ROTATION TECHNIQUES FOR 3D OBJECT INTERACTION ON MOBILE DEVICES

by

VLAD BUDA

Student number: 3427404

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Information and Computing Sciences
Game and Media Technology

UTRECHT UNIVERSITY

August 2012

Supervisor:

Dr. Wolfgang Hürst

Abstract

Recent technological advances of high-end mobile devices, such as smartphones and tablets, have paved the way for an increasing number of games, virtual reality environments and visualization systems, specifically tailored for the mobile device platform. However, 3D interaction techniques that complement the above 3D mobile applications, by providing a smooth and natural control mechanism have been largely neglected. This work extends traditional desktop rotation techniques that are based on the virtual trackball metaphor by accommodating them to the restrictions and functionalities of mobile devices. Instead of a typical setting involving a mouse and output monitor, users are able to interact directly with a 3D object by making simple finger gestures such as dragging or sliding on the display screen of the mobile device.

The main goal of this study is to provide an efficient and intuitive rotation technique for 3D object interaction on mobile devices capable of conveying important structural and semantic information about a 3D object. Different device-interaction styles, based on the combination of device orientation modes (landscape/portrait) and number of hands used (one/two-handed interaction) play an important role in user involvement. The investigated rotation techniques along with the subsequent device-interaction styles are compared, in an empirical study, in terms of performance and accuracy with respect to a multitude of rotation tasks. The results of this study indicate that 3D object interaction on mobile devices can greatly benefit from the use of appropriate rotation techniques. Rotation techniques that generate predictable rotation patterns perform better and are rated higher by users than rotation techniques that generate transitive rotations.

Table of Contents

Acknowledgements

I would like to thank my supervisor Wolfgang Hürst for his helpful advice and guidance throughout the project. I would also like to thank Martin de Jong and Jeroen van Knotsenburg for their suggestions and feedback during the implementation stages as well as Răzvan Turtoi for his proofreading work and everyone who participated in the evaluation process.

Chapter 1 - Introduction

With the advances in hardware, 3D mobile graphics has become one of the most predominant areas of research for the mobile platform. The idea of bundling powerful hardware features [Ancillotti 2011], such as dual and quad core processors, significant amount of RAM memory, large storage space and efficient graphics cards, into relatively small mobile devices, coupled with sensors such as touchscreen, accelerometer, gyroscope and compass that allow for different control input methods, has made the smartphone and tablet high-end mobile devices, well suited for 3D mobile graphics.

However, as newer functionalities and features have paved the way for 3D mobile applications such as games, virtual reality environments and visualization systems, 3D interaction techniques that provide a proper control mechanism for user involvement have largely been neglected. Based on the type of 3D system, 3D interaction can be divided into two main categories, 3D environment interaction which deals with the overall interaction in a large virtual environment and 3D object interaction which focuses on interacting with 3D objects as single entities. In the case of 3D object interaction, the view-camera is centered on the 3D object, with a fixed pivot point, from which all the main interaction tasks such as translation, rotation and scaling take effect. Translation and scaling are important factors in 3D object interaction, but they are not as vital for increasing the understanding of a 3D object as rotation is. As a consequence, this project focuses on rotation as the main 3D object interaction task and does not take into account translation and scaling.

The current majority of 3D mobile applications have impressive visual graphics and special effects, but the inefficient interaction techniques that they employ lead to poor and unnatural controls. An immediate solution would be to look at the 3D interaction techniques used on desktop and table-top display platforms. However, the transition of 3D interaction techniques from established desktop systems or table-top displays to mobile devices is problematic. This is mostly due to the inherent mobile constraints [Maniar, Bennett, Hand & Allan, 2008], such as insufficient screen real-estate, thumb occlusion, small physical keys, and a lack of knowledge about interaction techniques specific for mobile devices.

While increasing the framerate and render quality of 3D mobile applications depends largely on a system's hardware and software capabilities, efficient 3D interaction techniques rely on a system's input and output features along with a technical understanding of the user himself [Card & Moran,1986]. Users need to create a mental model of how the controls and input methods of the mobile device work in order to establish a fluent connection with the virtual applications presented on the mobile device. Therefore, interaction on mobile devices has two main functions, namely device-interaction and application-interaction. Device-interaction is based on a combination of one or two-handed interaction with the orientation modes of the mobile device and the use of finger or thumb. This leads to a multitude of device-interaction styles such as two-handed/landscape with finger or thumb, two-handed/portrait with finger or thumb, one-handed/landscape with finger or thumb and one-handed/portrait with finger or thumb. Application-interaction deals with the technical specifications of object rotation such as rotation techniques and the subsequent influencing parameters. The device-interaction and application-interaction functions are closely correlated and special care must be taken to insure that they work in a cohesive manner to achieve the desired objective. As a result, this project is focused on investigating both the rotation techniques as an application-interaction function as well as the different device-interaction styles that can be employed.

The main goal of this experimental study is to provide an efficient and intuitive rotation technique for 3D object interaction on mobile devices by means of touchscreen gestures such as dragging a finger or thumb across the display screen of the device. Such a technique should have an intuitive and predictable behavior, an easy learning curve along with the ability to convey important structural and semantic information about a 3D object by taking advantage of touchscreen gestures and the multitude of device interaction styles.

Providing an appropriate rotation technique for 3D object interaction as well as a comprehensive view of device interaction styles can bring important benefits to several areas of 3D mobile applications. For cultural heritage, museums may complement exhibitions by setting up a special

3D art catalogue where people can visualize and interact with 3D cultural artifacts. Archeologists can create an online database with 3D objects for others to inspect [Ericson & Olwal, 2011]. Another area of interest is advertising and retail, where shops can present their merchandize in a virtual catalogue for potential buyers to explore [MCW, 2012]. 3D mobile games [Gordongdgd, 2012] can benefit from these research findings as developers can integrate more appropriate settings for rotating 3D objects on mobile devices. Exploring CAD models on a mobile device [Autodesk Mobile Apps, 2012] could give a stronger sense of shape and volume compared to traditional desktop-based CAD tools and may save several iterations of rapid prototyping. The exploration of 3D phenomena (meteorological, medical, biological, etc.) could allow medical doctors, radiologists, biologists, students and researchers, to better understand complex anatomy in a mobile setting [DNA Learning Center, 2012].

The remainder of this thesis is structured as follows: In Chapter 2, an overview of background and related work is presented. Chapter 3 contains a thorough description of the design space which includes the investigated rotation techniques and device interaction styles. Details regarding the experimental framework used to evaluate the rotation techniques and device interaction styles are presented in Chapter 4. Results of the experimental studies showing the effectiveness and user preference of the proposed rotation techniques and device interaction styles are provided in Chapter 5 and discussed in Chapter 6. In Chapter 7, general conclusions and suggestions for future work are presented.

Chapter 2 - Background and related work

This chapter presents the state of the art in rotation techniques for 3D object interaction on desktop, table-top and mobile devices along with a detailed look into the different device-interaction styles that complement the rotation techniques.

2.1 Rotation techniques

According to [Henriksen, Sporring, & Hornbæk, 2004], there are four main categories of rotation techniques specific for the desktop platform, namely view-based techniques, controller-based techniques, multiple-degrees-of-freedom techniques and virtual trackball techniques. The issue of whether these rotation techniques can be successfully adapted for use on mobile devices is still on-going and examples of such implementations have only recently surfaced.

2.1.1 View-based techniques

Commercial applications and research prototypes rely mostly on view-based techniques, where several different views of the 3D object, corresponding to the xy, xz and yz projections, are presented to the user who, by using a slider, can rotate the object on one or two dimensions. View-based techniques suffer from drawbacks, most notably the need for large screen size, limited rotation freedom, which is usually constrained to one dimension at a time, and the inability of users to construct reliable mental models about the relative contributions and effects of all the different views of the object [Gallo, Minutolo, & de Pietro, 2010]. These limitations become increasingly difficult to overcome for mobile devices were screen-real estate is problematic and employing a four-panel viewport would significantly diminish the 3D object display size. Consequently, view-based techniques are best suited for desktop and possibly direct-touch tabletop platforms.

2.1.2 Controller-based techniques

Controller-based techniques are a popular category due to their explicit and precise controls. For these techniques, each direction of rotation is mapped to a controller, usually represented by a graphical user interface containing keys, buttons and sliders. Their dependency on a particular user interface, which has to be designed in such a way that allows for an easy learning curve, along with the increased time it takes users to switch between controllers, constitute disadvantages that can cause usability issues. However, the small screen size of mobile devices is beneficial for reducing the time needed to switch between controllers. Additionally, if an appropriate graphical user interface is employed, these techniques can potentially be effective for 3D object interaction on mobile devices [Buda, 2012].

2.1.3 Multiple-degrees-of-freedom techniques

A special category of rotation techniques is represented by multiple-degrees-of-freedom techniques where, in order to perform full object manipulation, input devices with additional degrees of freedom are used. According to certain studies [Hinckley, Tullio, Pausch, Proffitt, & Kassell, 1997], [Ware & Rose, 1999], these techniques might be more appropriate for rotating 3D objects than all other rotation techniques. However, it appears that users have great difficulties in dealing with the visual appearance and movement associated with rotating 3D objects by means of a multiple-degree-of-freedom device. Another disadvantage is the high cost of these input devices coupled with the lack of a standardized model. Notable examples of such implementations include the 3D physical trackball developed by [Kim, Seong, Hyun, Lee, & Choi, 2001], the touch-haptic touchball of [Choi & Kim, 2009] and the two 6DOF devices, GlobeFish and GlobeMouse by [Froehlich, Hochstrate, Skuk, & Huckauf, 2006].

In recent years, as inertial sensors such as accelerometers, gyroscopes and compasses, are being integrated into mobile devices, it has become convenient and beneficial to utilize mobile devices as multiple-degree-of-freedom input devices. [Katzakis & Hori, 2009] use a smartphone equipped with an accelerometer and digital compass as a 3DOF controller in order to rotate 3D objects. Their results indicate that for 3D object rotation tasks, the smartphone is faster than traditional 2D input devices such as a mouse and pen. Whereas [Katzakis & Hori, 2009] use a desktop monitor for visual feedback, which encumbers the whole system and makes it less mobile, [Sasakura, Kotaki, & Inada, 2011] go one step further and use the smartphone's display screen as output. While these techniques are becoming increasingly popular and easier to implement, especially given the advances in sensor technology, inherent drawbacks such as drift and noise errors [Kirkham, 2010] prevent them from becoming the standard techniques of 3D object interaction.

