

RESEARCH

Open Access



INSPECT: extending plane-casting for 6-DOF control

Nicholas Katzakis^{1*}, Robert J Teather², Kiyoshi Kiyokawa^{1,3} and Haruo Takemura²

*Correspondence:

katzakis@ist.osaka-u.ac.jp

¹ Graduate School of Information Science and Technology, Osaka University, Yamadaoka 1-1, Suita, Osaka, Japan
Full list of author information is available at the end of the article

Abstract

INSPECT is a novel interaction technique for 3D object manipulation using a rotation-only tracked touch panel. Motivated by the applicability of the technique on smartphones, we explore this design space by introducing a way to map the available degrees of freedom and discuss the design decisions that were made. We subjected INSPECT to a formal user study against a baseline wand interaction technique using a Polhemus tracker. Results show that INSPECT is 12% faster in a 3D translation task while at the same time being 40% more accurate. INSPECT also performed similar to the wand at a 3D rotation task and was preferred by the users overall.

Keywords: Interaction, 3D, Manipulation, Docking, Smartphone, Graphics, Presentation, Wand, Touch, Rotation

Background

Virtual object manipulation is required in a wide variety of application domains. From city planning [1] and CAD in immersive virtual reality [2], interior design [3] reality and virtual prototyping [4], to manipulating a multi-dimensional dataset for scientific data exploration [5], educational applications [6], medical training [7] and even sandtray therapy [8]. There is, in all these application domains, a demand for low-cost, intuitive and fatigue-free control of six degrees of freedom (DOF). Our work focuses on 6-DOF object manipulation at a distance with a large display for presentations and education. Examples of situations where there is a need to interact in 3D from a distance include the following:

Education A professor is demonstrating human anatomy by displaying 3D graphics on a large projector screen. He uses his device to rotate the model and answer questions from the students. The nature of the device allows him to leave the podium and approach the students while still being able to interact with the model, thus making the class more engaging.

Engineering An engineer is showing a 3D model of her latest design to team-mates. The device is used to rotate and translate the model, define slicing planes to inspect the interior and discuss the design with other participants.

Entertainment A group of children are playing a game in a museum while at the same time learning about physics by interacting with wooden blocks on a sandbox-like 3D environment on a large screen.

The design goals of such an interface include the following:

Off-Screen Users would need to maintain a distance from the display so as to not obstruct the view for others.

Without a desk surface or cables Users should be able to move around, approach the display with the controller in their hand (to show an area or point to a feature with their hand).

Without complicated instrumentation or expensive hardware Such a design lowers the barrier for entry, allowing classrooms and meeting rooms equipped with a projector to make more out of their existing setup.

No big arm/hand gestures An interface that is to be used on a daily basis and/or for many hours has to avoid large hand gestures which are bound to induce fatigue [9] and, in rare cases, even cause physical injuries to bystanders.

Simplicity Unlike technology enthusiasts, domain experts or educators often do not have the patience or motivation to learn a new, complicated interface.

Accuracy If the 3D model is detailed, the interface should allow the presenter to bring it closer and make fine adjustments to position and rotation.

This work builds on the work of Katzakis et al. [10] on 3D translation using a tracked touch-panel, motivated by its applicability on smartphones. Plane-casting offers isotonic position control *without* using any external position trackers, save for the orientation sensors in the device. We extend plane-casting and introduce INSPECT, a set of novel interaction techniques for off-screen virtual object manipulation using a smartphone. INSPECT stands for INdirect Six-DOF PlanE Control Technique, and it was designed for the purpose of inspecting a 3D object. We demonstrate that by using INSPECT it is possible, with a low-cost mode change, to perform 6-DOF virtual object selection and manipulation using a 3-DOF orientation-tracked touch panel and the 2-DOF per finger from the touch points.

The wide availability of smartphones and smartwatches motivates the need to explore the *indirect touch* design space and to identify an appropriate way to map the degrees of freedom afforded. We also evaluate INSPECT in a 3D movement task and a 3D rotation task, against a 'gold standard' direct technique with a magnetic tracker, to serve as baseline reference.

Related work

3D Manipulation has been studied in a number of usage contexts: Desktop computing, Virtual Reality using a head mounted display, Immersive large displays/cave systems and Tabletop/Tablet computing where touch is utilised.

Desktop For desktop computing 3D manipulation is typically performed with the mouse. For translation, since the mouse can only control 2-DOF, UI widgets are used for controlling the Z axis and for constraining motion to a certain axis [11]. Keyboard shortcuts are also commonly used (in software like Blender 3D [12]) for switching modes or constraining axes. For rotation, in addition to UI widgets, mapping the 2D motion of the mouse to 3D rotation is also used extensively [13]. The major problem with mouse-based manipulations is that they lack the ability to combine rotation and translation. To address these issues some new desktop devices have been proposed [14–16] but have not met with wide acceptance. These devices offer integral 6-DOF manipulations, but

lack ways to easily change modes (to lock to certain axes etc.) while the long homing time [17] makes it tedious to switch from device to keyboard/mouse to access different modes.

Accessing different modes is a point that will be discussed later on in our work.

Immersive VR In the Immersive VR domain typically a wand is used for 3D manipulation with Scaled HOMER [18] being the latest in a long series of techniques [3, 19–22]. Scaled HOMER is an extension to the classic hand-centered manipulation, HOMER technique [20]. A ray extended from the wand is used for object selection in most immersive VR techniques. A common problem, however, is that they require accurate, 6-DOF tracking of the wand. As such they need complicated instrumentation to set up with either magnetic trackers (Polhemus [23]) or optical tracking (Optitrack [24]). Magnetic trackers are particularly susceptible to interference from the environment while optical tracking systems depend on line-of-sight to the wand which might accidentally be occluded during interaction. They also tend to induce fatigue [25], as the user must keep their hands suspended in mid-air for extended periods of time.

Large displays/cave 3D interaction in front of a large display is not much different to immersive VR with an HMD and some of the techniques used in immersive VR could be applied to large displays. However, the major difference with HMD's and CAVE systems is that interaction in front of a large display often involves more than one user (e.g. a presenter and an audience, or two collaborators) and that makes it problematic to track the viewpoint. If the viewpoint cannot be tracked, implementation of ray-based techniques [26] becomes problematic because the ray does not look like it is emanating from the wand. Navidget [27], an alternative to ray techniques for large display interaction, uses 2D input on a tablet to position the camera in a 3D environment. Although this is similar to manipulating an object for inspection, their technique does not directly support object manipulation so it cannot be applied to collaborative scenarios (more than one users could not control the viewpoint simultaneously). The authors reported good usability for both novice and expert users based on questionnaires. Fröhlich [28] presented the cubic mouse, a box with three perpendicular rods passing through its center. The authors report positive reactions from participants, yet the device form factor makes it difficult to relax the non-dominant arm in a presentation as the rods can be accidentally pressed against the presenter's body and thus induce accidental input.

Song et al. used a Kinect to track the user's limbs in front of a large display and proposed a handlebar metaphor [9] for 3D object manipulation. Their users, however, complained about fatigue. In an attempt to address fatigue in large display interaction, Katzakis et al. [26] proposed a set of techniques that allow manipulation by holding the wand at hip height, yet that work, like other ray-based work is hard to implement in a situation where the viewpoint is not being tracked or when it is necessary to interact from the skewed position of a presenter.

Tabletop/touch Touch surfaces share some of the problem of the mouse, being limited to two integral DOF per touch point and typically do not allow for simultaneous translation and rotation. Various attempts have been made to address 3D manipulation using *direct* multi-touch. Reisman et al. [29] presented a multi-touch 3D object manipulation technique that depends on a constraint solver based on the user's perspective. However, their system was not empirically evaluated, and has some drawbacks such as ambiguous

or unwanted rotations. Hancock et al. [30, 31] introduced sticky tools, a technique used to support tabletop 3D object manipulation. Hancock's technique allows simultaneous translation and rotation on a subset of the available axes. Martinet et al. [32] proposed a 3D manipulation technique based on the separation of translation and rotation. Martinet demonstrated benefits from separating translation and rotation in 3D manipulation. Cohé et al. [32] introduced tBox, a 3D manipulation widget for touchscreens. The authors conducted a study using a 3D object assembly task and found tBox was an effective solution. Wilson et al. [33] used physics simulation to manipulate 3D objects with a tabletop display. Tse [6] investigated the use of touch and tangible objects for 3D object manipulation in education (i.e. presentations). They found that participants struggled with camera positioning but appreciated touch-based object rotation.

The aforementioned 3D interaction techniques that utilise *direct* touch are all limited by display size. When the display exceeds a certain size threshold, touch input starts to become cumbersome. The user is required to cover a large area with physical movements and parts of the display area are out of arm's reach. This is the case with large tiled displays, or those using projectors. In addition, physically approaching the display limits the user's activity to a very small area while in collaborative systems the interacting user obstructs the view for the rest of the group.

