INSPECT: Extending Plane-Casting for 6-DOF Control

Nicholas Katzakis* Osaka University
Robert J. Teather† McMaster University
Kiyoshi Kiyokawa‡ Osaka University Cybermedia Center
Haruo Takemura§ Osaka University Cybermedia Center

ABSTRACT

INSPECT is a novel set of interaction techniques for 3D object manipulation using a rotation-only tracked touch panel. Motivated by the applicability of the techniques 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 the techniques to a comparative user study in which INSPECT was preferred by the users overall.

Index Terms: H.5.2 [Information Inferfaces and Presentation]: User Interfaces—Input devices and strategies; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques

1 INTRODUCTION

Virtual object manipulation is required in a wide variety of application domains. Our work focuses on 6-DOF object manipulation at a distance with a large display for presentations and education and extends the Plane-Casting metaphor [1].

The design goals of an interface for presentations 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 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.

2 INSPECT

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. Therefore, we wanted to allow users to use the technique while looking directly at the large display, without having to look at the device (to manipulate widgets etc.).

Extensions to translation mode: To improve Plane-Casting [1], on which INSPECT is based, we added a "flick" gesture. 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 not 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. 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. Another extension to translation was that pinching the fingers *away* translates the object parallel to, and in the direction of the control plane normal. Conversely, pinching the fingers *together* (or "un-pinching") translates the object in the opposite direction.

*e-mail: katzakis@ist.osaka-u.ac.jp
†e-mail:teather@mcmaster.ca
‡e-mail:kiyokawa@ime.cmc.osaka-u.ac.jp
§e-mail:takemura@ime.cmc.osaka-u.ac.jp

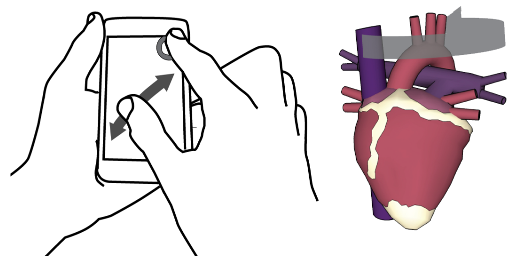


Figure 1: When in rotation mode, placing the first finger on the corner constrains rotation to a single axis (Y-axis in this case).

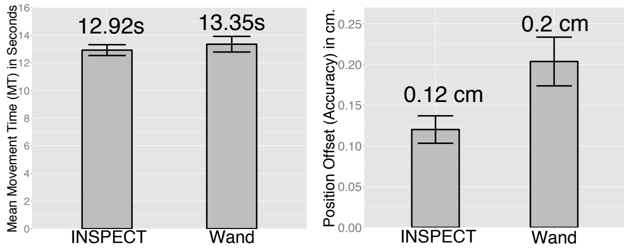
Extensions for rotation: In addition to the translation extensions discussed above, we added a new mode to enable rotation. The smartphone's volume-up button switches the system to rotation mode while being held pressed.

Horizontal finger motion (on the touch-screen X axis) rotates the object about the world Y axis. 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 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.

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. 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. [3]. Arcball+ uses the midpoint to rotate like the classical ARCBALL 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.

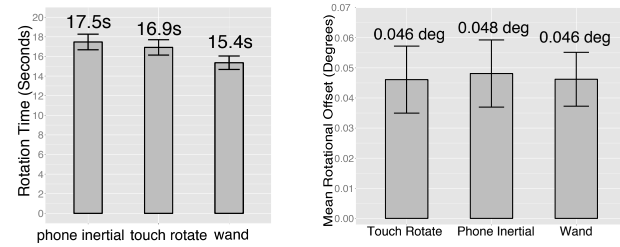
Our system also allows for single-axis constraint modes. Single-axis constrained rotations are activated by touching the display cor-



(a) Movement Time (lower is better).

(b) Distance between cursor and target (lower is better).

Figure 2: Results from the movement task (with Standard Error).



(a) RT means for rotation task (with Standard Error). Lower is better.