A particularly interesting and emerging topic in the field of mobile device interaction is Around-Device Interaction [Kratz & Rohs, 2009]. In a recent study, [Kratz, Rohs, Guse, Müller, Bailly, & Nischt, 2012] present a novel style of mobile interaction for rotating 3D objects where users hold the mobile device with one hand and with the other hand perform mid-air gestures in the proximity of the device to control rotation. This method alleviates some of the problems typical of other rotation techniques such as thumb occlusion by freeing up screen space. However, implementing an around-device interaction method is difficult as it requires additional hardware features which are not available by default in mobile devices. Also, users might experience fatigue due to a continuous two-handed interaction in which one hand has to remain in mid-air throughout the interaction process.

2.1.4 Two-handed techniques

More complex techniques for rotating 3D objects include two-handed techniques, where users perform rotations by using two hands to control two independent cursors [Zeleznik, Forsberg, & Strauss, 1997]. The idea is to map certain degrees of freedom to one cursor and other degrees of freedom to the second cursor so that users can fluently perform interaction tasks. [Debarba, Franz, Reus, Maciel, & Nedel, 2011] designed a 3D rotation system where one hand of the user controls a 3D cursor with six degrees of freedom and the other hand controls the orientation of a working area with a tracking-enabled mobile device with three degrees of freedom.

2.1.5 Virtual trackball techniques

Virtual trackball techniques allow for direct manipulation of 3D objects by providing rotation along several dimensions simultaneously and by integrating the controller and the object controlled. Rotation is accomplished by projecting the 2D mouse movements onto a 3D sphere also called virtual trackball to get the corresponding 3D points on the virtual trackball's surface. Thus, the 2D motion of the mouse is mapped to a 3D rotation from one projected point on the virtual trackball's surface to a second one.

Several factors indicate that virtual trackball techniques are the most appropriate rotation techniques for 3D object interaction on mobile devices. First, virtual trackball techniques have established themselves as an industry standard for 3D object interaction and are being used in specialized commercial applications such as Autodesk Maya, 3D Studio Max, Mudbox, etc., [Autodesk Software, 2012]. Second, unlike some of the other rotation techniques presented above, virtual trackball techniques do not require any unique graphical user interface, specialized viewport, physical or virtual controls, specialized input devices or inertial sensors, all of which can potentially hinder a smooth 3D object interaction process. Finally, the underlying control mechanism of virtual trackball techniques shares screen real-estate with the 3D object that it controls and therefore the entire display screen of the mobile device can be used as input for touchscreen gestures.

2.1.5.1 Virtual Sphere

[Chen, Mountford, & Sellen, 1988] developed the Virtual Sphere as the first virtual trackball technique to simulate the mechanics of a physical 3D sphere that can freely rotate about any arbitrary axis in 3D space. The 2D viewport location of the moving mouse is projected onto the sphere to get corresponding 3D points on the sphere's surface. Vertical and horizontal mouse movement at the centre of the 3D object is equivalent to "rolling" the imaginary sphere at its apex and produces rotation about the x and y-axis respectively. Arbitrary mouse movement along, or completely outside, the 3D object is equivalent to rolling the sphere at the edge and produces rotation about z. While this approach provides users with an easy way of rotating 3D objects on all three axes and it is the basis of all other virtual trackball techniques, it presents a significant problem, namely that the resulting rotation axis is not perpendicular to the mouse displacement vector and therefore the rotation of a 3D object is at times discontinuous.

2.1.5.2 Bell's Virtual Trackball

[Bell, 1988] improved the Virtual Sphere and instead of projecting the 2D mouse location onto a sphere, Bell projects it to a combination of a sphere and a hyperbola in 3D space. If the 2D point is close to the center of the 3D object, the surface is a sphere, otherwise it is a hyperbola. Thus, rotating a 3D object using this approach appears very smooth because the orientation of the axis of rotation is continuous as a function on the screen coordinates. However, a big disadvantage of Bell's VT is the fact that the smoothing treatment sacrifices the predictability of a simulated physical ball since the ball is no longer rotating along the great circle arc [Henriksen, Sporring, & Hornbæk, 2004]. Horizontal mouse movement at the center of the 3D object generates rotations about the up-vector of the viewport (y axis) but the same movement at the bottom or top border of the 3D object generates rotations about the look-vector of the viewport (z axis). The only thing that remains predictable is the clockwise and counterclockwise movement. For example, moving the mouse at the top border of the 3D object to the right rotates the 3D object clockwise and moving the mouse at the bottom border of the 3D object to the right rotates the 3D object counterclockwise.

2.1.5.3 Shoemake's Arcball

[Shoemake, 1992] implemented a special version of the virtual trackball, the so-called Arcball, which allows users to perform both free and constrained rotations using any direction as an axis and provides kinesthetic agreement between mouse motion and 3D object rotation along with transitive rotations. The Arcball projects points onto a sphere and uses a great arc to determine the rotation path. Instead of the formula used in the Virtual Sphere for rotating around an axis, Shoemake uses quaternions which also represent a rotation but twice as fast as in the Virtual Sphere [Kettner, 1995]. A detailed implementation description was written by [Shoemake, 1994] and while it is an "elegant" implementation, with many benefits to overall 3D object rotation, the Arcball has some prominent disadvantages that make it hard to use. One of the main drawbacks is the inability to customize the control-to-display ratio which means that the smaller the viewport the faster the 3D object is rotated. Second, the 3D object is rotated twice about the look-vector (z axis) if the mouse is moved once around the projection sphere, which may lead to confusion. Finally, the rotation seems to snap to the rim with a loss of user control and accuracy.

2.1.5.4 Continuous XY with added Z Controller

Alongside their Virtual Sphere, [Chen, Mountford, & Sellen, 1988] implemented another virtual trackball technique called Continuous XY with added Z Controller. The difference between the Virtual Sphere and the Continuous XY with additional Z Controller is that the Virtual Sphere allows continuous rotation about all three axes inside the circle that encompasses the 3D object while the latter only allows continuous control of two axes inside the circle and in order to rotate about the look-vector (z-axis), users must go outside the circle. The Continuous XY with added Z controller operates in two modes. If the mouse button is pressed while the mouse cursor is inside the circle, horizontal and vertical movement of the mouse will rotate the object horizontally and vertically on the screen. Diagonal movement will rotate the object the proportional amount about the x and y-axis (i.e. the axis of rotation is on the x-y plane and is perpendicular to the direction of mouse movement). If the mouse button is pressed while the mouse cursor is outside the circle, the user can rotate the whole object clockwise by going around the outside of the circle [Chen, 1991].

2.1.5.5 Hanson's Rolling Ball

[Hanson, 1992] presented the Rolling Ball, a virtual trackball technique that exploits a continous two dimensional motion (modeled after that of a ball rolling without slipping on a table) to reach any arbitrary three dimensional orientation. In addition to horizontal and vertical rotations about y and x respectively, this technique allows users to perform clockwise and counterclockwise rotations about z with respect to the screen perpendicular. This surprising aspect is due to a fundamental property of the group theory of spatial rotation [Ma, 2007] which states that moving a rolling ball controller in small clockwise circles must produce small counterclockwise rotations of the ball and vice versa.

2.1.5.6 Two-axis Trackball

Another virtual trackball technique is the Two-axis Trackball originally developed by [Thornton, 1979] , further studied and evaluated by [Evans, Tanner, & Wein, 1981] and integrated as part of the Continous XY with added Z Controller technique by [Chen, Mountford, & Sellen, 1988]. The Two-axis Trackball only allows rotation about the x and y axes which makes it very predictable. However, the Two-axis Trackball had no control over the look-vector and therefore cannot perform clockwise and counterclockwise rotations.

2.1.5.7 Fixed-right/up Trackballs

The Fixed-right/up Trackballs are modified versions of the original Two-axis Trackball which allow for transitive rotations due to the use of a combination of local and world vectors. The Fixed-right Trackball has vertical rotations in the plane set about the world’s right vector which is “fixed” regardless of how a 3D object is oriented, hence the name of the rotation technique, while the Fixed-up Trackball has horizontal rotations set about the world’s up vector and vertical rotations about the local right-vector. They are mostly utilized in highly specialized 3D interaction applications such as Autodesk 3D Studio Max and Autodesk Maya [Autodesk, 2012] and the majority of 3D mobile applications [Gordongdgd, 2012], [Export2Reality, 2012], because of their ability to perform rotations about the look-vector (z axis).

2.1.6 Comparison of VT techniques in related literature

Several comparison experiments have been undertaken to evaluate the usability of virtual trackball techniques for 3D object interaction on the desktop platform using a 2D mouse as input. [Chen, Mountford, & Sellen, 1988] compared several rotation techniques including the Virtual Sphere and the Continuous XY with additional Z Controller. Their findings indicated no practical differences in accuracy between the two rotation techniques. However, the Virtual Sphere was more suited for complex rotations and performed slightly better in terms of speed and user preference. [Jacob & Oliver, 1995] also compared the Virtual Sphere and the Continous XY, and in addition to the rotation task used by Chen, they also introduced a more complex inspection task. With respect to task completion time, the Virtual Sphere was faster than the Continous XY with additional Z Controller.

The results from a formal user study done by [Hinckley, Tullio, Pausch, Proffitt, & Kassell, 1997] showed that there are no significant differences in terms of task completion time and accuracy between Shoemake's Arcball and the Virtual Sphere. [Bade, Ritter, & Preim, 2005] evaluated four virtual trackball techniques, namely Bell's VT, Shoemake's Arcball, Two-axis Trackball and Fixed-up Trackball. They discovered that the Two-axis Trackball as well as Shoemake's Arcball performed best in terms of task completion time, but that their accuracy is lower than Bell's VT. Moreover, users considered the Two-axis Trackball to be the most convenient rotation technique. Recently, [Zhao, Shuralyov, & Stuerzlinger, 2011] investigated Bell's VT, Shoemake's VT along with the Two-axis Trackball and found no performance or accuracy differences among these techniques. The studies reviewed here omit information regarding implementation details for the virtual trackball techniques which makes comparison across studies difficult. Furthermore, the lack of a thoroughly described implementation process is likely to have a large impact on usability as the same rotation technique could have been implemented differently in each study case. This might explain the inconsistencies between results obtained in all the studies.