Indirect touch is ergonomically superior to direct touch while overcoming the aforementioned collaboration problems. Conversely, with indirect touch there is no cursor for selection [34]. In an attempt to overcome this DOF limitation Ohnishi et al. [35] used two touch pads resting on a desktop 3D selection and annotation scenario. Finally, Wigdor et al. [36] proposed a set of techniques that employ shaped touches to control gain, overcome occlusion avoidance, and manage separation of constraints in a 2D task. Their approach, however novel, requires a tabletop and would be hard to implement on a large screen interaction scenario.

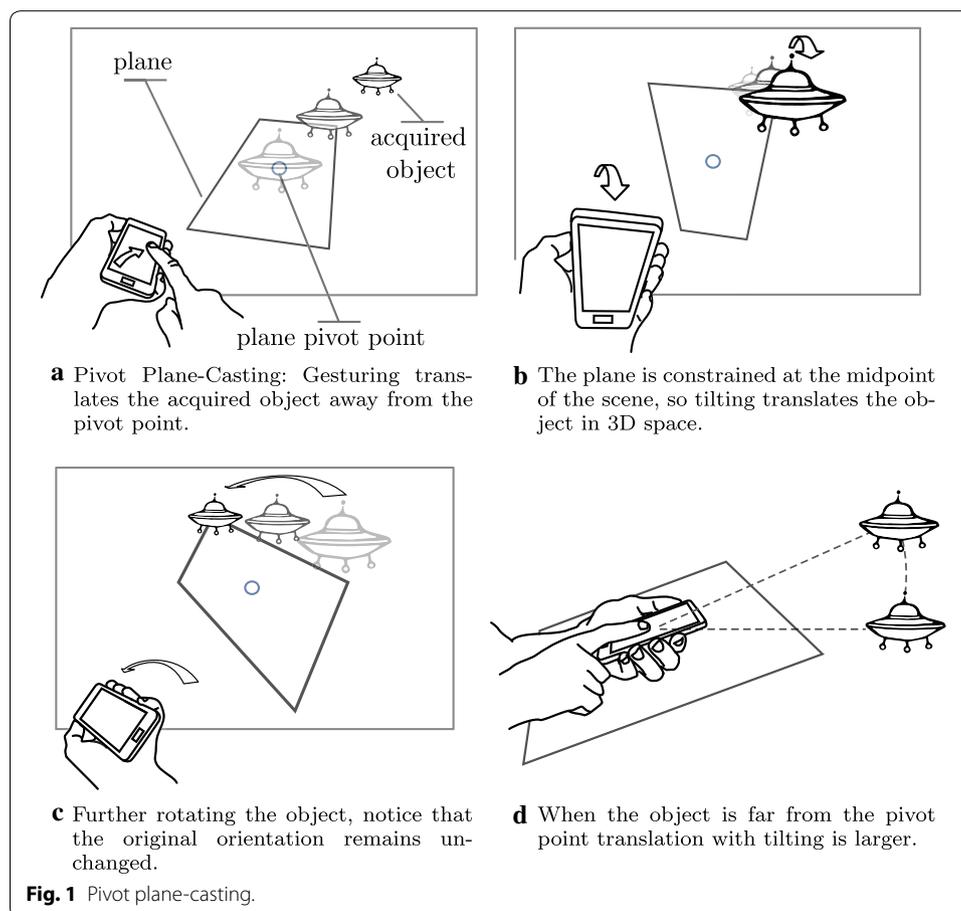
Proposed techniques

Plane-casting

The original plane-casting technique [10] supported 3D positioning of a pre-selected object. The core idea of *plane-casting* was that the manipulated object was free to move in 2D along a plane that was freely oriented in 3D space by the user (refer to Additional file 1: Video). Two variants of plane-casting were proposed:

In the first variant, *Pivot* plane-casting, the plane rotated about a pivot point located in the center of the 3D space. The orientation of the smartphone controlled the orientation of the movement plane about the pivot point. Swiping on the display translated the object in the corresponding axis on the movement plane. Thus, by translating the object on the plane away from the pivot point and then rotating the plane, the object could be positioned at any point (Fig. 1). A disadvantage of *Pivot* plane-casting is that it requires a "clutch" button to disable plane rotation. Without this capability, users would have to always hold the device at a fixed orientation to stabilize the object's position. They would thus be unable to relax their non-dominant hand (Fig. 1d).

The second variation, *Free* plane-casting, is similar to *Pivot* plane-casting in that swiping on the touch surface translates the object on the movement plane. The primary difference between the techniques is that *Free* plane-casting also translates the plane and



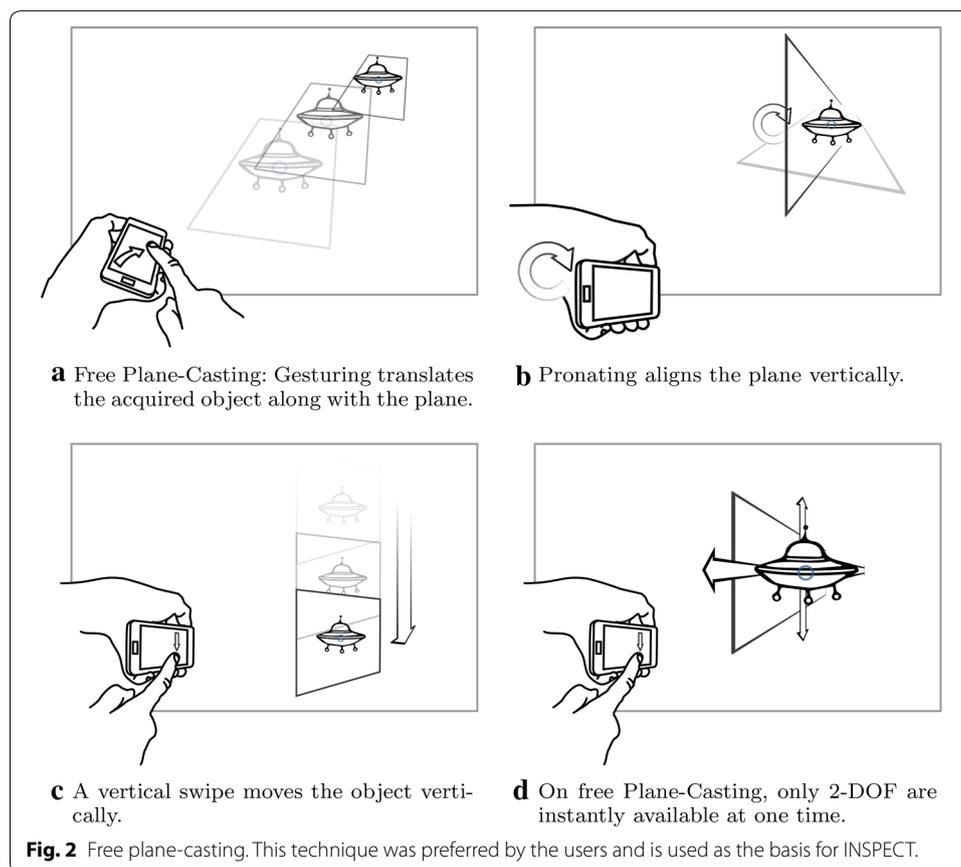
its pivot point along with the object (Fig. 2). In a sense, the object and the plane are “interlocked”, and move together in 3D space, always in the direction afforded by the plane’s orientation.

The two plane-casting variants offered comparable quantitative performance. However, participants strongly preferred *Free* plane-casting. We hence use this variant as the basis for INSPECT, and simply refer to it as *plane-casting* for the remainder of this article.

INSPECT: design decisions

Like plane-casting before it, INSPECT was intended for use with smartphones. Hence we carefully considered the capabilities of these devices in implementing INSPECT. Smartphones typically provide 3-DOF rotations from the combination of accelerometer, magnetometer and gyroscope, 2-DOF per finger from the touch-screen (usually with 2–3 fingers), and volume-up and volume-down buttons. Although some devices provide additional inputs, we used this minimal subset as these are the only universally available input streams on smartphones.

INSPECT was designed based on Jacob’s findings on performance gains when “*the structure of the perceptual space of a graphical interaction task mirrors that of the control space of the input device*” [37]. This is indeed the case with INSPECT, which in addition to having a good perceptual match (smartphone orientation matches that of



the movement plane), benefits from the use small muscle-groups [38], since most of the work is done by the fingers. Finally, the technique works with the dominant hand fingers interacting very close to the off-hand palm. This establishes a frame of reference for the dominant hand to manipulate against [39]. This design leverages the benefits of bimanual interaction that have been repeatedly discussed in the literature [40].

We also wanted to allow users to use the technique while looking directly at the large display, without having to look at the device. During presentations, the presenter's gaze guides the audience and if the presenter were to look at his device screen to manipulate it would create a disconnect with the audience. This also allows the presenter to interact in a natural standing pose, with their arms resting by their torso while the device is supported by both hands. Holding the device near the torso was a key design point for fatigue-free interaction. In contrast, many wand or gesture-based techniques require the user to hold the device with the arms extended, which induces fatigue. Finally, a small device held with the non-dominant hand gives users the freedom to point to the display with their dominant hand between manipulations. This is essential during presentations, for example.

