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

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.

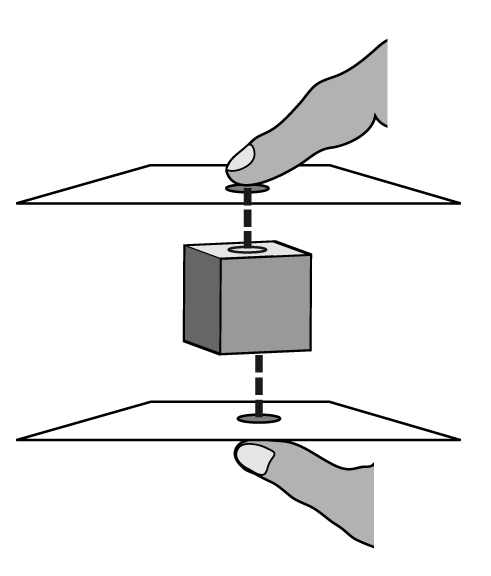
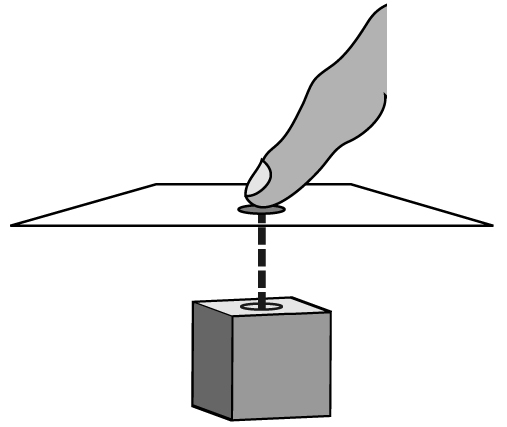


Figure 1: Virtual space between fingers: the object in virtual 3D space is projected to both the front and back surfaces of the device.

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

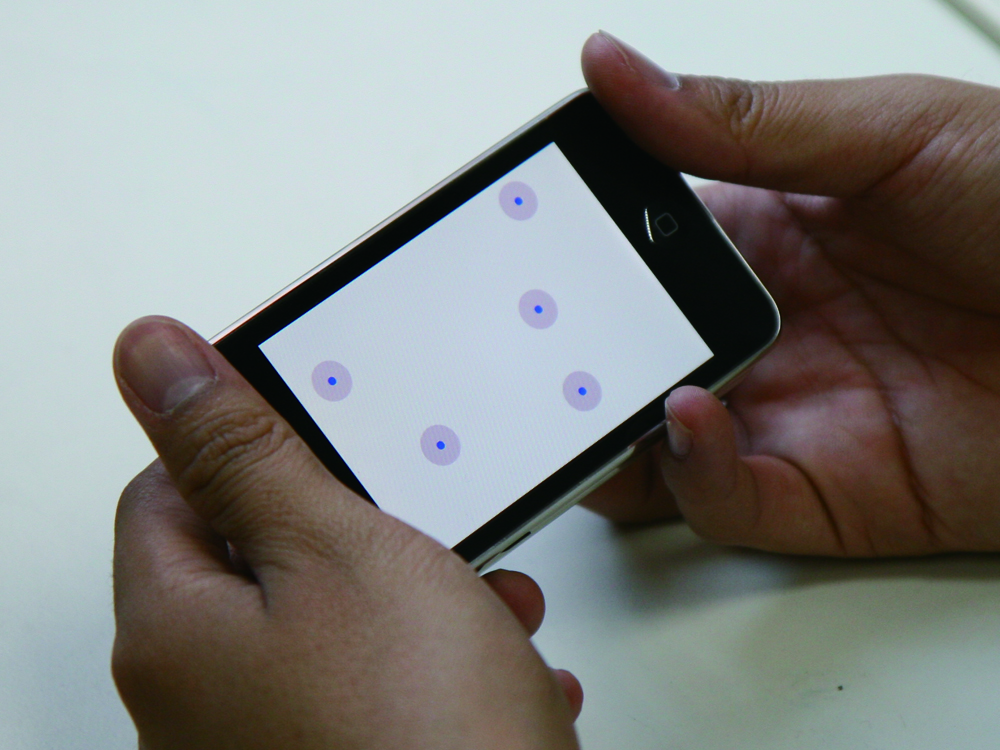


Figure 2: Prototype of our double side multi-touch input mobile device. The location of back side touch points are displayed to provide visual feedback.