In summary, the Two-axis Trackball seems to be the best virtual trackball rotation technique for 3D object interaction on the desktop platform using a 2D mouse as input. Unfortunately, there is little evidence to indicate that these results are reproducible on the mobile platform. Hence, assuming the Two-axis Trackball to be the most appropriate rotation technique for 3D object interaction on mobile devices can be a potential fallacy. Several studies have been performed to investigate the virtual trackball metaphor on mobile devices. [Decle & Hachet, 2009] focused on the usability of the virtual trackball on touch-based mobile devices. They compared the Two-axis Trackball with a modified version called Planned Trackball which makes use of incremental rotation and found that the Two-Axis Trackball outperforms the Planned Trackball. The results of a recent study by [Kratz & Rohs, 2010] indicate the superiority of virtual trackball techniques compared to tilt-based approaches for 3D rotation tasks on mobile devices.

2.2 Mobile device interaction and orientation

For touchscreen based mobile devices such as smartphones, different styles of interaction can be employed by users and therefore they need to be properly investigated in order to understand how they are influencing overall performance and preference. Device-interaction styles are based on several factors, most importantly one or two-handed interaction and landscape and portrait orientation modes. Additional factors that have an influence on overall usability include mobile phone design, mobile phone gestures [Döring, Shirazi, & Schmidt, 2010], usability scenario, application specifications, etc.

2.2.1 One-handed interaction

One-handed interaction represents the interaction in which users hold the mobile device with one hand and perform different operation with the thumb from the same hand. Multiple studies [Karlson & Bederson, 2006], [Parhi, Karlson, & Bederson, 2006], [Karlson & Bederson, 2007], have suggested that one-handed interaction is the most commonly used style of mobile device interaction given the limited mental, physical and visual resources available to users in most usability scenarios. Mobile devices with touchscreen based input such as smartphones present significant challenges for one-handed interaction. This is due to the larger display screen (relative to the size of the thumb), the weight and the shape of the devices, which makes them more difficult to maneuver.

For 3D mobile applications, such as games, medical imaging, visualizations, etc., one-handed interaction might be hard to achieve because certain operations require multi-touch input. Many of these 3D applications rely on touchscreen gestures performed on top of the display screen, such as zoom, which is normally achieved by pinching or spreading two fingers. [Miyaki & Rekimoto, 2009] designed a pressure sensing based interaction technique for one-handed interaction with mobile devices that can overcome the need for two fingers for operations such as zoom or flick but it requires hardware modifications of the mobile device. In one-handed interaction, users can only make use of maximum two fingers for input, the thumb and the index finger (in extreme cases). The optimal and most common case is where users hold the device with one hand and use the thumb from the same hand for input. This is again plagued by other problems such as thumb occlusion [Karlson & Bederson, 2007] and the fact that the thumb cannot reach the extremities of the touchscreen without users having to move the phone with the hand to bring it in position.

2.2.2 Two-handed interaction

While for one-handed interaction users always have a hand free, for two-handed interaction they hold the device with one hand and perform operations with the thumb or finger from the other hand. These styles of interaction are very efficient, especially for complex tasks, but require a larger amount of mental and physical resources which in turn diminishes user preference. Most two-handed approaches are built upon Guiard's kinematic chain model [Guiard, 1987], which assumes an asymmetric relationship between the two human hands. [Hinckley, Pausch, Proffitt, & Kassell, 1998] reject the common-sense viewpoint which states that "two hands save time by working in parallel" and argue for a different view on how the two-handed interaction structure works. Using both hands alters the two main functions of interaction, device-interaction and application-interaction and thus influences the basic problem-solving behavior of users. [Edge & Blackwell, 2009] compared different two-handed interaction styles for touch-screen applications and established that in most cases, users performed touch gestures with the dominant hand while holding the mobile device with the non-dominant hand.

2.2.3 Mobile device orientation

The portrait and landscape orientation modes have become standard in many mobile devices, especially smartphones and tablets. With the widespread integration of inertial sensors such as accelerometers, which are sensitive to both linear acceleration and the local gravitational field [Kirkham, 2010], automatic switching between portrait and landscape modes is now possible. The orientation of the mobile device has a direct impact on one and two-handed interaction. [Nicolau & Jorge, 2012] tested different interaction settings based on combinations of hand conditions such as one-hand/portrait, two-hand/portrait and two-hand/landscape which seems to indicate the strong relation of device orientation and one or two-handed interaction. [Wagner, Huot, & Mackay, 2012] also designed their experiment based on the landscape and portrait orientation modes.

Chapter 3 - Design space

This chapter presents the design considerations for the rotation techniques and the device-interaction styles. Mobile device constraints constitute the major factors in determining design considerations. The design principles upon which the rotation techniques have been investigated in related literature are required to be verified and adjusted to mobile constraints. Moreover, the multitude of device-interaction styles have to be classified and analyzed such that only the most intuitive ones are considered for testing.

3.1 Mobile device constraints

The inherent mobile device constraints have a direct impact on adapting the virtual trackball rotation techniques reviewed in the previous chapter. Constraints such as screen size and thumb occlusion could heavily damper the effectiveness of desktop rotation techniques when ported to mobile devices. Even though they might be efficient for the desktop platform, these rotation techniques can prove to be slow, ineffective and difficult to handle on mobile devices with touchscreen input.

The major problem of virtual trackball techniques for the desktop platform is the information mapping from the lower degree of freedom of the input device to the higher degree of freedom of the 3D rotation task [Kettner, 1995]. This lower-to-higher degree of freedom mapping problem is also present on mobile devices, where users employ touchscreen gestures, since explicit control of rotation about all three axes can only be achieved through multi-touch input. In multi-touch input, users make use of a predefined set of multi-touch gestures, with two or more fingers, in order to perform interaction tasks. Multi-touch input gestures are more appropriate for digital touchscreen table-top displays [Hancock, Cate, & Carpendale, 2009] which are of considerable size and are placed in a fixed position. However, multi-touch input can be inefficient and counter-intuitive for mobile devices given the limited screen size, the thumb occlusion problem and the limited mental and attention resources available.

The problem of obtaining explicit control over 3D rotation remains unsolved. The solution proposed by the Virtual Sphere, Bell's trackball and Shoemake's Arcball is to allow continuous rotation about all three axes inside a circle encompassing the 3D object. While this solution is valid for desktop settings, albeit with some drawbacks, it is rather difficult to integrate on mobile devices. On desktop systems, in order to control the 3D object, users make use of a 2D mouse with a small and precise virtual cursor which allows them to perform precise rotation strokes at the apex of the circle to control the clockwise and counterclockwise movements of the 3D object. On mobile devices, the small and precise virtual cursor is replaced with the finger, an obtrusive and imprecise input "device" and therefore, it is highly unlikely that by using the Virtual Sphere, Bell's VT or Shoemake's Arcball, the same rotation effects can be achieved on mobile devices.

In contrast to the solution mentioned above, the Continuous XY with added Z Controller only allows continuous rotation about the x and y axes inside the circle while for rotation about the z-axis, users need to go outside the circle. Given the limited screen size of mobile devices and the amount of space needed to perform rotations outside the circle, this solution is most likely not viable. However, the idea of using a continuous x-y rotation mapped to the entire screen size would allow users to control the rotation strokes by using the adjacent space of the 3D object thus potentially diminishing the thumb occlusion problem. Hanson's Rolling Ball along with the Two-axis Trackball and the Fixed-right Trackball and Fixed-up Trackball are the only rotation techniques that allow a lower-to-higher degree of freedom mapping which does not impede the rotation process.

3.2 Design principles

For the desktop platform, where rotation is achieved by means of a 2D input device such as a mouse, [Bade, Ritter, & Preim, 2005] identified four general design principles that are considered essential for predictable and pleasing rotation while [Fabien, 2011] added a fifth principle. It is important to determine which of these principles remain valid for the constraints of mobile devices.

Principle 1: Similar actions should provoke similar reactions. This principle refers to the importance of predictable behavior and in the case of 3D object interaction it states that the same mouse movement should yield the same cursor movement. The lack of a typical user-mouse-cursor metaphor on mobile devices suggests that this principle is not required for 3D object interaction on mobile devices and can be replaced by the second principle.

Principle 2: Direction of rotation should match the direction of 2D pointing device movement. The second principle applies to stimulus-response compatibility and kinesthetic correspondence between the direction of user action (mouse movement) and the direction of computer reaction (3D object rotation). For mobile devices, the mouse movements are replaced with touchscreen gestures and the computer is replaced with a mobile device.

Principle 3: 3D rotation should be transitive. This third principle refers to the natural transitive behavior of movements in a Euclidean space which enables users to return to the initial viewing position. Device movement from point A to point B and then to point C should be the same as a direct movement from A to C.

Principle 4: The control-to-display (C/D) ratio should be customizable. While this principle is perfectly valid for the desktop platform, for 3D object interaction on mobile devices, the absence of a remotely controlled cursor means there is a 1:1 correspondence between motor and display movement and thus no control-to-display ratio to manipulate [Karlson & Bederson, 2007].

Principle 5: *The input interface should allow the setting of an orientation by an integrated manipulation.* The fifth principle is based on the work done by [Hinckley, Tullio, Pausch, Proffitt, & Kassell, 1997] in a formal user study where it is concluded that the mental model of rotation is separated in each orientation view that can be manipulated. In other words, the input interface should be designed to enable a simultanenous variation of each degree of freedom of the rotation [Fabien, 2011]. On mobile devices, controlling multiple degrees of freedom simultanenously can only be achieved through multi-touch input or by using inertial sensors.