Extensions to translation mode

To improve object translation, we added a "flick" gesture to plane-casting. This allows the user to launch the object inertially in the direction of the flick. In position-tracked wands,

controlled by the arm, flicking motions are not so easy to perform because flicking requires a rapid acceleration of the wand. Such an accelerated motion is less than trivial to perform, and gets even more difficult when repetition is required. A finger gliding on a touch-surface, on the other hand, lends itself well to flicking. Flicking provides an alternative to the gain functions often used in 3D user interfaces to scale input [18]. Moreover, inertial flicking is often used in smartphone UIs for scrolling and other tasks. Consequently, we expect that smartphone users will be able to adopt flicking quickly due to its familiarity. Much like smartphone UIs, touching the touchscreen after flicking an object stops its movement. While inertial flicking has been explored previously for direct touch techniques [32, 33], and for 2D graphics [41] to the best of our knowledge, it has not been used with off-screen touch for 3D manipulations. To translate using flicking, the same gesture is used, as in plane-casting. When the finger is lifted following a gesture, if it has crossed a certain speed threshold, the object is launched inertially with flicking.

Translating objects with plane-casting required repeated supination/pronation motions for fine positioning orthogonal to the movement plane. We expected that this may frustrate users. Consequently, we added pinch gestures to move the object along the current movement plane's normal vector. Pinching the fingers *away* translates the object parallel to, and in the direction of the plane normal. Conversely, pinching the fingers *together* (or "un-pinching") translates the object in the opposite direction. When holding the device upright, this mapping is similar to that used by Sticky Tools [31] or most touch interfaces where pinching away brings the object closer to the surface (zoom). This yields vertical motion relative to the movement plane, and could be useful in visualisation applications where the plane acts as a slicing plane. We refer to this mode as *pinch translate*.

Extensions for rotation

In addition to the translation extensions discussed above, we added a new mode to enable rotation. Several tabletop systems use rotation techniques where the fingers directly touch the manipulated object and/or the display surface [6, 30, 31, 33]. However, we propose off-screen rotation using indirect touch which has not been explored previously in 3D graphics. The smartphone's volume-up button switches the system to rotation mode while being held pressed.

Including an explicit mode change for rotation might initially appear cumbersome. However, we argue that low-cost mode changes do not introduce a high cognitive demand. For example, the smartphone's volume buttons are available at natural grip positions. Pressing these buttons has a very low cognitive cost, similar to the "shoulder" or trigger buttons on modern game controllers. In addition, thumb pressure is counteracted by the forefinger when pressing these buttons, so the mode change has minimal (if any) effect on the movement plane orientation.

There is another interesting side effect of holding a button to switch between the translation and rotation modes. It is possible for experienced users to use the inertial flick feature, switch to rotation mode, and rotate while the object is still flying. While a form of simultaneous rotation and translation is also possible with Sticky Tools [31] and Wilson's work [33], we argue that this is easier with an explicit mode change and that

integrated translation and rotation modes might lead to accidental input or unpredictable/irreversible motions.

The rotation modes are straightforward:

Horizontal finger motion (on the touch-screen X axis^a) rotates the object about the world Y axis (Fig. 3). Vertical motion (device Y axis) rotates about the world X axis. This mode provides integral rotations on the X and Y axes that are performed with a single finger and will be referred to as *XY rotate*. *XY rotate* should not be confused with ARCBALL [13] despite the similarities. ARCBALL uses a function to project the 2D touch points onto a virtual sphere whereas *XY rotate* simply converts translation of the touch point to rotation. *XY rotate* thus exhibits a distinctly different behavior to ARCBALL. *XY Rotate* is a form of two-axes valuator [42] implementation for indirect touch.

To rotate the object about the Z axis we use two fingers which are pivoted about their midpoint. If the two fingers are moved in parallel, their motion is interpreted as a single touch point which induces the same rotation as *XY rotate*. This feature allows minor corrective adjustments to the X and Y axes while rotating about the Z axis without requiring lifting a finger from the screen or further mode changes. This mode will be referred to as *Z+XY rotate*. The *Z+XY rotate* mode is only possible because INSPECT is based on indirect touch. In direct touch systems the parallel motion of the two fingers is usually mapped to translation [31]. As such, a three axis integral rotation mode is, to the best of our knowledge, unique to our system. Users can also make fluid transitions between single finger and two finger rotations as desired. *Z+XY rotate* feels similar to rotating a physical trackball yet is different from Arcball+ by Rousset et al. [43]. Arcball+ uses the midpoint to rotate like the classical ARCBALL [13] algorithm. We avoided this approach because ARCBALL is known to affect the Z axis as well.

In any of the rotation modes, the orientation of the device is ignored. Rotations are always performed as if the device was held vertical facing the display.

During pilot testing, participants indicated that rotating unfamiliar objects without an obvious “up” orientation did not present any problems. However, rotating objects that

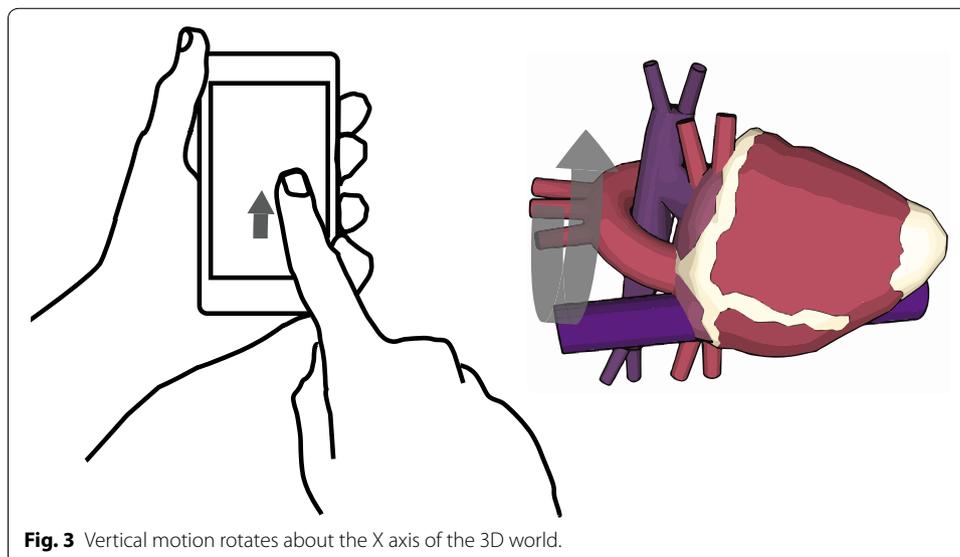


Fig. 3 Vertical motion rotates about the X axis of the 3D world.

had a clear “up” direction required more controlled rotations. For example, when rotating the human heart (a comparatively unfamiliar object), participants were happy with their rotation, even if the heart was slightly tilted off the Y axis. In contrast, when rotating an office chair (a familiar object with a clear “up” direction) participants would try harder to ensure the orientation was absolutely correct. For this reason, we decided to add single-axis constrained rotations.

Single-axis constrained rotations are activated by touching the display corners. Analysis of our pilot study touch data revealed that users rarely reach the touchscreen corners while moving objects with plane-casting (Fig. 4a). Consequently, we decided to use the screen corners for explicit rotation mode changes. The natural shape of the hand allows for a stationary finger in the screen corner, while another finger moves freely to control one DOF (Fig. 5). Thus, we introduced the following rotation mode changes depending on the touch point of the first finger to touch the display. Fingers are obviously not detected, but we make recommendations on which finger to use for better ergonomics:

