small, the performance of Front-side will be better; however, Both-side will perform better when the targets are large.

## Results

### Completion Time

Figure 10 shows the ten participants’ average completion times for each *Technique* and *TargetSize* in the object selection experiment. We performed an analysis of variance (ANOVA) for *Technique* X *TargetSize*. The main effect for each factor was: *Technique* (F2,838=9.874, p<0.0001), and *TargetSize* (F6,834=37.832, p<0.0001). *Technique* X *TargetSize* was significant interaction (F12,828=6.972, p<0.0001).

Figure 11:

We also performed post hoc Tukey’s pair-wise comparison. It showed that when the *TargetSize* was as small as 3 or 2 mm, Both-side was significant faster than Front-side (p<0.0001) and that Back-side was significant faster than Front-side (p<0.0001). It showed no significant result when *TargetSize* was 10, 8, 6, 5, or 4 mm.

### Extra Movement

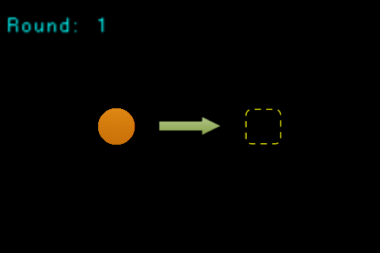


Figure 12: The scene of the object moving experiment

Figure X shows the mean and standard error of *Extra Movement*. An ANOVA was then performed for *Technique* X *TargetSize* and showed that: *Technique* (F2,838=35.665, p<0.0001), and *TargetSize* (F6,834=19.105, p<0.0001). *Technique* X *TargetSize* was significant interaction (F12,828=13.521, p<0.0001).

Tukey’s pair-wise comparison showed that when the *TargetSize* was 4, 3, or 2 mm, both Back-side and Both-side were significant smaller in *Extra Movement* than Front-side (p<0.0001). No significant result was shown when the *TargetSize* was larger than 5mm.

## Discussion

This results support hypothesis H1 partly. Both-side was significant faster than Back-side and Front-side. Since user is hard to use their thumb to select objects precisely, they may try many times to success. The same result was shown in *Extra Movement*. Users generated more *Extra Movement* when they used Front-side method. We consider this is why they did slower when they were using this gesture to select objects. On the other hand, we did not find significant difference between Back-side and Both-side, which is not what we expected. We thought that *Grab*, one of our Both-side techniques, would improve the performance of Back-side gesture. Though it didn’t get a significant result, the mean chat in figure X shows that Both-side was faster than Back-side in all *TargetSize*.

H2 was partly accepted. The mean chart in figure 10 showed that the average completion time increased as the target size decrease. It is obvious in figure 10 and figure 11 that Front-side did slower than Back-side and Both-side and its *Extra Movement* was the most in the case of small targets. However, there was no any significant result when the target size was large.

# EXPERIMENT 2: MOVE OBJECT

## Gestures

To move an object, we use our *Drag* gesture as Both-side technique. In Back-side input version, user taps once on the object with his/her index finger on the back of the device, and then slide the finger to move the object to target position. In Front-side input version, user taps once on the object with his/her thumb on the front of the device, and then slide the finger to move the object to target position.

Figure 13:

Figure 14:

Figure 15:

## Procedure

During each round in this experiment, our testing program generated one yellow circular plate. User selects the plate, and then moves it to the target area to complete this task. Figure 12 shows the testing scene. The independent variables in this experiment were *Technique* (Both-side, Back-side, Front-side, or Hybrid), and *Direction* (8 directions, E, NE, N, NW, W, SW, S, SE). For each direction, there were 4 repeat rounds. Totally a participant had to do 4 \* 8 \* 4 = 128 trials in this experiment.

## Hypotheses

Those hypotheses below are base on the operation of moving objects on the double-sided device.

(H3) Front-side will be faster than Back-side and Both-side, and generate less *Extra Movements.*

(H4) There will be no significant difference in completion time and *Extra Movement* between Both-side and Back-side.