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.

## 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.

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

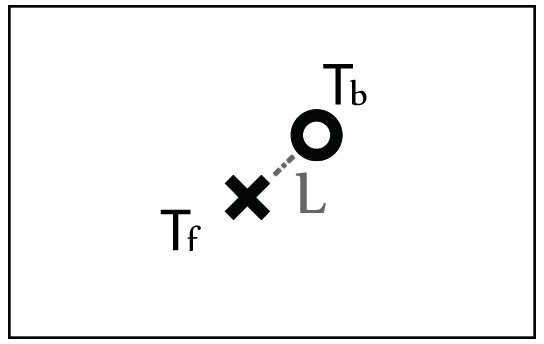
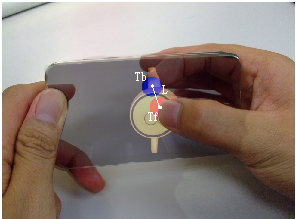


Figure 4: The *Grab* gesture is triggered when the distance between the two touch points on opposing sides is less than a predefined threshold.

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.

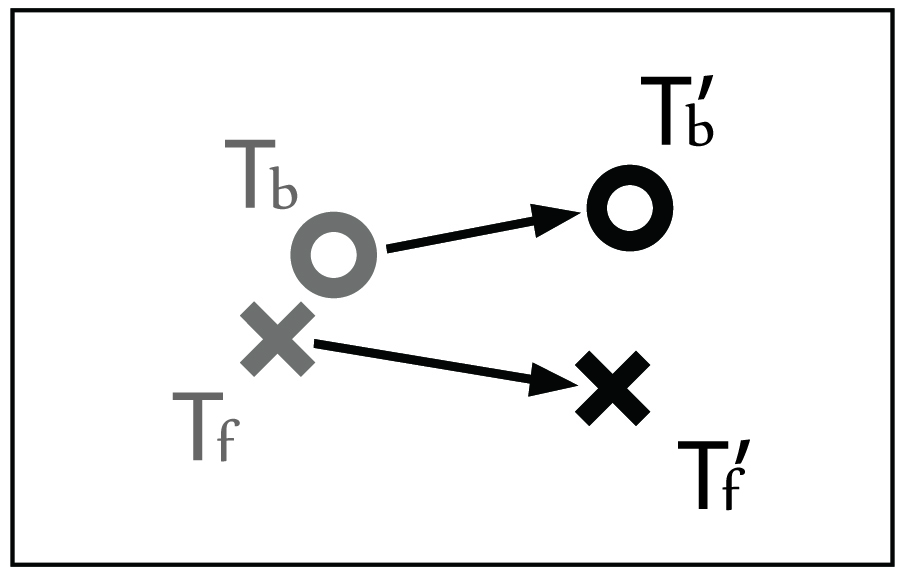
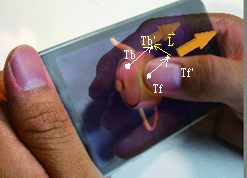


Figure 5: The *Drag* gesture. Touch points and trigger the begin state. and are the touch points after the fingers move. Let ; if *F* is positive, than the *Drag* gesture enters the dynamic state and the object moves along .