Although Principle 1, Principle 4 and Principle 5 are important design principles, especially for desktop settings and direct/indirect-touch tabletop displays, they are not necessarily valid for this current project and therefore are not considered as an assessment tool for virtual trackball techniques on mobile devices. None of the rotation techniques in Table 3-1 respect both Principle 2 and Principle 3. Hanson's Rolling Ball is capable of two basic rotations about an axis in the screen plane and rotation about the screen perpendicular axis. This property fulfills Principle 3 but does not respect Principle 2. In some cases, the orientation of the object can influence the transitivity of rotations which leads to Principle 3 not being respected for those special cases. The Two-axis Trackball fulfills Principle 2 since the direction of rotation always matches the direction of finger movement. However, a combination of two rotations using the Two-axis Trackball is not transitive and therefore Principle 3 is not fulfilled. The Fixed-right Trackball and Fixed-up Trackball offer the advantage of transitive rotations meaning that Principle 3 is always fulfilled but since the rotation direction is contrary to the direction of finger movement if the 3D object is rotated upside down, Principle 2 is not respected. Since Hanson's Rolling Ball meets only one principle and partially the other one it has not been considered for further evaluation.

Table 3-1 Comparison of rotation techniques based on design principles

Rotation techniques / **Design Principles**	Hanson's Rolling Ball	Two-axis Trackball	Fixed-right Trackball	Fixed-up Trackball
Principle 2	-	+	-	-
Principle 3	+/-	-	+	+

3.3 Rotation techniques for mobile devices

This section presents a detailed description of the Two-axis Trackball, Fixed-right Trackball, Fixed-up Trackball and their innate control mechanism for 3D object rotation on mobile devices (Figure 3.1).

3.3.1 Two-Axis Trackball

The image plane of the mobile device is regarded as having two orthogonal axes x and y, parallel to the borders of the mobile device. All finger movements on the touchscreen such that from p_{i-1} to p_i, where p_i is the current position of the finger at time i are resolved into components dx and dy. The values of dx and dy can then be used to control the rotation of 3D objects about different axes.

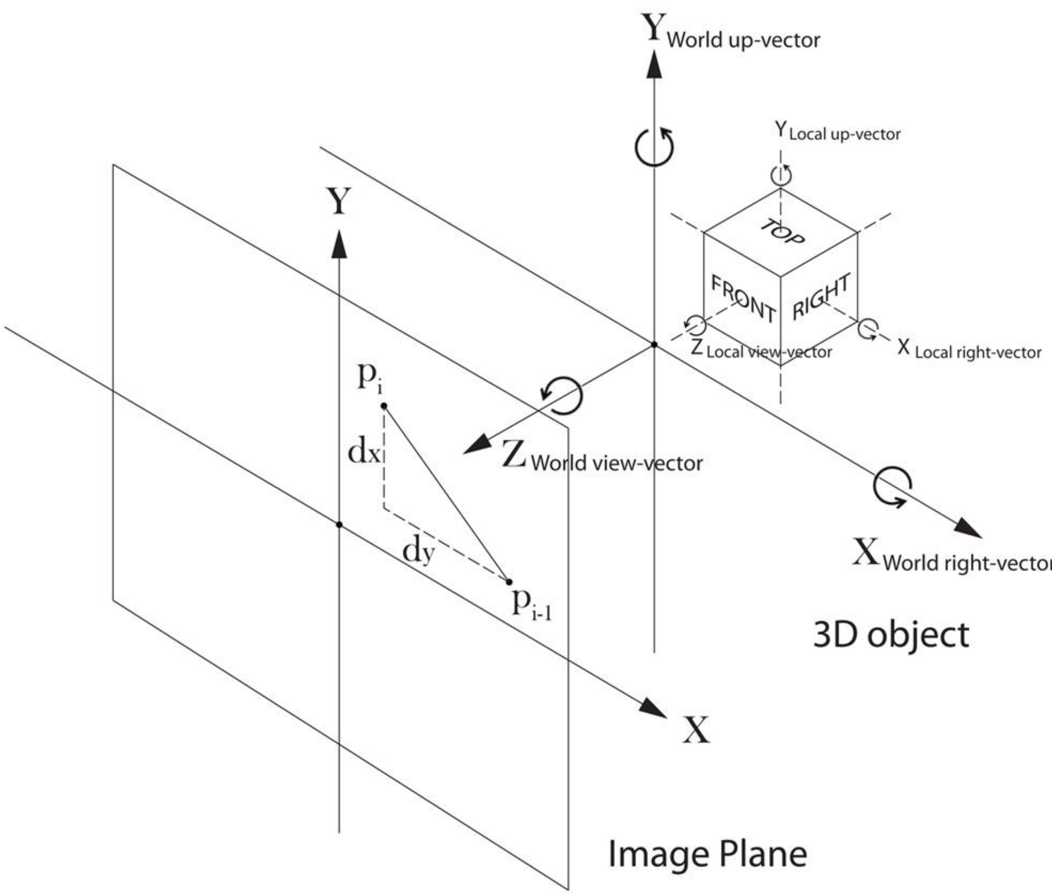

Figure 3.1 Two-axis trackball rotation technique

The position of the finger throughout the entire rotation process is irrelevant for all rotation techniques and only the difference between the previous and current finger position is needed to calculate the rotation amount and rotation direction. Thus, horizontal finger movements on the touchscreen of the mobile device will generate rotations about the world-up vector (y axis) and vertical finger movements will generate rotations about the world-right vector (x axis) while diagonal finger movements will generate rotations about a combination of both world-up and world-right vectors (y and x axes). The Two-axis Trackball does not offer any explicit rotation about the world view-vector (z axis).

3.3.2 Fixed-up Trackball

While the Two-axis Trackball generates rotations only about world space vectors, the Fixed-up Trackball uses a combination of world and local space vectors and maps horizontal finger movements about the local up-vector of the 3D object and vertical finger movements about the world right-vector. Diagonal movements are mapped to a combination of local up-vector and world-right vector. The combined mapping of both world and local space allows for rotations about the view-vector to be generated but also causes a loss in predictability, especially when the 3D object is turned upside down as can be seen in Figure 3.2. Moreover, diagonal movement generates very unnatural rotations.

3.3.3 Fixed-right Trackball

The Fixed-right Trackball also uses a combination of world and local space vectors similar to the Fixed-up Trackball but they are inverted meaning that horizontal finger movements are now mapped about the world-up vector and vertical finger movements about the local right-vector. Using the world's fixed-look vector is also possible, thus building the Fixed-look Trackball, but the generated rotations are far too unpredictable to contribute to 3D object rotation in any meaningful way.

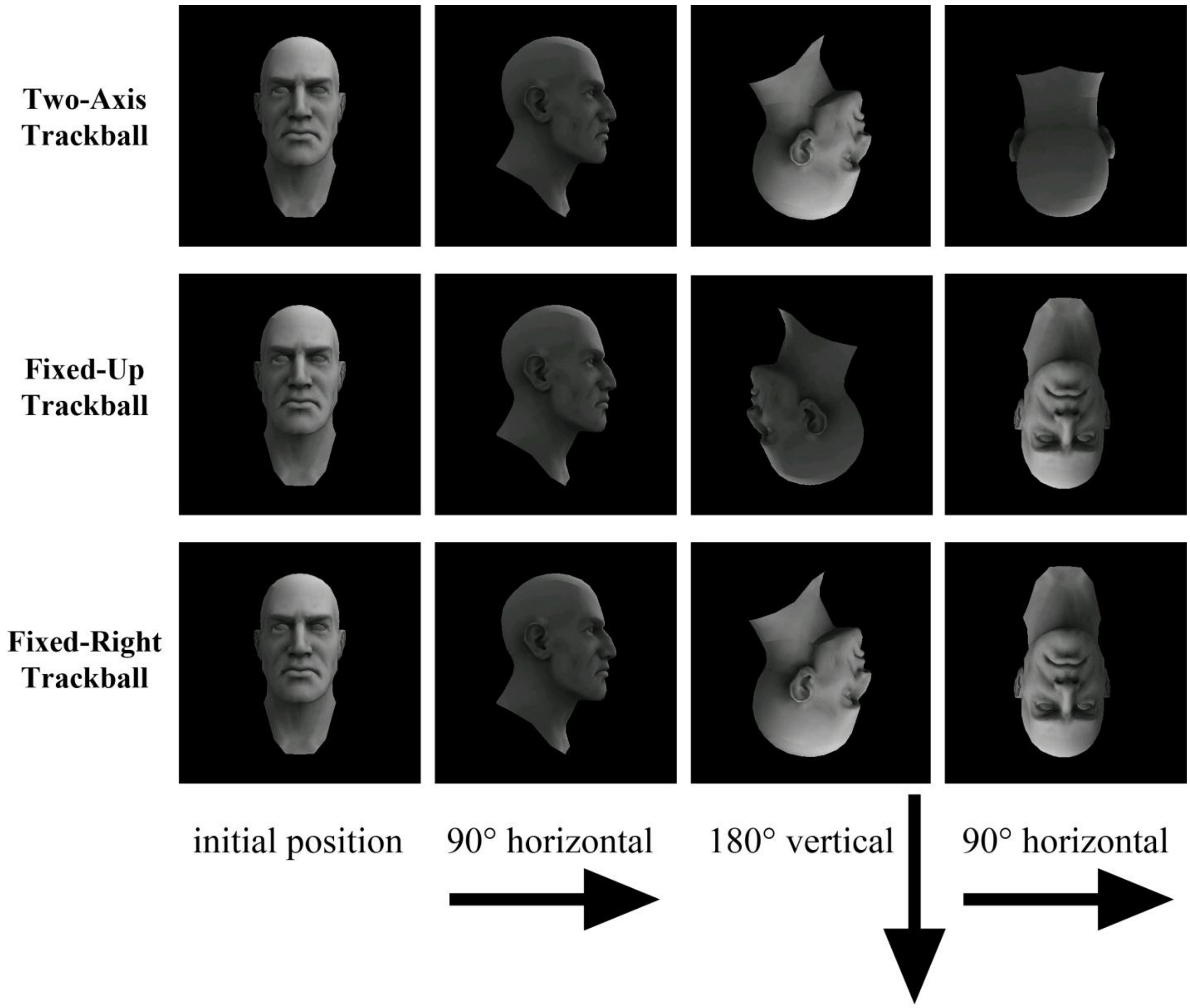

Figure 3.2 Different rotations generated by the three rotation techniques using the same finger movement

3.3.4 Fixed-Right Trackball with increased rotation amount

[Jacob & Oliver, 1995] define 3D rotation as having two parameters: “the axis around which the rotation is to be made and the amount of rotation”. For mobile devices, where limited screen sizes do not allow long continous strokes of rotations, increasing the rotation amount might provide a way to decrease task completion time and improve accuracy. The dimensions of the display screen of the mobile device used in this experimental comparison are: 92 mm in landscape, 55 mm in portrait and 10.7 mm diagonally (Figure 3.3). This is the usable size that can be used to perform touchscreen gestures. The default amount of rotation that can be performed by dragging a finger across the screen has been defined as: 270 degrees for portrait, 450 degrees for landscape and 524 degrees for diagonal (Table 3-2). It is important to mention that the amount of rotation is not affected by the size of the 3D object. For the Fixed-right trackball with increased rotation amount the rotation is increased with 125 degrees for portrait, 180 degrees for landscape and 230 degrees for diagonal. These settings were established during preliminary tests and informal evaluations prior to this experimental comparison.