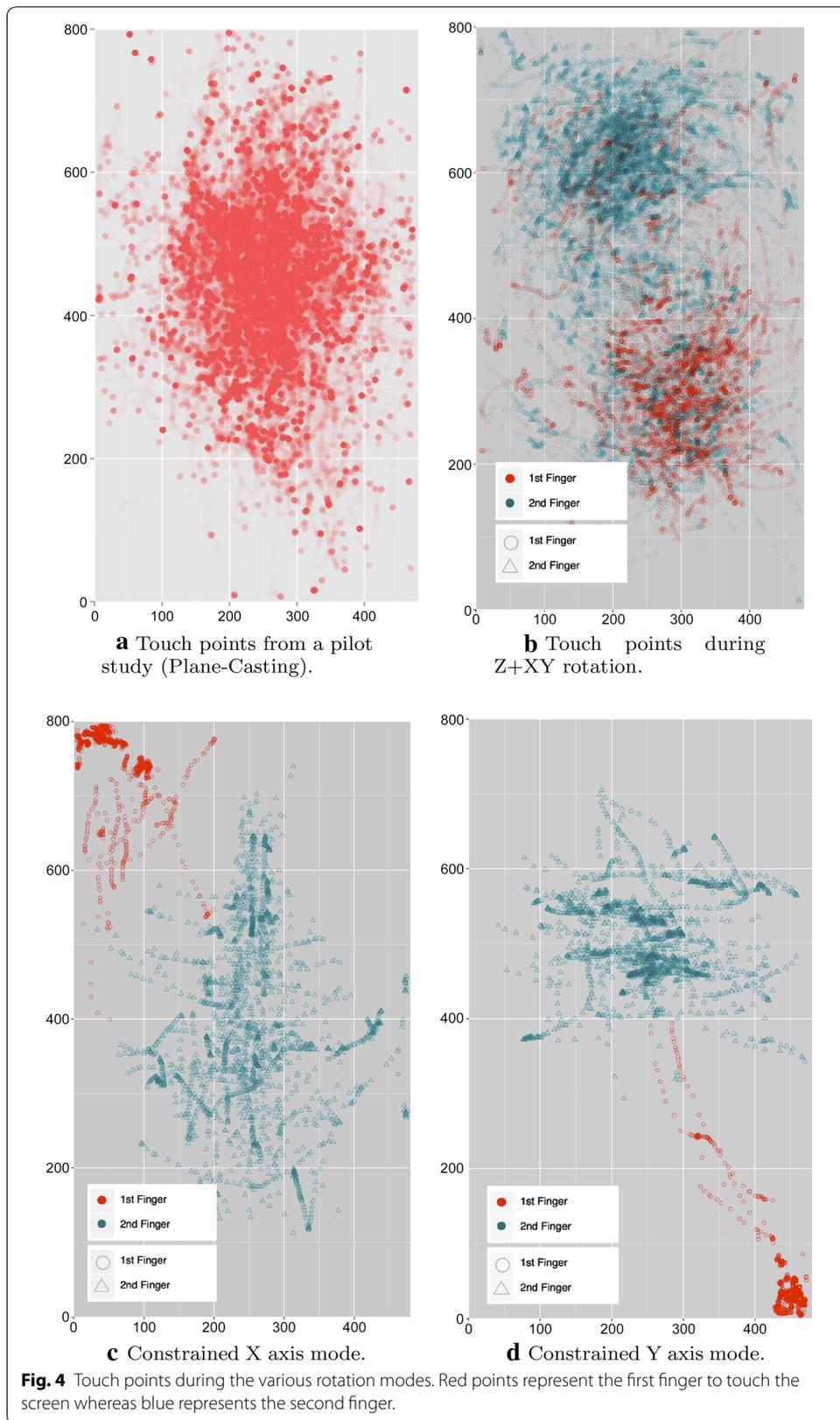
(X) The forefinger on the top-right corner constrains rotation about the display’s Y axis. The thumb is used to control rotation. (Fig. 5). (Y) A thumb on the bottom-left corner constrains rotation to the display’s X axis with the forefinger is used to control rotation. (Z) A thumb on the bottom-right corner constrains rotation to the display’s Z axis. The forefinger is used to control rotation.

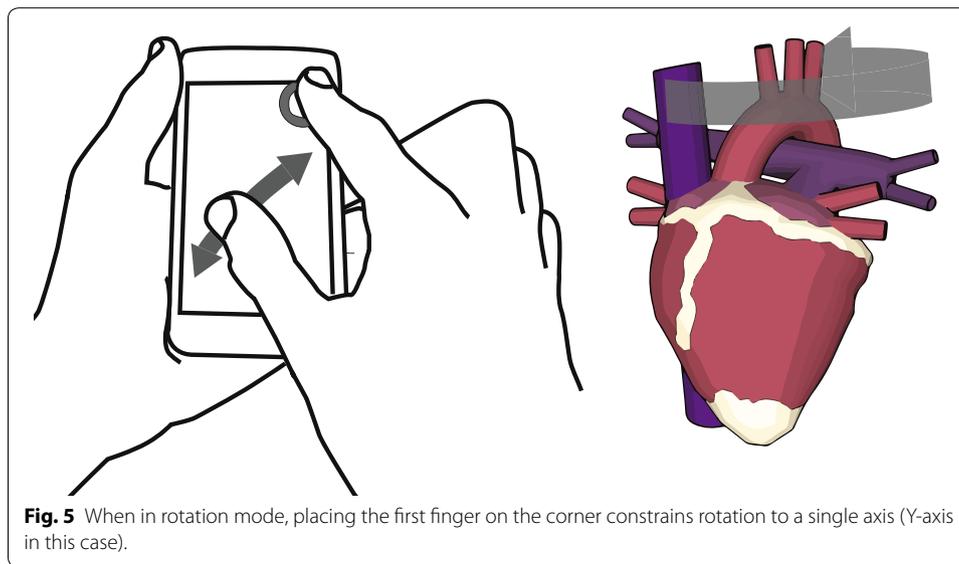
For example, touching the top-right corner of the touchscreen activates Y axis constrained rotation mode (Fig. 5). The thumb’s vertical motion on the touchscreen is ignored and only the horizontal component rotates the object about the constrained Y axis.

Using axis constraint mode requires the first touch to be near the corner. The finger is subsequently free to roam the touch-screen so long as it remains touched. This was intended so that the initiating finger doesn’t impede the motion of the second finger, which actually performs the rotation. This novel way of accessing additional modes by placing a touch on the corners is a feature unique to INSPECT, and could be further extended for accessing additional modes with double-taps, swipes, etc. (Additional file 1: Video)

We decided against using more than two fingers for switching modes or added functionality. Unlike tablets, where users have access to a large surface, small smartphone touch screens cannot easily accommodate many fingers. Similarly, using more than two fingers precludes implementing our technique on *very* small touch screens, such as those found on smart watches. Similar to the translation mode, we added inertial flicking to all three axes in rotation mode. Similar to the translation mode, if a finger exceeded a certain speed threshold, the system kept the object spinning about its center with a fixed decay rate. This entire set of indirect touch rotation techniques will be referred to as *touch-rotate* in the evaluation section.

Finally, we also supported direct rotation using the smartphone orientation sensors. Direct rotation was activated by double-tapping and holding the volume-up button. Rotation was relative to the device orientation at the time of the button press. This mode allowed users to clutch to avoid strenuous wrist positions. This technique will be referred to as *phone-inertial* in the evaluation. INSPECT’s inertial rotation mode is similar to that of the Flying Mouse [44]. However, the designers of the Flying Mouse made an unexpected





design choice. They require users to keep their thumb on a UI widget on the touch screen to access various modes (including the inertial rotation mode). In addition to having to aim (and keep the finger stationary) to access the mode, when the device is rotated the user can no longer see the UI widget and is thus difficult to know if he/she is pressing it correctly. Finally, such a design choice makes it difficult to do the rotation bimanually.

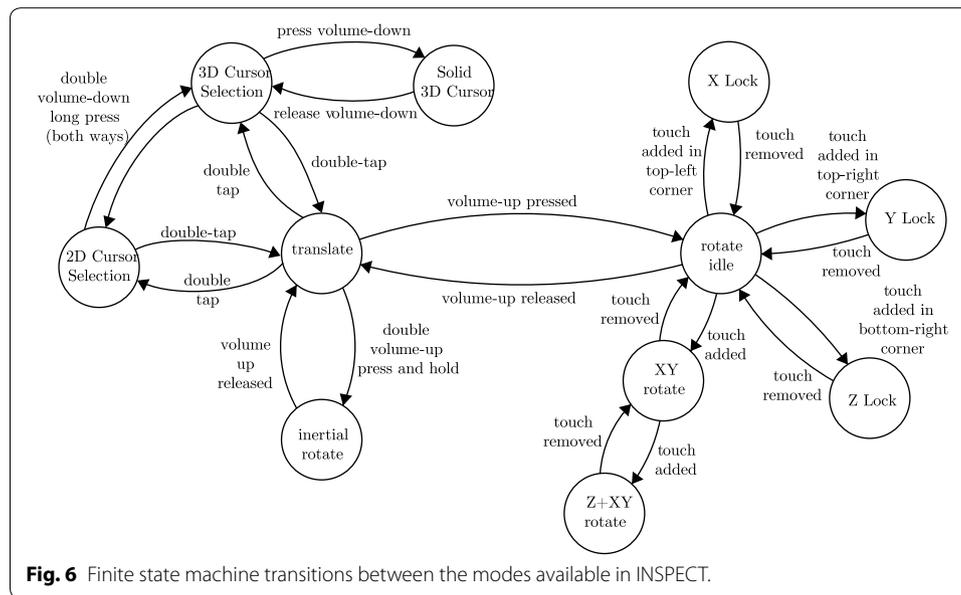
It should be noted that some state-of-the-art *direct* touch techniques (like tBox [32] and Sticky Tools [31]) which are designed for tabletops, can be adapted for use on vertical or handheld touch displays. Such techniques are not mutually exclusive and could complement INSPECT, depending on how close the user is to the display.