### 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.



### Flip

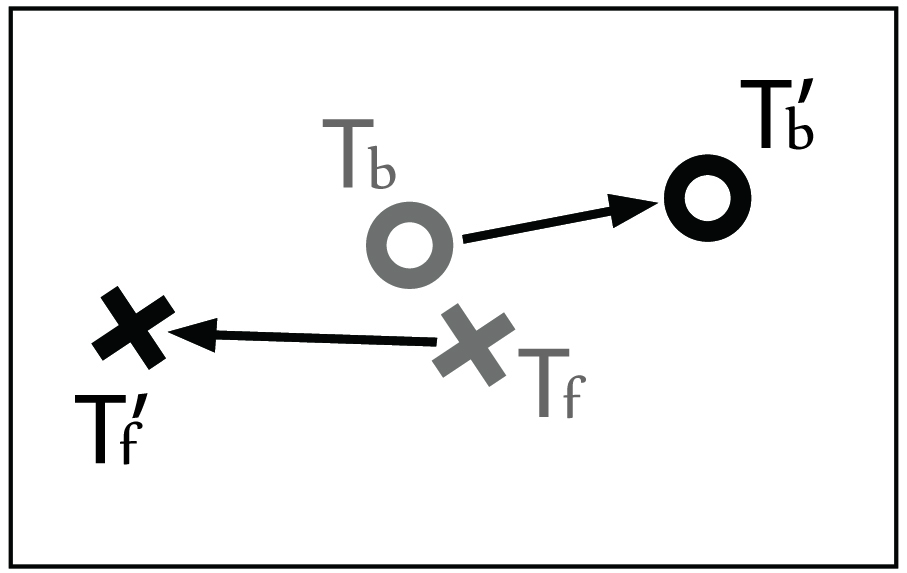


Figure 6: The *Flip* gesture. Let ; if *F* is negative, the *Flip* gesture enters the dynamic state and the object rotates about the axis perpendicular to which passes through the center of the object.



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.

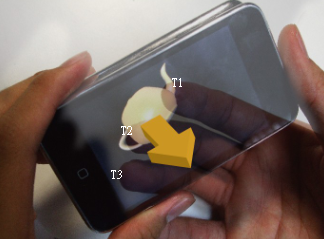


Figure 7: The *Push* gesture. ,, and are touch points from back side. The object moves toward the user when fingers keep touching it; the velocity varies according to number of touching fingers.



### Push

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

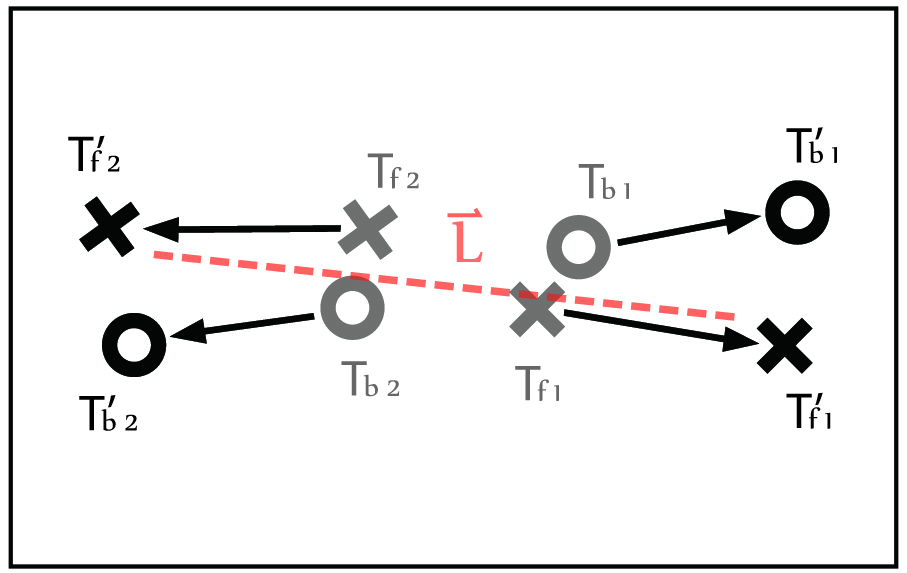
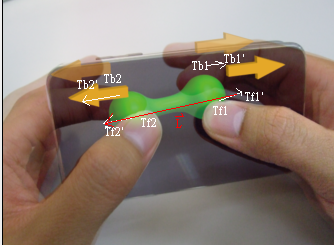


Figure 8: The *Stretch* gesture. Index 1 and 2 represent different hands. Let , the object enlarges/deflates as increases/decreases.



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.

## Visual Feedbacks

We designed several visual feedbacks to help users more easily to understand where their fingers are.

Our front-side visual aid is designed in that the front-side touch points create visible shadows as if the object is pressed by the finger. These shadows let us know where is the exact area we can select objects when our thumbs touch front side screen.

Furthermore, since users cannot see fingers’ positions on the device’s back-side surface, it is essential to provide users with visual feedbacks on the front-side screen that shows the fingers’ positions on the device’s back-side. Such visual feedbacks can significantly reduce the difficulty for users to move their fingers to a target point on the device’s back-side surface. We 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 gestures such as grabbing or dragging. They correspond to the cursor on personal computers; targets are not occluded because each dot is only six 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.