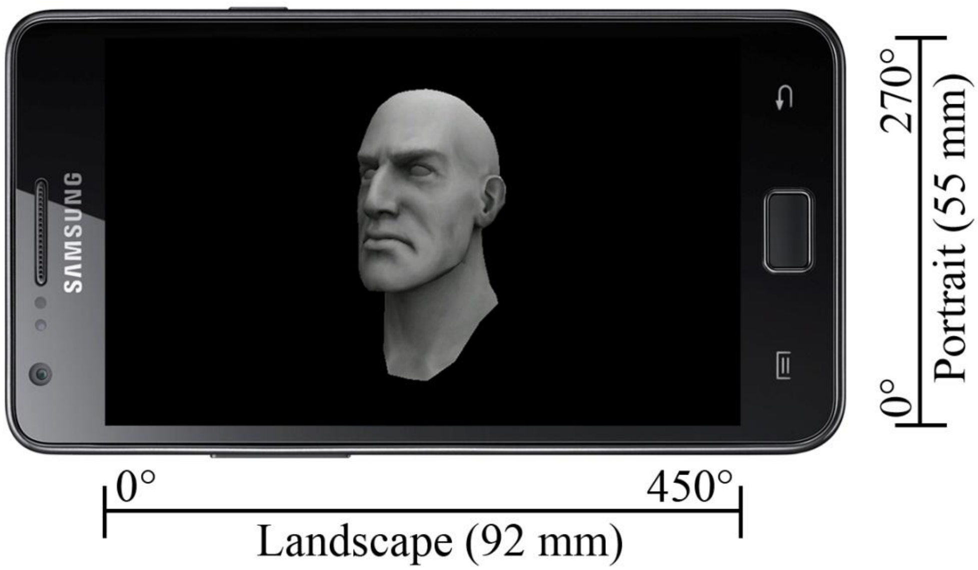

Figure 3.3 Default rotation amount (in degrees) across the usable screen size (in mm)

Table 3-2 Rotation amount for each rotation technique

Techniques	**Fixed-up trackball**	**Fixed-right trackball**	**Fixed-right trackball w/ increased rotation amount**
Portrait	270°	270°	400°
Landscape	450°	450°	630°
Diagonal	524°	524°	764°

3.4 Mobile device interaction styles

Device-interaction styles are based on the number of hands used (one or two-handed interaction), the orientation of the device (landscape or portrait) and the use of either the thumb or the finger. Presented below are all the possible device-interaction styles available for the smartphone.

3.4.1 Two-handed/landscape interaction

As depicted in Figure 3.4, there are two device-interaction styles available for two-handed interaction in landscape mode. Users can hold the mobile device with two hands and utilize either the left or the right thumb to perform rotation strokes. This device-interaction style is called two-handed/landscape with thumb. The alternative is to hold the mobile device with one hand and perform rotation strokes with the finger from the other hand. An important thing to note is that users cannot use two thumbs, two fingers or a combination of finger and thumb at the same time on the display screen. This is because the application was not designed to support multi-touch interaction and therefore cannot register more than one finger or thumb at a time.

Figure 3.4 Two-handed/ landscape with thumb interaction (left) and two-handed/landscape with finger interaction (right)

3.4.2 Two-handed/portrait interaction

The same device-interaction styles available for landscape can be employed for two-handed interaction in portrait mode as can be seen in Figure 3.5. However, because it aggravates the thumb occlusion problem, the two-handed/portrait with thumb interaction style is considered inadequate and inefficient for 3D object interaction on smartphones. This device-interaction style might be more appropriate for mobile devices with bigger display screen size such as tablets. Therefore, the two-handed portrait with thumb interaction style was not included in the experimental study.

Figure 3.5 Two-handed/portrait with finger interaction (left) and two-handed/portrait with thumb interaction (right)

3.4.3 One-handed/landscape and one-handed/portrait interaction

For one-handed interaction in both landscape and portrait modes, the available device-interaction style is to hold the mobile device with one hand and perform rotation strokes with the thumb. In landscape mode this approach is rather counter-intuitive and ineffective since users have a hard time stabilizing the mobile device. Moreover, because of the hand’s unnatural position, the thumb has a limited range of motion. It has been therefore excluded from the experimental study. In portrait mode, however, this approach seems very natural and matches typical operations on mobile devices such as texting, calling, visualizing images, etc.

Figure 3.6 One-handed/portrait with thumb interaction (left) and one-handed/landscape with thumb interaction (right)

Chapter 4 - Experimental setup

A series of experiments were carried out in order to evaluate the concepts described in the previous chapter. In the first experiment, three rotation techniques were investigated, namely the Fixed-right Trackball, the Fixed-up Trackball and the Fixed-right Trackball with increased rotation amount along with two device interaction styles, the two-handed/landscape with finger and the two-handed/landscape with thumb. The results from this first experiment together with additional related literature reviews led to several more influential concepts such as the Two-axis Trackball and more device-interaction styles such as two-handed/portrait with finger and one-handed/portrait with thumb. This chapter presents a thorough description of all the details involved in setting up both experiments along with the process and methods of collecting data.

4.1. Common experimental setup

Consistency is an important factor when designing an experimental setup, especially when it involves multiple experiments. Therefore, the two experiments share a number of common elements that are specifically kept consistent to insure a minimum level of bias when comparing the data of both experiments. These common elements include the mobile device used to develop, run and test the application as well as the experiment tasks (with the exception that in the second experiment three new tasks were introduced), scenes, 3D objects, emoticons, lighting and the subsequent settings of these elements.

4.1.1 Apparatus

The mobile device used for developing and performing the tasks for both experiments is a Samsung I9100 Galaxy S2 (Figure 4.1) which, according to a report by [GSM Arena, 2011], has a dual-core 1.2 GHz Cortex-A9 processor, 1GB RAM, Mali-400MP graphics processor, running on Android 2.3.3 Gingerbread. These powerful hardware features allow for a smooth render of 3D objects at high frame-rates. The smartphone’s dimensions are 125.3mm x 66.1mm x 8.49mm and a weight of 116g. The display is a Super AMOLED Plus capacitive touchscreen with a maximum resolution of 480 x 800 pixels. These new types of mobile display screens have integrated digitizers [Neowin, 2012] that allow for better response time and higher refreshes rates.

Figure 4.1 Mobile device used in the experimental comparison (Samsung I9100 Galaxy S2)

4.1.2 Scenes

The general setup of a scene is presented first along with details about the role of each element. A thorough description of the 3D objects and emoticons, two of the most important elements of the scenes, is given next. The secondary elements of scenes such as background color and lighting are less obvious to users but require an appropriate structure to eliminate bias.

The controlled variables of a scene are: 3D objects, emoticons, lighting, and background color. Each task has five scenes plus an additional tutorial scene. The tutorial scene contains a 3D object representing a human head while the other five scenes each contain one 3D object, namely cube, sphere, pyramid, ceramic and vase. An invisible circle, 40 mm in diameter, with the center of origin coinciding with the center of mass [Mirtich, 1996] of a 3D object is drawn. The role of this circle is two-fold. First, it insures that the size of 3D objects is kept consistent and second, it is also a reference for the pivot point of rotation. All 3D objects are assigned a light grey color while the background of the scene is assigned a black color. Emoticons are visual targets drawn on the surface of 3D objects, on top of the light grey color and are categorized by type (happy face, straight face and sad face) and color (red, blue and yellow) and for certain tasks combinations of the two categories are used which leads to a total of nine emoticons. The lighting is static and does not change regardless of how the object is rotated in relation to the view-camera. Half of the 3D object is lit and the other half presents a diffuse shadow.

4.1.2.1 3D objects

3D objects are represented by two types of shapes, basic shapes such as cube, sphere and pyramid and more complex shapes such as ceramic, vase and human head as can be seen in Figure 4.2. In choosing these shapes, several criteria were considered. First, taking into account the potential uses of rotation techniques such as examining 3D objects in museum exhibitions, virtual catalogues and scientific visualization, the types of objects that users generally encounter in real-life applications can be deducted. Second, there is a need to eliminate any advantages users might gain by integrating 3D objects which are not axisymmetric (rotational symmetric on at least one axis) such as plants, trees, organic shapes, etc. Third, a wide majority of studies regarding 3D object interaction use generic 3D objects such as human head [Partala, 1999] ,house [Chen, Mountford, & Sellen, 1988], [Hinckley, Tullio, Pausch, Proffitt, & Kassell, 1997], but [Bade, Ritter, & Preim, 2005] emphasize the need for more complex objects that do not provide conclusive hints regarding the object's orientation. The final criterion in choosing these 3D objects is the need for sufficient surface space for the placement of visual targets (emoticons).

The placement of emoticons on the surface of 3D objects leads to another problem, namely the UV mapping process [Lanier, 2008] which involves assigning pixels in an image to surface mappings on the polygon. In order to place emoticons on 3D objects, the 3D object needs to be unwrapped as a two dimensional texture called a UV texture map. The letters "U" and "V" denote the axes of the 2D texture map. The UV mapping process involves three steps, unwrapping the 3D object, creating the texture and applying the texture on the 3D object. When unwrapping the 3D object, distortions can occur depending on the shape of the 3D object. Therefore, in order for emoticons to be efficiently displayed it is important to evenly align the U and V coordinates of the 3D objects. This is done in Autodesk Maya, a specialized 3D modeling application.