Selection

We also considered two object selection modes for use with INSPECT. The first used a relative 2D cursor. In this mode, one finger moves the cursor relative to its current position, similar to the trackpad commonly found on notebook computers [41]. Objects under the 2D cursor are highlighted and touching the screen with a second finger or double-tapping selected the object. We also prototyped a *virtual hand*-like selection mode. In this mode, a spherical cursor (virtual hand) is controlled by plane-casting and intersects the desired object. Either of these two selection modes could be applicable depending on the context of usage. We found the former to be quicker, yet the latter offers the potential to select occluded objects or “nudge” objects to reveal the desired one (e.g. in a 3D sandbox game/educational application with collision detection). A double-tap and long press on the volume-down button switched between the 2D and 3D cursors. When in 3D cursor mode, holding the volume down button made the 3D cursor solid for bumping against other objects (Fig. 6). Neither of those selection modes was formally evaluated because there is extensive literature on 3D selection.

Evaluation

There is no widely accepted standard for off-screen 6-DOF control and the Flying Mouse [44] for the iPhone only works with companion apps. We thus elected to compare against



a ‘gold standard’ technique using a 6-DOF Polhemus sensor. Direct manipulation using a tracked wand is widely used and accepted as being an intuitive way of interacting with a 3D object. For example, similar techniques are extensively used in 3D interaction for visualization and virtual reality [18]. Although we did not expect INSPECT to outperform the wand technique, we thought it necessary to provide this comparison as reference.

Although numerous precision-enhancing techniques have been proposed for direct manipulation techniques, we instead opted for a basic “virtual hand” type technique using the Polhemus-tracked wand. There are three main reasons for this. First, most precision-enhancing techniques require tracking the user’s body [18], which is contrary to our low-instrumentation objective. Second, our intended application scenario assumes a shallow 3D space, which does not require users to position objects so far away (remote positioning was a primary reason why enhanced precision functions were first developed [18, 22]). Finally, using a “raw” direct manipulation technique such as that used in our study makes it easier to replicate the experiment. We used a Polhemus tracker as it should offer better accuracy than other low-cost devices (e.g., the Sony PS Move and PSEye camera).

Participants

Twenty participants took part in the study (within-subjects). Fourteen participants were male. Their ages ranged from 21 to 36 years (mean age 28 years). All had normal or corrected-to-normal vision. All but one were right handed. Participants self-reported regular gaming habits but little to no 3D user interface experience. Participants were compensated with \$20 upon completion of the experiment.

Apparatus

Hardware setup

The experiment was conducted using a PC running Ubuntu Linux with a 2.4 GHz processor and 16 GB of RAM. The graphics card was an NVIDIA 680GT with 2 GB of RAM. A

Sanyo PDG-DWL2500J ultra-short focus projector was used as the display. The projector offers a $1,920 \times 1,080$ pixel resolution at a 16:9 aspect ratio. The display size was 320 cm diagonally, and the display was monoscopic. Participants stood 300 cm from the projected screen. Fig. 7 depicts the equipment setup used in the experiment.

In the INSPECT condition, participants held a smartphone in their non-dominant hand, while touch-gesturing on the smartphone's touchscreen with their dominant hand. The smartphone used for the experiment was a Samsung *Galaxy SII* running Android OS 4.0.3. This device features a 10.5 cm diagonal screen at a $1,280 \times 720$ pixel resolution, a quad-core 1.4 GHz processor, and 1 GB of RAM. The smartphone was connected to the PC via a WiFi network.

In the wand condition participants held a Sony Move controller in their dominant hand. Although the *Move* is normally tracked by the Sony *PSEye* camera, we instead used a Polhemus *Fastrack* receiver for superior tracking accuracy. The control-display ratio was set to a fixed gain (no acceleration) for both translation and rotation in this condition. Participants held the top button (thumb) on the wand to activate object translation and the rear trigger button (forefinger) to activate object rotation. Both translation and rotation were relative to the position/orientation of the wand upon pressing the button.

A 3Dconnexion *Space Navigator* was also used as a foot switch. The sole purpose of the device was to advance trials.

Software setup

The experiment used custom software running on the PC, which was written in C++ and OpenGL 3.3. Custom software on the smartphone was written in Java and communicated with the PC software via Google protocol buffers for the sensor data and TUIO messages for the touch events. The software presented the experimental tasks described below. In

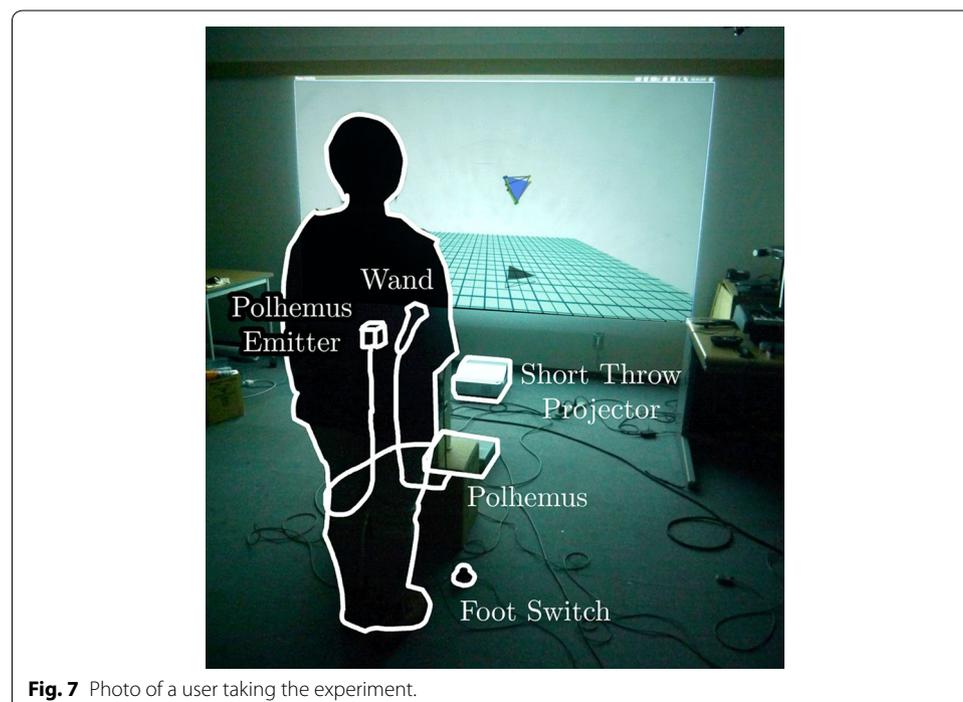


Fig. 7 Photo of a user taking the experiment.

both tasks, the scene was displayed with the target object floating over a “floor”—a flat plane with a cross-hatch pattern textured on it. Shadow rendering was included to help facilitate depth perception. Both the cursor and the target cast shadows onto the floor (Fig. 8). The software also logged the trial length, the cursor position and orientation, touch events on the smartphone, and other relevant metrics.

Procedure

Upon arrival to the lab, participants were given a short briefing about the experiment. This included a verbal explanation of the experiment purpose, the tasks, and a description of the techniques being compared in each task (described in detail below). The experimenter then demonstrated the available control modes and participants were allowed to practice both techniques until they felt confident to begin the task. Following the briefing, participants were asked to perform both the movement task and the rotation task.

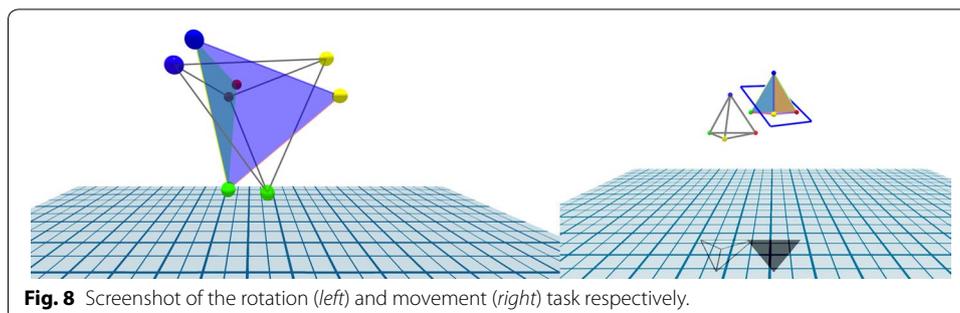
Once participants felt they had a good match to either the target position or orientation (depending on the task), they would press the foot switch to advance to the next trial. Rather than enforcing a fixed “success” threshold, participants were free to judge when the match was accurate enough. This allows data from the experiment to be additionally analysed for 3D pointing (Fitts-law) type studies. Participants were asked to maintain a consistent balance between speed and accuracy throughout the task. They always completed the movement task followed by the rotation task. The motivation to split the 6-DOF docking task to a movement and rotation task was that depending on the form factor of the smartphone, the volume up and down buttons can be difficult to press. We thought that this would introduce a new variable to the completion time and would make it difficult for other researchers to replicate our study. Additionally, because INSPECT is a *set* of techniques, a 6-DOF docking task would not allow us to individually assess the strengths and weaknesses of each sub-technique of INSPECT. Splitting the task in a movement task and a rotation task will potentially reveal weak points of INSPECT and show areas that need improvement.

Specific procedural details for each task are outlined below.

Movement task

The movement task required matching the cursor position to the target position. INSPECT and the Wand technique were compared using this task.