# 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 task 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.

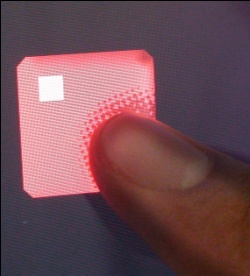
The pilot users could not *Grab* the objects easily in this study. We used the touch point of back-side finger as the index to select objects on the screen in this design. When the objects went small, it was hard for users to *Grab* it successfully at first time. On the other hand, two of our users considered 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.

# USER STUDY

We conducted a formal user study to evaluate usability of double-side gestures in 2D/3D object manipulation. The following inquiries guided this user study:

* How does the usability of the double-side gestures (grab, drag and stretch) compare with the usability of the equivalent front-side-only and backside-only gestures?
* What effect does the target size have on the usability of the double-side gestures in comparison to the usability of equivalent front-side-only and backside-only gestures?



(a) (b)

Figure 9: Visual feedback in our design. (a) Front-side visual aid. (b) Back-side visual aid.

All the tasks were tested for two-handed manipulation. We wanted to simulate the situation when users hold the device with horizontal direction. More and more games or applications on mobile devices are designed for operating with two hands and in landscape mode. Therefore, in our following experiments, users could just only use their thumbs to manipulate the objects on front-side and they can use other four fingers for the back-side manipulation.

## Participants

Ten student participants (5 males, 5 females) from our university were recruited as participants in the study. Half of them were students in the EE/CS departments, and the other half were students outside the EE/CS departments, including marketing, management, and literature. Half of them had prior experience of using multi-touch input on mobile devices, and the other half had never used any multi-touch input interface prior to this user study. All of them are right-handed. Ages ranged from 19 to 22.

User study was conducted in our research lab, during which each participant was given an office chair and a desk. They could choose to sit in any comfortable way. The test took around 45 minutes per participant.

## Apparatus

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

Each experiment consisted of three phases: (1) demonstration phase, (2) familiarization phase, and (3) experimental phase. In the demonstration phase, we explained the device operation to the participants who were also shown how to perform different finger gestures on the double-side, front-side-only, and backside-only input devices. In the familiarization phase, participants were given a set of finger gesture tasks for practice. The practice tasks were the same as those used during the experimental phase, except that no measurement was recorded. Given varying skill level and learning time, participants were given the flexibility in their practice times which ranged from 5 to 10 minutes. Note that participants were only allowed to ask questions during the demonstration and familiarization phases but not during the experimental phase when asking questions would adversely affect the usability measurement.

During the experimental phase, each participant performed three sets of tasks using different finger gestures on each of three input surfaces. Performance logs were automatically recorded for analysis and comparison.

## Measures

For usability evaluation, the primary measure was the *task completion time*, which is defined as the time it takes a participant to perform a given task using a specific finger gesture on a specific input surface. The secondary measure was *extra movement*. Since all tasks in this user study were designed to require only one single finger gesture movement, extra movement counts the number of extra (i.e., more than one) finger actions per task. For example, if it took two finger actions to complete a task which only need one finger, the count of extra movement would be one. Extra movement is a good approximation on the amount of input error, because in general case

# Grab Experiment

Grab gesture is used for selecting an object. The Grab gesture for Backside-only input involves an index finger tapping once on the object from the back surface of the device. The Grab gesture for Front-side-only input requires a thumb tapping once on the object from the front surface of the device. We limited users to use thumbs only since the environment we want to test is holding device in horizontal direction.

Figure 10: The mean Extra Movement for InputSurface X TargetSize in Grab Experiment.

The independent variables in this experiment were the input surfaces (*InputSurfaces*: Both-side, Front-side-only and Backside-only inputs), and the size of target object (*TargetSize:* 10, 8, 6, 5, 4, 3, 2 mm). For each target size on a given input surface, four rounds of the same object selection manipulations were repeated. In each round, the test program generated a yellow circular object at a random position. Participants then select the object to complete this task. A screen shoot is shown in Figure 9. Each participant performed the object selection task a total of 84 times, or 3 (input surfaces) \* 7 (target sizes) \* 4 (repeated rounds).