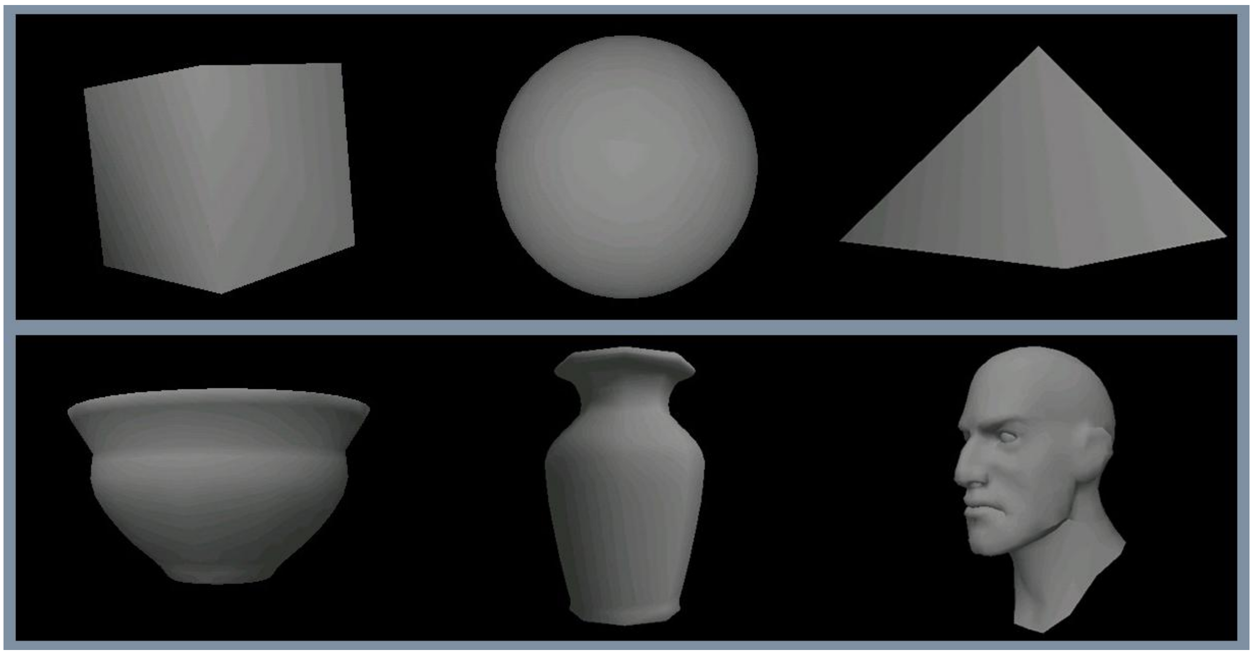

Figure 4.2 3D objects used in the experimental comparison

The resolution of the 2D texture maps is also an important element that needs to be kept consistent. All 3D objects have the same UV coordinate resolution eliminating any inconsistencies between the emoticons' size in relation to blurriness. The resolution of the 2D texture map is set at 256x256 pixels and the size of emoticons ranges from 16x16 pixels to 26x26 pixels. The 16x16 pixels is the minimum resolution at which users can safely distinguish the emoticons without having to move the phone towards them as was discovered in a previous study [Buda, 2012].

4.1.2.2 Emoticons

Emoticons are two-dimensional visual targets based on color and type which are placed on the surface of 3D objects. There are three types of emoticons: happy face, straight face and sad face and three colors: red, blue and yellow. By combining them, nine possible combinations of emoticons can be obtained. There are several advantages to using emoticons for this particular research study. Users should be required to perform rotation strokes in as many directions as possible which would translate in real life when someone wants to inspect an object. Therefore, emoticons fit perfectly in this scenario since they can be oriented in different directions so that users will have to rotate carefully to identify the correct emoticon. Another advantage of using emoticons is that users are already familiar with their shapes and usage eliminating any bias or learning curve that might occur if different visual targets would be used.

The three emoticon types were chosen based on their wide-spread familiarity and the fact that they are similar enough in style (eyes, top curve, bottom curve and line) to make sure users carefully inspect them and not simply guess the result. They basically contain the same elements (eyes and a straight line which is then bent to one side and to the other) while other emoticons have additional elements such as teeth which would make it easier for users to identify them but harder to remember.

For certain tasks, emoticon types are set with three colors, namely red, blue and yellow. There are several reasons for choosing these colors. First, the association of green and red has to be eliminated because of users who might suffer from color blindness. Second, these colors are primary colors, most commonly known as the RYB color model [Gurney, 2010]. Third, the chosen colors are sufficiently wide apart on the color spectrum so users cannot confuse them with other close colors. Finally, the contrast between the black background of the scene, the light grey color of the 3D objects and the colors of emoticons is good enough to insure that appropriate discrimination between these colors can easily be made.

4.1.3 Rotation tasks

[Henriksen, Sporring, & Hornbæk, 2004] stated the importance of using a broader selection of tasks in evaluations of rotation techniques. They recommend using tasks that are closer to actual work tasks, where attention has to be divided between controlling rotation and solving the work task. [Bade, Ritter, & Preim, 2005] also suggested using more complex rotation tasks that might reveal potential weaknesses in rotation techniques. While most studies rely on orientation tasks, where users have to align a 3D object to a certain orientation based on a visual indicator, the more complex inspection tasks, involving target discovery, target counting, target distance estimation, target identification and target spatial association on the surface of 3D objects appear to be more effective in discovering weaknesses and strengths in rotation techniques (Figure 3.1). A comprehensive description of the rotation tasks used in both experiments are presented below.

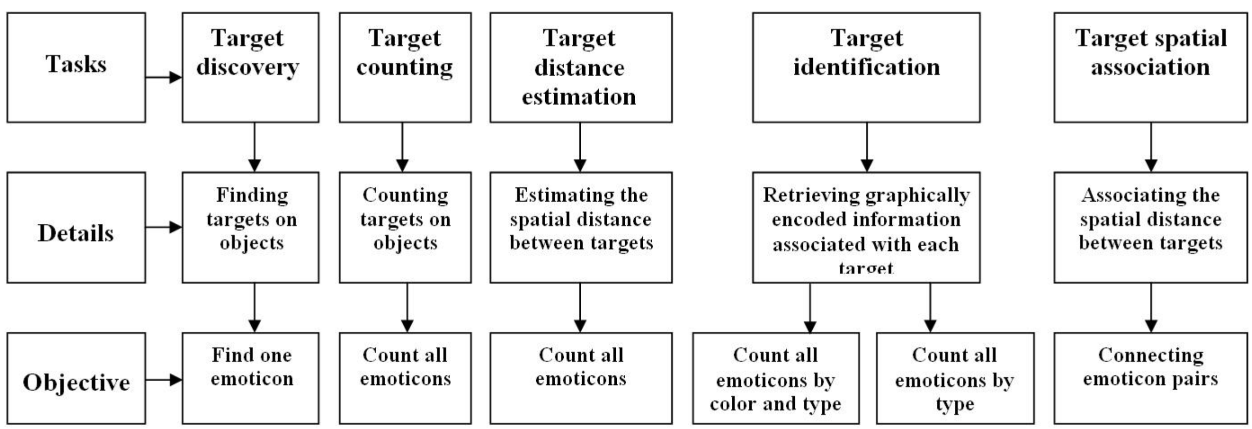

Figure 4.3 Rotation tasks used in the experimental comparison

4.1.3.1 Find one emoticon

For the "find one emoticon" task users have to identify one particular emoticon which has been placed on the surface of the 3D objects by naming the color and type of that specific emoticon. For example, if a 3D object contains an emoticon of red color and straight face type then users need to name it as "red straight face". This task serves a double purpose. On the one hand, it is an appropriate way of insuring that users can actually distinguish the correct color and type of emoticons. For both experiments, during this task, participants who had problems telling the color or type of emoticons were identified and their results were not taken into account. Since the task of finding one emoticon is always the first one to be done, it can be used to validate all other following tasks. On the other hand, it gets users familiar with the rotation techniques, mobile device controls, general interface and device interaction styles along with the procedure of collecting data. Emoticons are only placed on the half back side of the object (180 degrees) which faces away from the user. They are placed in such a way that they cover different parts from one side to another and are orientated with a minimum of 20 and a maximum of 45 degrees to insure that users have to rotate the object slightly to get a better view of the correct emoticons.

4.1.3.2 Count all emoticons

The task of counting all emoticons requires users to count all the emoticons drawn on a 3D object regardless of the type and color. Users need to express their answer in a numerical fashion and not by naming each emoticon in part. For example, if a 3D object has five emoticons on its surface then users need to verbalize the number five. In an experimental study regarding rotation techniques for 3D object interaction on mobile devices using touchscreen gestures, [Decle & Hachet, 2009] and [Kratz & Rohs, 2010] used the task of counting all emoticons drawn on a 3D object as the primary task of their experiment. They use this task to measure the performance and preference of users over two rotation techniques and motivate their choice by stating that the task requires users to rotate all sides of the object very carefully therefore forcing users to perform rotation strokes in all directions. This task has been adapted to suit the purpose of this project and by carefully placing emoticons on 3D objects it can be made certain that users will need to rotate the object in all directions. In designing this task, a decision was made to place a minimum of four emoticons and a maximum of nine emoticons on a 3D object to insure that all sides of the object are covered (minimum) and that only unique emoticons are considered (maximum).

4.1.3.3 Estimate distances

The task of estimating distances between emoticons requires users to search for two emoticons with the shortest distance between each other. All 3D objects have three emoticons placed on their surface. There are two emoticons of the same color but of different type and a third emoticon of a different color. This third emoticon is always positioned closer to one of the two emoticons with the same color. Users have to identify and verbalize to which of the two emoticons is the emoticon with the different color closest to. Whereas tasks such as finding one emoticon and counting all emoticons require users to rotate the 3D object on all sides to successfully complete the task, estimating distances is more about small back-and-forth rotations that tests how well the rotation techniques deal with spatial distance between visual cues on a 3D object.

Figure 4.4 Finding one emoticon (left), Counting all emoticons (center) and Estimating distances (right)

4.1.3.4 Count particular emoticons by color and type

While the task of counting all emoticons on a 3D object is a very useful rotation task which provides important information about how users count the visual cues, it is essentially a simple and ordinary task and it does not force users to make really small rotation strokes. Counting particular emoticons based on a combination of color and type adds another level of detail and forces users to inspect the 3D object even more carefully and split the rotation process into small increments. Users are informed at the beginning of each scene about the particular emoticon that they need to count. For example, they might need to count all the straight red face emoticons. For this particular task, there are a minimum of one and a maximum of four emoticons drawn on the surface of the 3D object.