Upon starting a movement task trial, a semi-transparent tetrahedral cursor was already acquired for movement. The cursor appeared in the center of the screen (Fig. 8).



Participants were instructed to move the cursor to match the position of a wireframe target using the current technique (either INSPECT, or Wand). The target was the same shape and size as the cursor. The four corners of the cursor and target were each a different color sphere. The coloured corners were primarily important in the rotation task (i.e., to determine orientation of the cursor and target), but were shown in the movement task for consistency. During the movement task, the cursor and target always maintained an upright orientation so as to rule out effects of rotation during this task. The software presented targets at 12 pre-defined positions, one at a time, randomly shuffled for every participant. Each position corresponded to one of 12 vertices of a regular icosahedron centered at the origin. Each position was tested with two different distances from the origin (same direction, twice the euclidean distance).

Rotation task

The rotation task required that the participant match the cursor orientation to the target orientation (Fig. 8). This was performed independent of the movement task, and was used to compare the two smartphone based rotation techniques (rotation using touch and using the inertial sensors) to direct rotation using the wand.

As in the movement task, the cursor and the target both appeared centered at the origin in the rotation task. Participants had to match the cursor to 12 pre-defined rotations shuffled for each participant, twice. Target rotations were generated in the same pseudo-random order for all participants. The rotation task included two techniques using the smartphone (touch rotation and direct rotation) and direct rotation using the wand.

Design

Due to the differences in the tasks, we present design details for each task separately. Participants always completed the movement task prior to the rotation task. However, all other condition orderings within each task were counterbalanced according to a Latin square.

Movement task

The movement task used a single within-subjects independent variable, movement technique. The two movement techniques compared were wand and INSPECT. For each movement technique, participants performed multiple movement tasks at two different distances in each of 12 directions. Consequently, each participant completed a total $2 \text{ movement techniques} \times 2 \text{ movement distances} \times 12 \text{ directions} = 48 \text{ movement trials}$. Over all 20 participants, this corresponds to a total of 960 movement trials.

The dependent variables for this task were movement time (MT), measured in seconds, and the euclidean distance between the target and cursor centres upon trial completion, measured in cm. This latter dependent variable served as a measure of accuracy. Movement time was measured as the time from when the trial began to the time the participant pressed the foot pedal.

Rotation task

The single within-subjects independent variable for the rotation task was rotation technique. Three rotation techniques were compared: touch-rotate, phone-inertial, and wand

direct. Each rotation technique was evaluated twice with each of 12 randomly generated target rotation angles. This produced a total of 3 rotation techniques \times 2 repetitions \times 12 target orientations = 72 rotation trials for each participant. Over all 20 participants, this yielded 1,440 rotation trials in total. The dependent variables for the rotation task were rotation time (RT), measured in seconds. This was measured as the time from when the target appeared until the time participants pressed the foot pedal upon completing the rotation trial.

Hypotheses

We hypothesized that the wand technique will be faster than the other techniques (*h1*) as it leverages natural movements that participants are accustomed with from their daily life. Also that the wand technique and phone-inertial mode will be less accurate than the touch rotation techniques (*h2*) because holding the wand in a distal position will have an adverse effect. We also believe that the phone inertial mode will suffer from the form factor of the smartphone, which is not ideal for rotations like a fingerball [45] is for example. Finally, that due to the nature of holding the smartphone close to the torso and the wand in a distal position we expected that overall participants will prefer INSPECT and will complain about fatigue using the wand technique (*h3*).

Results

Movement task results

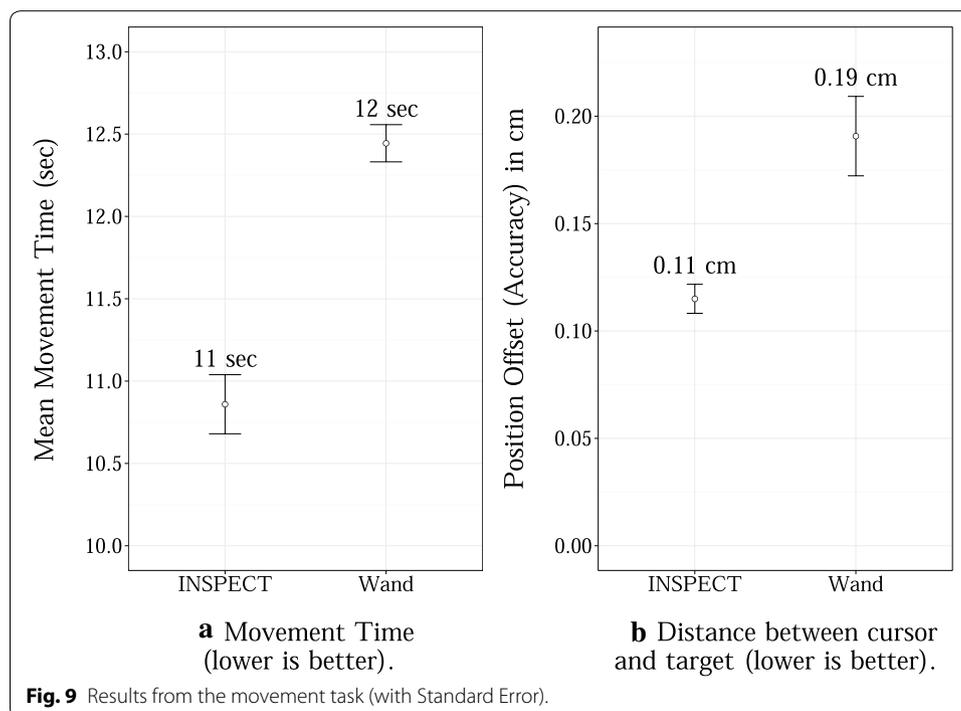
During the movement task there were only two techniques. Consequently, the data were analyzed using a one-way ANOVA. Technique had a significant effect on movement time (MT). ($F = 126.9, p < 0.001$). Participants completed the task 12% quicker using INSPECT (Fig. 9a). This is contrary to our first hypothesis, but encouraging overall as it shows that interaction with INSPECT is quicker than direct manipulation with the wand.

A one way ANOVA revealed a significant effect on accuracy for movement technique ($F = 188.764, p < 0.001$) (Fig. 9b). Participants were more accurate in matching the target position using INSPECT than with the wand—the error distance with the wand was about 40% higher than that of INSPECT. The average distance mismatch for the wand was 0.19 cm with 0.11 cm for INSPECT. This confirms our second hypothesis: (*h2*) INSPECT is more accurate than the wand.

Log file analysis revealed that participants rarely used the flick gestures. Approximately 10% of the recorded frames used flicking. We believe there are two reasons for this. First, participants had no prior experience using the techniques. As such, they did not feel confident launching the object around with inertia-based flicks. Second, the distance to the targets were not long enough to warrant flicking. We believe a different task requiring longer translation distances (e.g., an outdoor AR task) would increase the value of flicking.

Rotation task results

The data from the rotation task was subjected to a repeated measures ANOVA test. Technique had a significant effect on rotation time (RT) ($F_{2,23} = 162.15, p < 0.05$). A post-hoc analysis indicated that the phone inertial rotation was slower than touch rotation and the



wand. No statistical significance was found between the wand and touch rotate. Mean scores for rotation time are shown in Fig. 10a.