(b) Rotational Offset means for rotation task with standard error.

Figure 3: Results from the rotation task (with Standard Error).

ners. Analysis of our pilot study touch data revealed that users rarely reach the touchscreen corners while moving objects with Plane-Casting. 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 (Figure 1). 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. (Figure 1). **(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 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 (Figure 1). The thumb's vertical motion on the touchscreen is ignored and only the horizontal component rotates the object about the constrained Y axis. Finally, our system provided object selection via two methods. A virtual hand-like 3D cursor, and a 2D cursor with relative control.

3 Evaluation

We elected to compare against a 'gold standard' technique using a wand tracked by a 6-DOF Polhemus sensor. Twelve participants took part in the study.

The movement task required matching a 3D cursor position to to 12 pre-defined rotations [2] shuffled for each participant, twice. INSPECT and the Wand technique were compared using this task. In the rotation task, the cursor and the target both appeared centered at the origin and participants had to match the orientation. The rotation task included two techniques using the smartphone (touch rotation and inertial sensor rotation) and direct rotation using the wand. 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.

The paired t-test revealed a significant effect on accuracy for movement technique ($\mathbf{t}_{287} = -3.37, p = 0.0008$) (Figure 2b). Participants were more accurate in matching the target position using INSPECT than with the wand - the error distance with the wand was about 66% higher than that of INSPECT. The average distance mismatch for the wand was 0.2 cm (Figure 2). The data from the rotation task was subjected to a repeated measures ANOVA test. Technique did not have a significant effect on rotation time (RT) ($F_{2,22} = 0.58, p = 0.56$) or on rotation accuracy ($F_{2,22} = 0.13, p = 0.98$) (Figure 3) .

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), 3 questions in total. The qualitative results were notably skewed in favor of INSPECT. 9/12 participants preferred INSPECT overall. 10/12 thought that it felt more accurate and was less fatiguing. Participants commented that INSPECT was both fun to use and easy to understand.

4 Discussion

In indirect touch techniques (such as INSPECT) the user does not need to be terribly precise when starting the touch gesture. Direct touch techniques like tBox and Sticky Tools need aiming for every gesture. On the other hand, indirect touch, suffers from selection problems. With Sticky Tools, the beginning of the touch gesture can simultaneously indicate object selection. With INSPECT, the manipulated object must be explicitly selected first with a cursor or a side button press. This might complicate the UI of applications supporting selection of multiple different objects.

Some state-of-the-art touch techniques designed for tabletops can be adapted for use on vertical or handheld touch displays. For example, techniques like tBox and Sticky Tools could complement INSPECT, depending on how close the user is to the display. Another advantage of INSPECT compared to widget-based approaches is the relative simplicity in implementation (tBox's UI widgets are more than trivial to program) while preventing the overhead and clutter induced by graphic widgets.

INSPECT could also be used to guide a quadrocopter, a robot, or a new genre of games that feature integral 3D translations. Finally, a smartphone-form controller seems ideal for presentations, but the techniques presented here also have applications in desktop VR, CAVE environments, head-mounted displays, and augmented reality. The same manipulation techniques could also be applied to a virtual camera for navigation. It should also be simple to extend INSPECT to include symbolic input via voice or another input method editor for a complete interactive controller.

References

[1] N. Katzakis, K. Kiyokawa, and H. Takemura. Plane-Casting: 3d Cursor Control with a Smartphone. In *Proceedings of 3DCHI: Touching and Designing 3D User Interface, Workshop at ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 13–21. ACM, 2012.

[2] N. Katzakis, K. Seki, K. Kiyokawa, and H. Takemura. Mesh-grab and Arcball-3d: Ray-based 6-DOF Object Manipulation. In *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction - APCHI*, pages 129–136. ACM, 2013

[3] É. Rousset, F. Bérard, and M. Ortega. Two-Finger 3d Rotations for Novice Users: Surjective and Integral Interactions. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, pages 217–224. ACM, 2014.