## Grab Results

Figure 9: The mean Task Completion Time for InputSurface X TargetSize in Grab Experiment. In this and all later charts, error bars represent standard deviation.

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

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 input was significant faster than Front-side-only input (p<0.0001) and that Backside-only input was significant faster than Front-side-only input (p<0.0001). No significant result was found when *TargetSize* was 10, 8, 6, 5, or 4 mm.

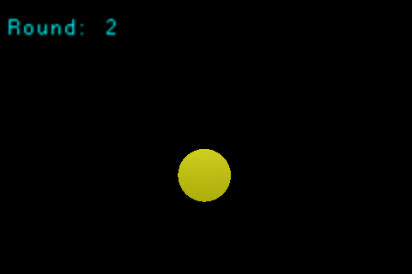


Figure 11: The scene of the *Grab* experiment

Figure 11 plots the mean and standard error of extra movements from the ten participants on each *InputSurface* and *TargetSize* in the object selection experiment. 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.

Grab results show that Both-side input becomes increasingly faster than Back-side-only and Front-side-only inputs with shrinking target size. Grab results also show a growing number of extra movements for the Front-side-only input as the size of target object shrinks. This suggests increasing difficulty for participants to precisely aim their thumbs on smaller targets on the Front-side-only input. In other words, participants had to try many times before they succeeded in selecting small targets, leading to slower task completion times and increased numbers of extra movements.

Although grab results for the completion times and extra movements do not show a significant difference between Backside-only and Both-side inputs, Both-side input slightly outperforms Backside-only input on the average task completion time and the average numbers of extra movements for all *TargetSizes*.

# Drag Experiment

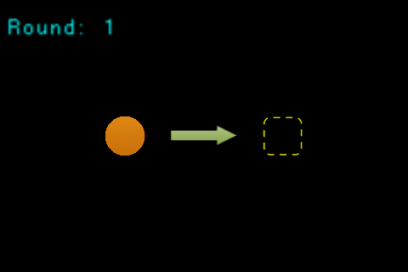


Figure 15: The scene of the *Drag* experiment

Drag gesture is used to move an object. The Drag gesture for Both-side input is defined previously. The Drag gesture for Backside-only input involves tapping once on the object with an index finger on the back surface of the device and then sliding the finger to move the object to a target position. The Drag gesture for Front-side-only input requires tapping once on the object with a thumb on the front surface of the device and then sliding the thumb to move the object to a target position.

The independent variables in this experiment were *InputSurfaces* (Both-side, Back-side and Front-side inputs) and dragging *Directions* (8 directions, E, NE, N, NW, W, SW, S and SE). For each dragging direction on a given input surface, four rounds of the same object drag manipulation were repeated. In each round, our testing program generated a yellow circular object. Participants first select the object, and then move it to a target area to complete this task. A screen shoot is shown in Figure 12. Each participant performed the object dragging task a total of 96 times, or 3 (input surfaces) \* 8 (directions) \* 4 (repeated rounds).

Figure 12: The mean Task Completion Time for InputSurface in Drag Experiment.

Figure 13: The mean Extra Movement for InputSurface in Drag Experiment.

## Drag Results

Figure 14: The mean Task Completion Time for InputSurface X Direction in Drag Experiment.

Figure 13 shows the average task completion time of the ten participants on each *InputSurface* in the Drag experiment An ANOVA test showed a main effect for *InputSurface* (F2,957=15.631, p<0.0001), but not *Direction*. *InputSurface* X *Direction* showed no significant interaction. Tukey’s pair-wise comparison showed that Front-side-only input was significantly faster than Both-side and Backside-only inputs (p<0.01).

Figure 14 shows the mean and standard error of extra movement of ten participants on each *InputSurface* in the Drag experiment. An ANOVA was then performed for *InputSurface* X *Direction* and showed a main effect for *InputSurface* (F2,838=35.665, p<0.0001). There was no significant interaction in *InputSurface* X *Direction*. Tukey’s pair-wise comparison showed that the *Extra Movement* of Backside-only and Front-side-only inputs were significantly less than Both-side input (p<0.01).

There was a significant result which Front-side faster than the others in completion time but 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.*

No significant result was shown in *Directions*. However, figure 15 shows that in Both-side gesture, some directions (NW or SW) do have a little effect on the mean of completion time and *Extra Movement* in different *InputSurfaces*. 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.