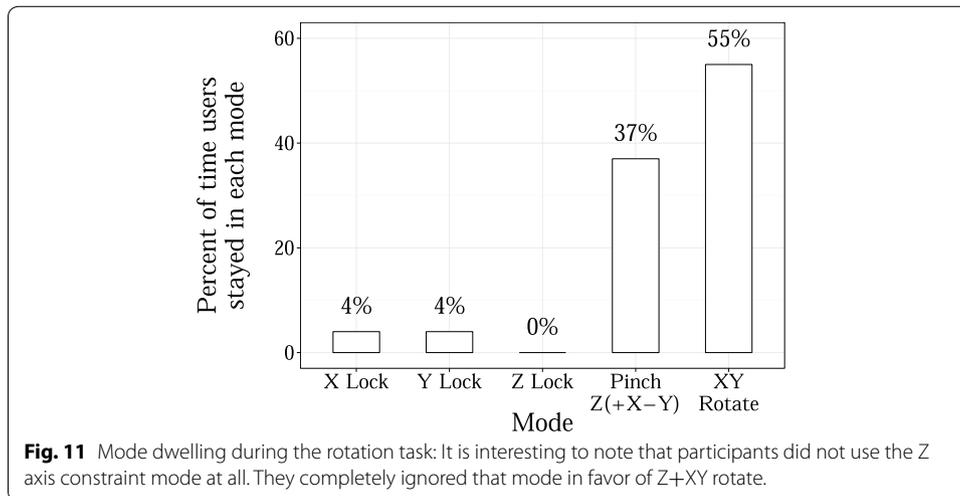
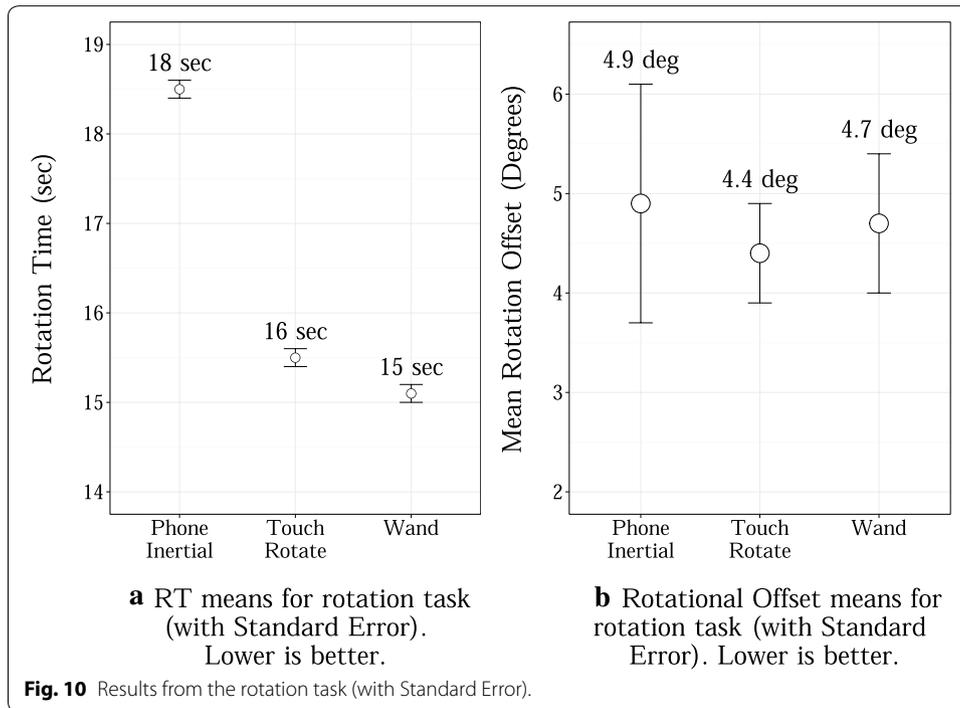
Upon ending each rotation trial, rotation accuracy was calculated as the angular difference (extracted from the quaternion) between the cursor orientation and the target orientation (Fig. 10b). The repeated measures ANOVA showed no statistical significance ($F_{2,22} = 1.13, p = 0.78$).

These results partially validate h2 for the rotation task yet we expected the time gap to be larger than two seconds. The form factor of the smartphone is indeed not ideal for rotations. We also hypothesized that touch rotate will perform better than the direct rotation techniques (wand and the phone inertial). This was not the case, however, and in an attempt to discover the reasons we analysed the time users spent on each rotation mode of touch rotate.

Mode dwelling during rotation task

During the rotation task, some of the pre-defined target orientations were simple 90° rotations about a single axis. In those trials, participants simply had to use the corresponding axis constraint rotation mode and control one DOF to accurately match the rotation. Had the participants used the axis lock modes they would have achieved almost perfect rotations. However, despite the availability of the constraint modes, participants did not use them very much (Fig. 11)

We believe one reason for the relative under-use of the constrained rotation modes was the somewhat arbitrary choice of corner-to-axis mapping. There was no easy mnemonic or meaningful mapping for the participants to remember to activate the constraint rotation modes. Consequently, participants instead spent most of their time in the single finger rotation mode (XY Rotate), followed by the pinch rotation mode for Z



axis control. The XY Rotate mode (single finger) controls two axes. That is two out of three axes to be manipulated. This should theoretically give it close to a 66% use. Interestingly, this was not the case. We speculate that this was because while the pinch rotate mode is mainly used for Z axis rotation, it can also control rotation about the X and Y axes by simultaneously moving both fingers parallel to each other.

We postulate that these mode dwelling results partially explain why, during the rotation task, touch rotate and wand performed similarly both in terms of completion time and accuracy. Because participants did not use the additional modes that would have potentially offered an advantage of accuracy and speed.

Touch areas

Visualization of participants' touch-points confirms findings from our pilot studies. Depending on the control mode, the participants' touches do not cover the entire touch-screen and some areas are unused (Fig. 4). For example, the bottom of the screen during the Z+XY rotation mode (Fig. 4b) was unused. These under-utilized parts of the touch-screen offer screen real-estate for adding further system control options, or UI-widgets.

Subjective results

Following completion of the experiment, participants were asked to state their preference between INSPECT and the wand technique. They were also asked to choose which technique felt more accurate and which technique had greater impact on the limbs (in terms of fatigue), three questions in total.

The qualitative results were notably skewed in favor of INSPECT. 17/20 participants preferred INSPECT overall. 18/20 thought that it felt more accurate and was less fatiguing. Participants commented that INSPECT was both fun to use and easy to understand. They further commented that the buttons on the side of the Samsung Galaxy SII were slightly hard to press which made mode changes slightly difficult. We agree with this assessment, and believe that INSPECT would benefit from having a few easy to press buttons on the forefinger side of the device. Finally, a number of participants commented favourably on the Z+XY rotation mode. They liked that the mode also allowed for XY rotation adjustments, and commented that they found it easy to use.

Discussion

Indirect touch, as opposed to direct touch interaction, suffers from the problem of selection. With Sticky Tools or tBox, the beginning of the touch gesture can simultaneously indicate object selection. With INSPECT, the manipulated object must be explicitly selected first with a selection step. This might complicate the use of INSPECT in applications requiring frequent selection among multiple different objects. Also, the magnetometer in smartphones is slightly susceptible to electromagnetic interference. If the user moves away from the display and sits while resting his arms on a metal structure (e.g., a desk) they may need to re-calibrate the orientation to avoid drift. The rotation mode would be unaffected in this case.

The movement task results come as a bit of a surprise. The wand technique leverages experience from daily use of the arms. We thus expected novice participants to perform better with it than with INSPECT. However, performance with INSPECT was actually better than the wand. This result validates our design decisions and indicates that a technique designed based on the aforementioned design principles shows tangible benefits. Namely:

- Bimanual in nature.
- Perceptual space of interaction task mirroring control space of input device.
- Small muscle groups—Interacting with palm/fingers rather than arm/forearm.
- Dominant hand interacting close to the Off-hand.

The rotation mode dwelling results as well as the low use of the flicking mode indicate that our novice participants were not familiarised enough with INSPECT to access the

additional modes and confined themselves to just using the basic ones. Even if performance in the evaluation tasks could have benefited from the use of these modes, the task did not demand their use and as such participants did not make the extra effort to utilise them. We think that with time and experience users will feel comfortable accessing, and benefit from these additional modes. Nevertheless, making these modes as well as additional ones more easily accessible could possibly have significant benefits for INSPECT. This is a point for improvement that must be carefully considered. Although the wide availability of smartphones motivates designs that can be used with smartphones, a different form factor, one that has access to a greater number of easily accessible buttons without affecting the size of the touch area could be beneficial.

In our experiments, users stood directly across from the display. In a collaborative situation that might be the case, but in a presentation scenario, the presenter would most likely be standing in a skewed position, to the side of the display. Unlike HOMER and other ray based techniques, INSPECT does not require the smartphone's position to be tracked relative to the display. Because of this we believe INSPECT will be more robust to the user moving relative to the display. That, however, remains to be demonstrated experimentally and is a potential goal for future research.

The recent proliferation of touch devices such as smartwatches and smartphones/tablets of different sizes begs the following questions: What effect does the screen size have on the performance of INSPECT? The screen real-estate of a smart-watch would potentially make it difficult to control modes that require more than one finger, like *pinch translate* and $Z(+X-Y)$ *rotate*. How could one overcome screen size limitations?

Conclusion and future work

We have presented INSPECT, a set of novel indirect touch techniques for 3D manipulation using a low cost input device such as a smartphone. The proposed technique to a certain extent meets the design goals set at the beginning of this paper, such as simplicity, accuracy and low-instrumentation/cost. INSPECT was overwhelmingly preferred over the wand technique by our experiment participants. The evaluation revealed that INSPECT performs 12% faster than a baseline wand technique for a 3D translation task while achieving 40% better accuracy and performs almost on-par at a 3D rotation task. The diverse rotation modes proved challenging for our novice participants and finding ways to enable simple access to a variety of modes remains a target for future work.

Endnote

^aA note on axes: When the device is held upright (Fig. 5), the axes on the device are identical to the axes on the display.

Additional file

Additional file 1: Video. Video demonstrating the techniques with narration.

Authors' contributions

NK designed the techniques, developed the software, ran the evaluations and drafted the manuscript. RJT helped draft the manuscript. KK helped draft the manuscript. HT supervised the project and helped draft the manuscript. All authors read and approved the final manuscript.

Author details

¹ Graduate School of Information Science and Technology, Osaka University, Yamadaoka 1-1, Suita, Osaka, Japan.

² McMaster University, 1280 Main Street West, Hamilton, ON L8S 4L8, Canada. ³ CyberMedia Center, Machikaneyama 1-32, Toyonaka, Osaka 560-0043, Japan.