(H5) *Direction* will affect the speed and *Extra Movements* of each *Technique.*

## Results

### Completion Time

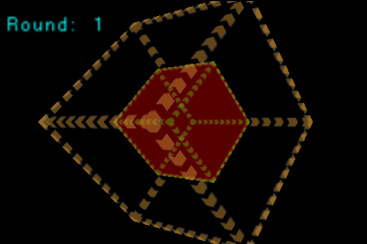


Figure 16: The scene of the object size changing experiment

Figure 13 shows the 10 participants’ average completion times for each *Technique* and *Direction* in the object moving experiment. An ANOVA test showed that a main effect for: *Technique* (F2,957=15.631, p<0.0001), but not *Direction*. *Technique* X *Direction* showed no significant interaction. Tukey’s pair-wise comparison showed that Front-side was significant faster than Both-side and Back-side (p<0.01).

Figure 17:

### Extra Movement

Figure 14 shows the mean and standard error of *Extra Movement*. An ANOVA was then performed for *Technique* X *Direction* and showed that: *Technique* (F2,838=35.665, p<0.0001). There was no significant interaction in *Technique* X *Direction*. Tukey’s pair-wise comparison showed that the *Extra Movement* Back-side and Front-side were significant less than Both-side (p<0.01).

## Discussion

H3 can be accepted since the significant result which Front-side faster than the others in completion time. H4 was partly supported by our results. There was no significant difference in completion time between Both-side and Back-side, but not in *Extra Movement*. As we expect, moving two fingers is slower that moving only one finger. Something unexpected is that Front-side and Back-side were no significant difference in *Extra Movement*, and that the *Extra movement* of Both-side were significant larger than Front-side and Back-side. We consider the reason is that both Front-side and Back-side gesture in moving objects were single-touch manipulation. Users can easily keep only one finger on the front or back of the screen without interruption, but they can hardly keep both fingers moving always together. When users use Back-side gesture, they could not predict the moving direction precisely, just as using Front-side gesture. As a result, Back-side was faster in completion time but got more *Extra Movements.*

H5 is rejected by the insignificant result. However, figure 15 shows that in Both-side gesture, some directions (NW) do have a little effect on the mean of completion time and *Extra Movement*. In addition, some participants said that it was difficult for them to *Drag* (our Both-side technique) an object toward North, because of the size of hand. The participants with small hands sometimes hit the part of the hand between the thumb and index finger. The main problem is the thickness of our prototype device. The result implies that we might make our next prototype thinner to make users more comfortable.

Figure 18:

## EXPERIMENT 3: CHANGE THE SIZE OF OBJECT

## Gestures

To enlarge/reduce the size of an object, we use our *Stretch* gesture as Both-side technique. In Back-side input version, users enlarge/reduce the size of object by expanding/shrinking with two index fingers which are on the back side of device. In Front-side input version, user expands/shrinks their two thumbs to conduct this manipulation.

## Procedure

In first ten rounds, our testing program generated one green cube inside an orange cube. Participants have to enlarge the size of the cube inside to the same size as the outer one. In the last 10 rounds, there was a green cube outside an orange cube. Reduce the size of the green cube to the same size of the orange one to complete one trial. Figure 16 shows the testing scene.

The independent variables in this experiment were *Technique* (Both-side, Back-side, Front-side), and *ScaleType* (Enlarge or Reduce). For each *ScaleType* of target, there were 9 repeat rounds. Totally a participant had to do 3 \* 2 \* 9 = 54 trials in this experiment.

## Hypotheses

Those hypotheses below are base on the operation of changing the size of objects on the double-sided device.

(H6) Front-side and Back-side will be faster than Both-side, and less *Extra Movement*.

(H7) There will be no significant difference in completion time and *Extra Movement* between Front-side and Back-side.

(H8) *ScaleType* will affect the performance. Reducing the size of objects will be slower than enlarging objects.