# Stretch Experiment

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.

Figure 16: The mean Task Completion Time for InputSurface X ScaleType in Stretch Experiment.

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 *InputSurfaces* (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 (input surfaces) \* 2 (scale types) \* 9 (repeated rounds) = 54 trials in this experiment.

## Stretch Results

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: *InputSurfaces* (F2,537=7.446, p<0.001), and *ScaleType* (F1,538=26.589, p<0.0001). *InputSurfaces* 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).

Figure 17: The mean Extra Movement for InputSurface X ScaleType in Stretch Experiment.

Figure 18 shows the mean and standard error of *Extra Movement*. An ANOVA was then performed for *InputSurfaces* X *ScaleType* and showed that: *InputSurfaces* (F2,537=17.736, p<0.0001), and *ScaleType* (F1,538=12.855, p<0.01). *InputSurfaces* X *ScaleType* was significant interaction (F2,538=15.132, p<0.0001). Tukey’s pair-wise comparison showed that Back-side-only and Front-side-only input were significant faster than Both-side input (p<0.01).

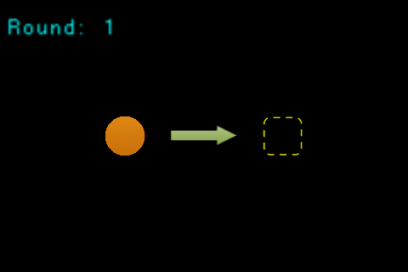


Figure 18: The scene of the object selection experiment

Figure 19: Subjective agreement of each experiments in our user study.

Back-side-only input was significant faster than Both-side input, but there was no significant difference between Front-side-only input and Back-side-only input. Like the result in experiment 2, Front-side-only and Back-side-only need fewer fingers to complete the gesture. Both this two input method used two fingers, and they were relative but not absolute moving actions. The participants can use them easily without interruption. In addition, expanding and shrinking gestures in single surface 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.

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.

# Subjective agreements

Figure 19 shows the average ratings (a higher score means better preference), calculated from the result of questionnaires filled out by participants, for each of three input surfaces (Both-side, Back-side and Front-side) in selecting, moving, and changing the size of an object. On intuitiveness, participants preferred Both-side input over Back-side-only and Front-side-only inputs. Although Both-side input did not have the best quantitative results (i.e., the task completion time and the extra movement) in the Drag and Stretch experiments, qualitative results from subjective agreements showed that participants appreciated its intuitiveness. On ease-of-use and ease-of-learning, participants found no significant differences among Both-side, Back-side, and Front-side inputs.

Our explanation for the results is that the majority of people used to be familiar with front-side touch screen. Therefore, not surprisingly, they believe that it is much easier to use. However, double side gestures are so intuitive that they can easily remember these gestures and thus they considered these gestures were easy to learn. While in the Stretch case, although users felt it was intuitive, the both-side gesture required user to use four fingers and thus they generally believed that it was difficult to learn.

# people are aDiscussion

Our findings show that when performing actions on the *xy*-plane (e.g., Drag and Stretch), participants preferred single-side inputs **(single-side or front-side?)** over Both-side input because these actions involve only one surface of an object and single-side gestures perform faster than Both-side gestures. When performing actions along the *z*-axis or the back-side surface of an object, participants preferred Both-side input over single-side inputs because they found that both-side interaction with virtual objects is closer to the real hand interaction with physical objects. We therefore propose the Hybrid input that combines the advantages of single-side and Both-side inputs. For example, the Hybrid input enables users to switch to single-side finger gestures for *xy*-plane actions and Both-side finger gestures for *z*-axis or back-side actions..

### Unconscious touch points

We observed that participants often hold a device with multiple fingers contacting the device’s back-side surface. These back-side fingers sometimes caused unintended touch points and triggered accidental Both-side or Back-side gestures. To reduce these unintended mistakes, accelerometer sensors can be used to first detect how the device is currently held by hands and then filter out these unintended touch points.

# Conclusion & future work

We have presented several sample finger gestures that use the front, the back, and both 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. For our future work, we want to apply this double-sided multi-touch technique to more commonly-used applications such as text input and precision selection on a small screen device. We also plan to create a lighter and thinner prototype with Both-side input surfaces. .

# 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).