4.1.3.5 Connect emoticon pairs

Connecting emoticon pairs requires users to identify connections between three pairs of emoticons that are inter-linked by lines. The emoticon pairs are always of different color and only one pair of emoticons per object is of the same type, for example, yellow sad face connected to blue straight face and red happy face connected to blue happy face. The line that connects the emoticon pairs is a dark gray and the size of the line is clearly distinguishable and does not impede the process of identifying the connected pairs. This task tests the ability to follow established rotation patterns on a 3D object.

4.1.3.6 Count particular emoticons by type

Counting particular emoticons by their type represents an even deeper level of search because the type is harder to spot than a combination of color and type.

Figure 4.5 Count particular emoticons by color and type (left), Connect emoticon pairs (center) and Count particular emoticons by type (right)

4.1.4 Counterbalancing

In order to minimize potential problems such as confounding among variables or the carry-over effect and to control order effects in a repeated measures design, counterbalancing was employed. Counterbalancing is a method used in usability testing for establishing task order such that each condition appears in each position an equal number of times. In the case of this project, there is one group of participants per experiment that are testing several manipulated variables (rotation techniques and mobile device interaction styles). For each of these manipulated variables, there are controlled variables such as tasks, scenes, 3D objects, etc., that have to be counterbalanced. In order to counterbalance the order of these controlled variables it is important to find all the possible ordering combinations and arrange them based on a Latin squares design. The color, type, placement, orientation and size of the emoticons along with the type and orientation of the 3D objects are also counterbalanced in the same manner.

4.1.5 Methods of data collection

The measured variables of this project are the average task completion time, total and average number of mistakes made per task, and user preference. These elements are measured using a two-part questionnaire which contains the user preference, verbalization of answers for tasks which deals mostly with the number of mistakes and log files that record the time spent on each task.

4.1.5.1 Two-part questionnaire

For both experiments, two part questionnaires were used (Appendix A and Appendix B). There are differences between these two questionnaires since they target different independent variables but there are also a lot of common elements. In the first part of the questionnaire, participants were required to answer more general questions regarding their age, occupation, dominant hand, whether or not they have vision problems, smartphone ownership, interest and preoccupation with mobile device applications or games and familiarity with rotation of 3D objects. The second section of the questionnaire, which was presented to participants after completing all tasks, focused on measuring subjective satisfaction (the difficulty and the intuitiveness of certain rotation techniques and device interaction style), the willingness to utilize these options in a real life situation as well as suggestions for future implementations.

4.1.5.2 Verbalization

While performing the tasks, participants verbalized their responses to the observer which then wrote these responses down on a piece of paper. There are several reasons for choosing the method of verbalizing instead of selecting from a list of in-application options. First of all, by verbalizing, participants are required to maintain a high level of concentration which yields more accurate results thus making sure that a lack of appropriate focus from the participants does not affect the results. Second, significant answer time was saved by eliminating an in-application option list for each answer. Third, participants are more inclined to also comment during the tasks on issues such as a recurring difficulty of a particular 3D object, which was the case for the sphere, the physical buttons on the side of the smartphone which were pressed by mistake multiple times, the static lighting, the seams in the 2D texture maps of the 3D objects, etc. Another example of this verbal feedback is when participants had to count all the emoticons, which seemed to be a very demanding task and a wide majority of participants made remarks such as “this is hard”, “very annoying“, “I wish I had an indicator somehow”. Such remarks are indicative of how much effort a participant puts into solving a particular problem as well as what the patterns of behavior are when dealing with problem solving tasks.

4.1.5.3 Log files

For each experiment scene, a log file that contains the time spent solving the task for that particular scene is saved on the smartphone’s hard-disk. Time is recorded from the moment users enter the scene until they exit that particular scene. In cases where participants accidentally exited the application or the application crashed while in a scene, they had to redo the scene and the log file was overwritten.

4.2 First experiment

The manipulated variables of the first experiment consist of the following rotation techniques: Fixed-right Trackball, Fixed-up Trackball, Fixed-right Trackball with increased rotation amount and device-interaction styles: two-handed/landscape with thumb and two-handed/landscape with finger. The measured variables are the average task completion time, total and average number of mistakes made per task, and user preference.

4.2.1 Participants

A number of 25 experts in the field of game and media technology (2 females and 23 males), with an average age of 23.96 years and with a high knowledge of mobile applications and games, took part in the experiment. Among these participants, 24 are right-handed and one participant left-handed, 11 reported vision problems, 20 owned a smartphone and 13 said that they were already familiar with rotation of 3D objects on mobile devices.

4.2.2 Procedure and instructions

Participants were first briefed about the overall procedure, the measured and manipulated variables, the methods of collecting data and the duration of the experiment. They were then presented with a two-part questionnaire for which they had to provide personal details regarding their age, occupation, dominant hand, etc., and answer questions about any vision problems, smartphone ownership, interest and preoccupation with mobile applications and games and familiarity with rotation of 3D objects. Instructions on how to operate the mobile device and how to complete each task were also given in the beginning. They were told to hold the smartphone in a comfortable position with both hands in landscape mode and that during the experiment they are not allowed to rotate the smartphone itself and they cannot position it on the table. Before each task, participants were given appropriate information on how to complete the task and they could also practice on a tutorial scene. After completing all tasks, participants had to fill in the second part of the questionnaire and answer questions regarding task difficulty, preference for one measured variable over the other, etc. At the last question, regarding additional options that could be implemented to improve the 3D object interaction, participants were shown several applications which included global rotation, zoom, incremental rotation and flicking.

4.3 Second experiment

The findings from the first experiment raised questions as to how the one-handed and two-handed interaction in both landscape and portrait modes would influence performance and user preference. While in the first experiment landscape mode was mandatory, the focus for the second experiment expanded to cover additional mobile device-interaction styles. Also, related literature along with the suggestions from users indicated that another rotation technique might be more appropriate for 3D object rotation than the ones previously tested. The manipulated variables are the Two-axis Trackball, Fixed-up Trackball and the different mobile device-interaction styles. The measured variables are the same from the first experiment.

4.3.1 Participants

A number of 25 participants (6 females and 19 males), with an average age of 26.12 years and with a moderate knowledge of applications and games for mobile devices, took part in the experiment. Among these participants, 6 of them are left-handed and 19 right-handed, 7 reported vision problems and 16 owned a smartphone (Table 4-1).

Table 4-1 Comparison among the participants of the two experiments

Experiments	**First experiment**	**Second experiment**
Number of participants	25	25
Average participant age	23.96	26.12
Males	23	19
Females	2	6
Left handed	1	6
Right handed	24	19

4.3.2 Procedure and instructions

The general procedure for the second experiment is similar to the first experiment with some exceptions. The two-part questionnaire from the second experiment contained more specific questions about the way in which participants interact with mobile devices, whether they use one or two hands, what orientation mode preferred and the familiarity with 3D applications. For finding one emoticon, participants completed the task using the Two-axis Trackball or Fixed-up Trackball in either landscape or portrait mode so they could get used to the two orientation modes. After that, at the beginning of all other tasks, participants were allowed to choose which one of these two modes (landscape or portrait) they wanted to use for the task at hand.

Chapter 5 - Results

In this chapter, the results of the first and second experiments are presented. These results include performance measurements such as average task completion time, total and average number of mistakes per task, accuracy rates, as well as more subjective indicators, in particular user preference.

5.1 First experiment

The first experiment evaluated three rotation techniques: Fixed-right Trackball, Fixed-up Trackball, Fixed-right Trackball with increased rotation amount, and two device-interaction styles: two-handed/landscape with thumb and two-handed/landscape with finger.

5.1.1 Fixed-right Trackball vs. Fixed-up Trackball

A paired two samples for means t-test revealed no significant differences between the average task completion times (Figure 5.1) of the Fixed-right Trackball and the Fixed-up Trackball for finding one emoticon ($p=0.5107 > 0.05$), counting all emoticons ($p=0.4852 > 0.05$) and estimating distances ($p=0.3168 > 0.05$). The Fixed-right Trackball yielded a higher number of mistakes for counting all emoticons and estimating distances than the Fixed-up Trackball (Figure 5.2) which equates to a lower accuracy by the Fixed-right Trackball (Table 5-1).

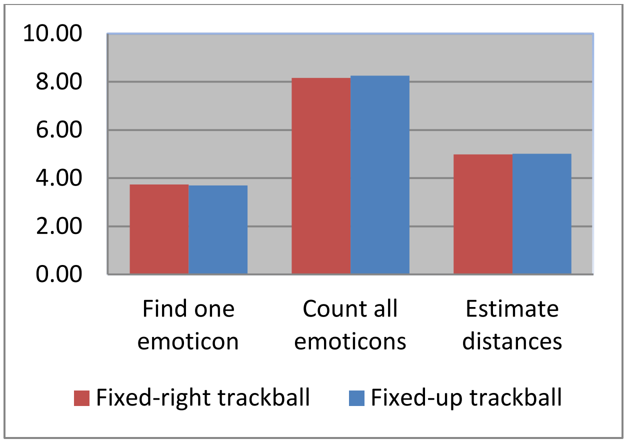

Figure 5.1 Average task completion time (in seconds) for the Fixed-right Trackball and Fixed-up Trackball

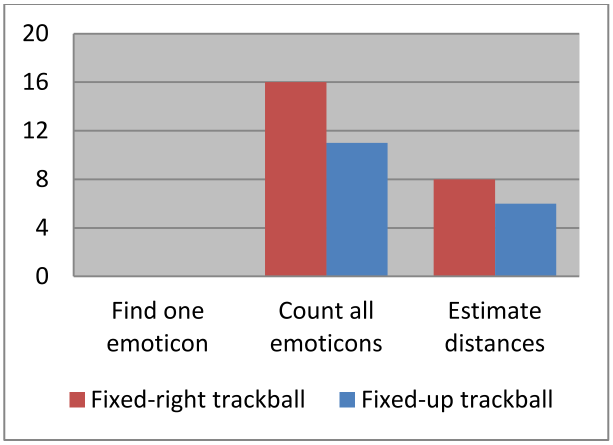

Figure 5.2 Total number of mistakes per task for the Fixed-right Trackball and Fixed-up Trackball

Table 5-1 Accuracy rates per task (in percentages) for the Fixed-right Trackball and the Fixed-up Trackball

Rotation techniques / Tasks	**Fixed-right Trackball**	**Fixed-up Trackball**
Find one emoticon	100%	100%
Count all emoticons	33%	54%
Estimate distances	67%	75%

5.1.2 Fixed-right Trackball vs. Fixed-right Trackball with increased rotation amount

In terms of average task completion time, the Fixed-right Trackball was slower than the Fixed-right Trackball with increased rotation amount, for finding one emoticon and estimating distances and faster for counting all emoticons (Figure 5.3). A paired two sample for means t-test found significant differences for finding one emoticon ($p=6.44\times10^{-9} < 0.05$), counting all emoticons ($p=0.0303 < 0.05$) and estimating distances ($p=1.66\times10^{-9} < 0.05$). The number of mistakes made with the Fixed-right Trackball was higher than the number of mistakes made with the Fixed-right Trackball with increased rotation amount for all tasks (Figure 5.4). Also, the accuracy rates, presented in percentages in Table 5-2, were lower for the Fixed-right Trackball.