Acknowledgements

The authors would like to thank Prof. Doug Bowman for his valuable advice and the experiment participants for their time. We would also like to thank Prof. Tomohiro Mashita for his support.

Compliance with ethical guidelines**Competing interests**

The authors declare that they have no competing interests.

Received: 2 February 2015 Accepted: 21 June 2015

Published online: 30 July 2015

References

- Piekarski W, Thomas BH (2001) Tinmith-metro: New outdoor techniques for creating city models with an augmented reality wearable computer. In: IEEE Proceedings fifth international symposium on wearable computers, pp 31–38
- Whyte J, Bouchlaghem N, Thorpe A, McCaffer R (2000) From cad to virtual reality: modelling approaches, data exchange and interactive 3D building design tools. *Autom Constr* 10(1):43–55
- Stoakley R, Conway MJ, Pausch R (1995) Virtual reality on a wim: interactive worlds in miniature. In: CHI. ACM Press/Addison-Wesley Publishing Co, pp 265–272
- Hilliges O, Kim D, Izadi S, Weiss M, Wilson A (2012) Holodesk: direct 3D interactions with a situated see-through display. In: Proceedings of the SIGCHI conference on human factors in computing systems, ACM, pp 2421–2430
- Laha B, Bowman DA (2013) Volume cracker: a bimanual 3D interaction technique for analysis of raw volumetric data. In: Proceedings of the 1st symposium on spatial user interaction, ACM, pp 61–68
- Tse E, Xin M, Antonyuk V, Lai H, Hudson C (2013) Making 3D content accessible for teachers. In: ITS. ACM, New York, pp 125–134. doi:10.1145/2512349.2512803
- Satava RM (1993) Virtual reality surgical simulator. *Surg Endosc* 7(3):203–205
- Hancock M, ten Cate T, Carpendale S, Isenberg T (2010) Supporting sandtray therapy on an interactive tabletop. In: CHI. CHI '10. ACM, New York, pp 2133–2142. doi:10.1145/1753326.1753651
- Song P, Goh WB, Hutama W, Fu C-W, Liu X (2012) A handle bar metaphor for virtual object manipulation with mid-air interaction. In: CHI, pp 1297–1306
- Katzakis N, Kiyokawa K, Takemura H (2012) Planecasting: 3D cursor control with a smartphone. In: Proceedings of 3DCHI: touching and designing 3D user interface, ACM, pp 13–21
- Bier EA (1987) Skitters and jacks: interactive 3D positioning tools. In: Proceedings of the 1986 workshop on interactive 3D graphics, ACM, pp 183–196
- Blender Online Community (2002) Blender—a 3D Modelling and Rendering Package. Blender Foundation, Amsterdam, The Netherlands. Blender Foundation. <http://www.blender.org/>
- Shoemaker K (1992) Arcball: a user interface for specifying three-dimensional orientation using a mouse. *Graph Interface* 92:151–156
- Hirzinger G (1982) Robot-teaching via force-torque-sensors. In: Proceedings of the sixth european meeting on cybernetics and systems research
- Fröhlich B, Hochstrate J, Skuk V, Huckauf A (2006) The globefish and the globemouse: two new six degree of freedom input devices for graphics applications. In: CHI. CHI '06. ACM, New York, pp 191–199. doi:10.1145/1124772.1124802
- 3DConnexion. <http://www.3dconnexion.com/products/spacenavigator.html>. Accessed 14 July 2015
- Hinckley K, Wigdor D (2002) Input technologies and techniques. The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications, 2nd edn. CRC Press, pp 151–168
- Wilkes C, Bowman DA (2008) Advantages of velocity-based scaling for distant 3D manipulation. In: VRST, ACM, pp 23–29
- Pierce JS, Stearns BC, Pausch R (1999) Voodoo dolls: seamless interaction at multiple scales in virtual environments. In: Proceedings of the 1999 symposium on interactive 3D graphics, ACM, pp 141–145
- Bowman DA, Hodges LF (1997) An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In: I3D, ACM, pp 35–38
- Poupyrev I, Ichikawa T, Weghorst S, Billinghurst M (1998) Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques. In: Computer graphics forum, vol 17. Wiley Online Library, pp 41–52
- Poupyrev I, Billinghurst M, Weghorst S, Ichikawa T (1996) The go-go interaction technique: non-linear mapping for direct manipulation in vr. In: UIST, ACM, pp 79–80
- Polhemus Fastrack. <http://polhemus.com/motion-tracking/all-trackers/fastrak>. Accessed 14 July 2015
- Optitrack. <http://www.optitrack.com/products/flex-3/>. Accessed 14 July 2015
- Hincapié-Ramos JD, Guo X, Moghadasian P, Irani P (2014) Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In: Proceedings of the SIGCHI conference on human factors in computing systems. CHI '14, ACM, New York, pp 1063–1072. doi:10.1145/2556288.2557130
- Katzakis N, Seki K, Kiyokawa K, Takemura H (2013) Mesh-grab and arcball-3d: Ray-based 6-dof object manipulation. In: Proceedings of the 11th Asia Pacific conference on computer-human interaction. APCHI '13, ACM, New York, pp 129–136. doi:10.1145/2525194.2525198

27. Hachet M, Dècle F, Knödel S, Guitton P (2008) Navidget for easy 3D camera positioning from 2D inputs. In: 3DUI. <http://iparla.inria.fr/publications/2008/HDKG08>
28. Frohlich B, Plate J, Wind J, Wesche G, Gobel M (2000) Cubic-mouse-based interaction in virtual environments. *IEEE Comput Graph Appl* 20(4):12–15
29. Reisman JL, Davidson PL, Han JY (2009) A screen-space formulation for 2D and 3D direct manipulation. In: UIST, ACM, New York, pp 69–78. doi:10.1145/1622176.1622190
30. Hancock M, Carpendale S, Cockburn A (2007) Shallow-depth 3D interaction: Design and evaluation of one-, two- and three-touch techniques. In: CHI, ACM, pp 1147–1156
31. Hancock M, Ten Cate T, Carpendale S (2009) Sticky tools: full 6dof force-based interaction for multi-touch tables. In: ITS, ACM, pp 133–140
32. Cohé A, Dècle F, Hachet M (2011) tbox: a 3D transformation widget designed for touch-screens. In: CHI, ACM, pp 3005–3008
33. Wilson AD, Izadi S, Hilliges O, Garcia-Mendoza A, Kirk D (2008) Bringing physics to the surface. In: UIST, ACM, pp 67–76
34. Voelker S, Wacharamanatham C, Borchers J (2013) An evaluation of state switching methods for indirect touch systems. In: Proceedings of the SIGCHI conference on human factors in computing systems. CHI '13, ACM, New York, pp 745–754. doi:10.1145/2470654.2470759
35. Ohnishi T, Katzakis N, Kiyokawa K, Takemura H (2012) Virtual interaction surface: Decoupling of interaction and view dimensions for flexible indirect 3D interaction. In: 2012 IEEE symposium On 3D user interfaces (3DUI), pp 113–116. doi:10.1109/3DUI.2012.6184194
36. Wigdor D, Benko H, Pella J, Lombardo J, Williams S (2011) Rock & rails: extending multi-touch interactions with shape gestures to enable precise spatial manipulations. In: Proceedings of the SIGCHI conference on human factors in computing systems, ACM, pp 1581–1590
37. Jacob RJK, Sibert LE, McFarlane DC, Mullen MP Jr (1994) Integrality and separability of input devices. *ACM Trans Comput Hum Interact* 1(1):3–26
38. Zhai S, Milgram P, Buxton W (1996) The influence of muscle groups on performance of multiple degree-of-freedom input. In: CHI, ACM, pp 308–315
39. Guiard Y (1987) Asymmetric division of labor in human skilled bimanual action: the kinematic chain as a model. *J Motor Behav* 19:486–517
40. Balakrishnan R, Hinckley K (2000) Symmetric bimanual interaction. In: CHI. CHI '00, ACM, New York, pp 33–40. doi:10.1145/332040.332404
41. Apple: Magic Trackpad. <https://www.apple.com/magictrackpad/>. Accessed 14 July 2015
42. Chen M, Mountford SJ, Sellen A (1988) A study in interactive 3-d rotation using 2-d control devices. *SIGGRAPH Comput Graph* 22(4):121–129. doi:10.1145/378456.378497
43. Rousset É, Bérard F, Ortega M (2014) Two-finger 3D rotations for novice users: surjective and integral interactions. In: Proceedings of the 2014 international working conference on advanced visual interfaces, ACM, pp 217–224
44. ViSSee: Flying Mouse. <https://itunes.apple.com/us/app/flying-mouse/id590741906>. Accessed: 14 July 2015
45. Hinckley K, Tullio J, Pausch R, Proffitt D, Kassell N (1997) Usability analysis of 3D rotation techniques. In: Proceedings of the 10th annual ACM symposium on user interface software and technology. UIST '97. ACM, New York, pp 1–10. doi:10.1145/263407.263408

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