## Results

### Completion Time

Figure 17 shows the average completion times for each *Technique* and *ScaleType* in the object scaling experiment. An ANOVA test showed that a main effect for: *Technique* (F2,537=7.446, p<0.001), and *ScaleType* (F1,538=26.589, p<0.0001). *Technique* X *ScaleType* was significant interaction (F2,538=9.090, p<0.0001).

Tukey’s pair-wise comparison showed that Back-side was significant faster than Both-side (p<0.0001).

### Extra Movement

Figure 18 shows the mean and standard error of *Extra Movement*. An ANOVA was then performed for *Technique* X *ScaleType* and showed that: *Technique* (F2,537=17.736, p<0.0001), and *ScaleType* (F1,538=12.855, p<0.01). *Technique* X *ScaleType* was significant interaction (F2,538=15.132, p<0.0001).

Tukey’s pair-wise comparison showed that Back-side and Front-side were significant faster than Both-side (p<0.01).

## Discussion

H6 is supported by our results that Front-side and Both-side were significant faster than Both-side. Like the result in experiment 2, Front-side and Both-side need fewer fingers to complete the gesture. The participants can use them easily without interruption. In addition, expanding and shrinking gestures are now widely used in current multi-touch mobile device. Half of our participants had the experience of multi-touch manipulation, so they might be more familiar with this kind of operation.

There was no significant difference between Front-side and Both-side, which means H7 could be accepted.

Figure 17:

Figure 18:

The result supports H8. Figure 17 shows that Both-side, Front-side, and Back-side are all slower when user reduces the size of an object. Figure 18 shows the same thing in *Extra Movement*. Some participants said that reducing objects’ size makes their fingers packed into a small area, so they could hardly complete the task. It could be improve by accelerate the reducing speed, but it may make this gesture less precise at the same time. We might let the users to define the parameter on their own in the future work.

# discussion

## Design guidelines

Based on the results and observation of the user evaluation, we seggest some guidelines for designing input interfaces on double-side multi-touch mobile devices.

* *Hybrid both-side and single-side gestures*: When doing actions such as selecting and moving objects in *xy*-plane, users only have to access one surface of objects. For these kinds of actions, single-side gestures are more suitable and can be performed faster. For actions involving accessing back sides of objects or motion along *z*-axis, double-side gestures make the operation more close to the interaction of objects and hands in real world. If developers combine the merits of both single-side and double-side gestures and design a hybrid input method, users can benefit from the mixed type of interaction.
* *Posture when holding the device*: In performing gestures which require two hands to simultaneously touch the front and back sides, users fixed the devices within their palms. This posture would constraint their fingers to access the touching regions near the center of screens. Users may pay more attention so as not to let the device fall out of their hands. Actions involving accessing contents near edges of screen should avoid using these kinds of gestures.
* *Unconscious touch points*: Users usually hold mobile devices with two or more fingers contacting back side of devices to fix them. The system should identify those fixing or unconscious touch points and neglect them so they would not affect the recognition of expected gestures. Techniques such as accelerometers could help the system detect how the device is being held so the system can decide which touch points to filter out in recognizing gestures.

# Conclusion & future work

We have presented several sample finger gestures that use both the front and back sides of a mobile device. In addition to defining algorithms to recognize different gestures, we develop applications for hand-held devices with multi-touch input interfaces to evaluate the usability of the design principles. We present a user study that quantitatively and subjectively compared user behavior when manipulating a mobile device’s 3D interface between conventional single-sided and double-sided multi-touch input gestures.

In this paper, we focused on digital 3D space; we plan to next apply this double-sided multi-touch technique to more commonly-used applications such as text input and precision selection on a small screen device. The prototype, a combination of two iPod-touches, is cumbersome as mobile devices go: we plan to implement this technique on a lighter, thinner device.

# acknowledgements

# REFERENCES

1. Albinsson, P.-A. and Zhai, S. *High precision touch screen interaction*. In Proceedings of CHI, pp. 105–112, New York, NY, USA, 2003. ACM Press.
2. Balakrishnan, R. and Hinckley, K. *Symmetric bimanual interaction*. In Proceedings of CHI, pp. 33–40, New York, NY, USA, 2000. ACM Press.
3. Baudel, T. and Beaudouin-Lafon, M. (1993). *Charade: remote control of objects using free-hand gestures*. Communications of ACM, 36(7). pp. 28–35.
4. Baudisch, P. and Chu, G. *Back-of-device Interaction allows creating very small touch devices*. In Proceedings of CHI 2009, Boston, MS, April 4–9, 2009.
5. Benko, H., Wilson, A., and Baudisch, P. *Precise Selection Techniques for Multi-Touch Screens*. In Proceedings of CHI 2006, Montreal, Canada, April 2006, pp. 1263–1272.
6. Forlines, C., Wigdor, D., Shen, C., Balakrishnan, R. (2007). *Direct-Touch vs. Mouse Input for Tabletop Displays*. In Proceedings of the 2007 CHI Conference on Human Factors in Computing Systems.
7. Kabbash, P., Buxton, W., & Sellen, A. (1994). *Two-handed input in a compound task*. ACM CHI. pp. 417–423.
8. Leganchuk, A., Zhai, S., & Buxton, W. (1998). *Manual and cognitive benefits of two-handed input: an experimental study*. ACM TOCHI, 5 (4). pp. 326–359.
9. Mark Hancock, Sheelagh Carpendale, and Andy Cockburn, *Shallow-depth 3d interaction: design and evaluation of one-, two- and three-touch techniques*, In Proceedings of the SIGCHI conference on Human Factors in Computing Systems, April 28–May 03, 2007, San Jose, California, USA.
10. Moscovich, T. and Hughes, J. *Indirect mappings of multi-touch input using one and two hands*, In Proceeding of the twenty-sixth annual SIGCHI conference on Human Factors in Computing Systems, April 5–10, 2008.
11. Oka, K., Sato, Y., and Koike, H. (2002). *Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems*. In IEEE International Conference on Automatic Face and Gesture Recognition, pp. 429–434.
12. Schwesig, C., I. Poupyrev, and E. Mori. *Gummi: a bendable computer*. In Proceedings of CHI'2004. 2004: ACM, pp. 263–270.
13. Sugimoto, M. and Hiroki, K. *HybridTouch: an intuitive manipulation technique for PDAs using their front and rear surfaces*. In Proceedings of MobileHCI '06, pp. 137–140.
14. Wexelblat, A. (1995). *An approach to natural gesture in virtual environments*. ACM TOCHI, 2 (3). pp. 179–200.
15. Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., Shen, C. *LucidTouch: A See-Through Mobile Device*. In Proceedings of UIST 2007, pp. 269–278.
16. Wigdor, D., Leigh, D., Forlines, C., Shipman, S., Barnwell, J., Balakrishnan, R., Shen, C. *Under the table interaction*. Proceedings of UIST 2006 – the ACM Symposium on User Interface Software and Technology, pp. 259–268.
17. Wilson, A. D., Izadi, S., Hilliges, O., Garcia-Mendoza, A., Kirk, D. *Bringing physics to the surface*. In Proceedings of 21st ACM Symposium on User Interface and Software Technologies (ACM UIST), Monterey, CA, USA, October 19–22, 2008.
18. Wobbrock, J.O., Myers, B.A. and Aung, H.H. (2008) *The performance of hand postures in front and back of device interaction for mobile computing*. International Journal of Human-Computer Studies 66 (12), pp. 857–875.
19. Wobbrock, J., Morris, M.R., and Wilson, A. *User-Defined Gestures for Surface Computing*. In Proceedings of CHI 2009, in press. (Best Paper Nominee)
20. Wu M., Shen C., Ryall K., Forlines C., Balakrishnan R. *Gesture Registration, Relaxation, and Reuse for Multi-Point Direct-Touch Surfaces*. In Proceedings of IEEE TableTop, pp. 185–192, IEEE Computer Society, Los Alamitos, USA (2006).

**The columns on the last page should be of approximately equal length.**