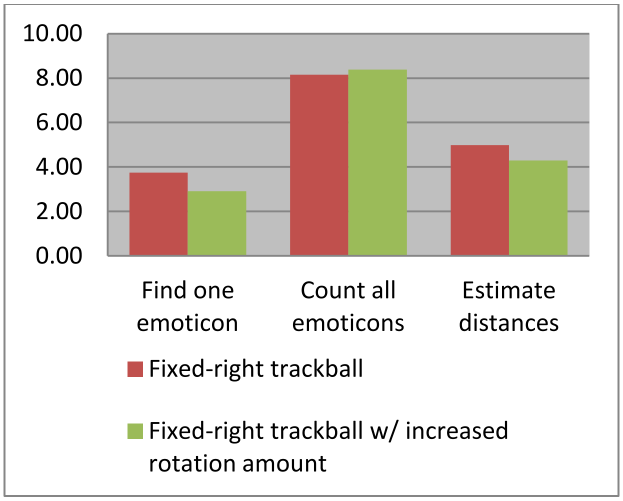

Figure 5.3 Average task completion time (in seconds) for the Fixed-right Trackball and Fixed-right Trackball with increased rotation amount

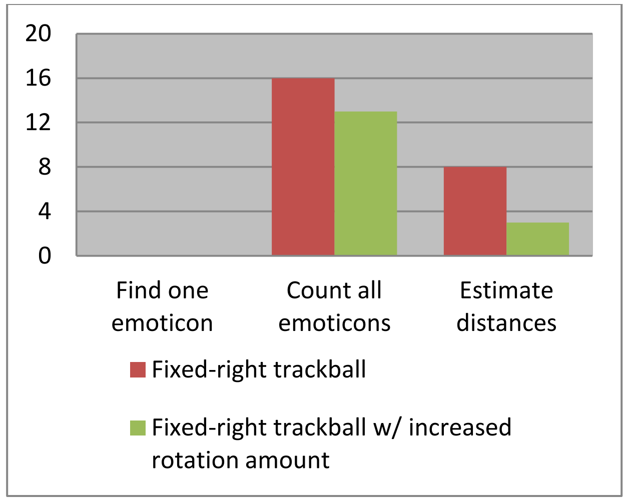

Figure 5.4 Total number of mistakes per task for the Fixed-right Trackball and Fixed-right Trackball with increased rotation amount

Table 5-2 Accuracy rates per task (in percentages) for the Fixed-right Trackball and the Fixed-right Trackball with increased rotation amount

Rotation Techniques / Tasks	**Fixed-right trackball**	**Fixed-right trackball w/ increased rotation amount**
Find one emoticon	100%	100%
Count all emoticons	33%	46%
Estimate distances	67%	87%

5.1.3 Fixed-up Trackball vs. Fixed-right Trackball with increased rotation amount

The Fixed-right Trackball with increased rotation amount proved to be faster than the Fixed-up Trackball for finding one emoticon and estimating distances (Figure 5.5). A paired two sample for means t-test revealed that, with the exception of the task of counting all emoticons, ($p=0.41 > 0.05$) there are significant differences between the average task completion times of the two rotation techniques for finding one emoticon, ($p=9.98\times10^{-12} < 0.05$) and estimating distances ($p=2.22\times10^{-8} < 0.05$). The Fixed-up Trackball yielded fewer mistakes for counting all emoticons, although the accuracy rates are very close (Table 5-3), and more mistakes for estimating distances (Figure 5.6).

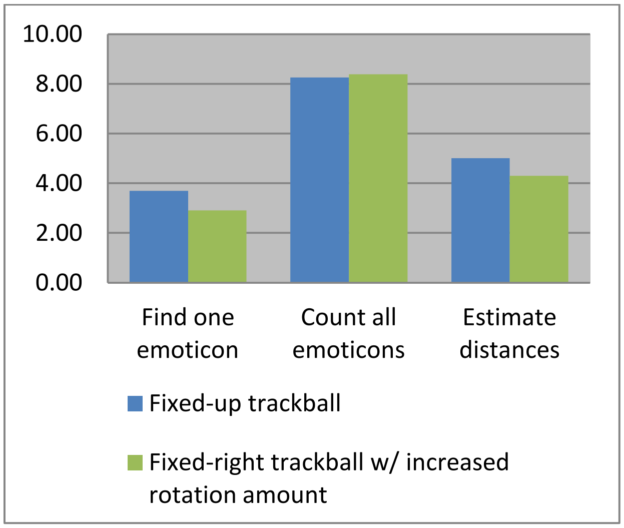

Figure 5.5 Average task completion time (in seconds) for the Fixed-up Trackball and Fixed-right Trackball with increased rotation amount

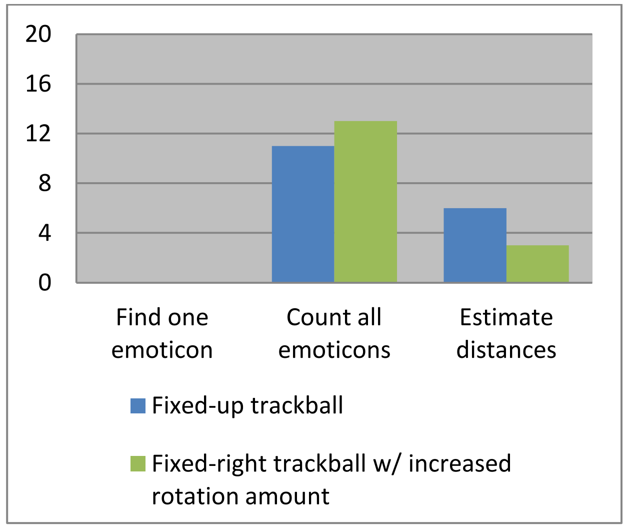

Figure 5.6 Total number of mistakes per task for the Fixed-up Trackball and Fixed-right Trackball with increased rotation amount

Table 5-3 Accuracy rates per task (in percentages) for the Fixed-up Trackball and the Fixed-right Trackball with increased rotation amount

Rotation Techniques / Tasks	**Fixed-up trackball**	**Fixed-right trackball w/ increased rotation amount**
Find one emoticon	100%	100%
Count all emoticons	54%	46%
Estimate distances	75%	87%

5.1.4 Device- interaction styles

The two device-interaction styles evaluated in this first experiment are compared based on their average task completion time and average number of mistakes per task for all three rotation techniques.

5.1.4.1 Find one emoticon

For the task of finding one emoticon (Figure 5.7), a two sample assuming unequal variances t-test showed no significant differences between the average completion time of the two-handed/landscape with finger interaction style and the two-handed/landscape with thumb interaction style using the Fixed-right Trackball ($p=0.1675> 0.05$) and Fixed-up Trackball ($p=0.1077 > 0.05$). However, a significant difference was discovered for the Fixed-right Trackball with increased rotation amount ($p=0.0290 < 0.05$).

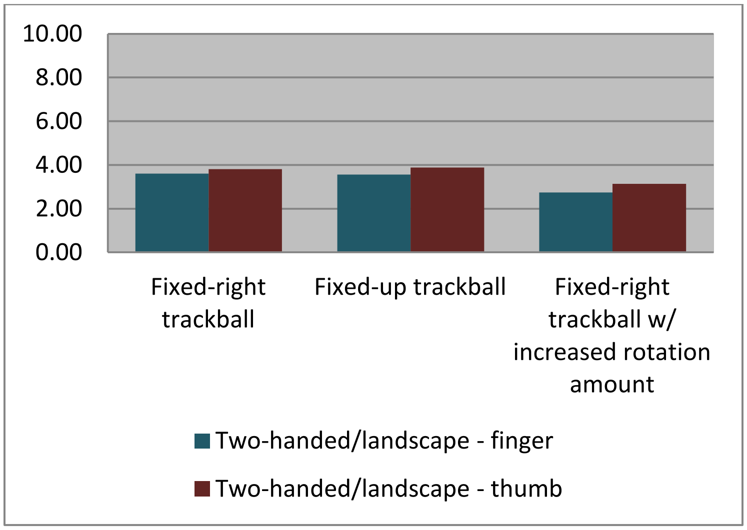

Figure 5.7 Average completion time for finding one emoticon using two-handed/landscape interaction for all rotation techniques

5.1.4.2 Count all emoticons

For counting all emoticons (Figure 5.8), a two-sample assuming unequal variances t-test found no significant differences between the average completion times of the two-handed/landscape with finger and two-handed/landscape with thumb using the Fixed-up Trackball, ($p=0.1720 > 0.05$) and Fixed-right Trackball with increased rotation amount, ($p=0.1311 > 0.05$). However, a significant difference was noticed for the Fixed-right Trackball with increased rotation amount, ($p=0.0414 < 0.05$). The average number of mistakes made using the two-handed/landscape with finger interaction style was higher than the number of mistakes made using the two-handed/landscape with thumb for the Fixed-right Trackball and Fixed-up Trackball but lower for the Fixed-right Trackball with increased rotation amount (Figure 5.9).

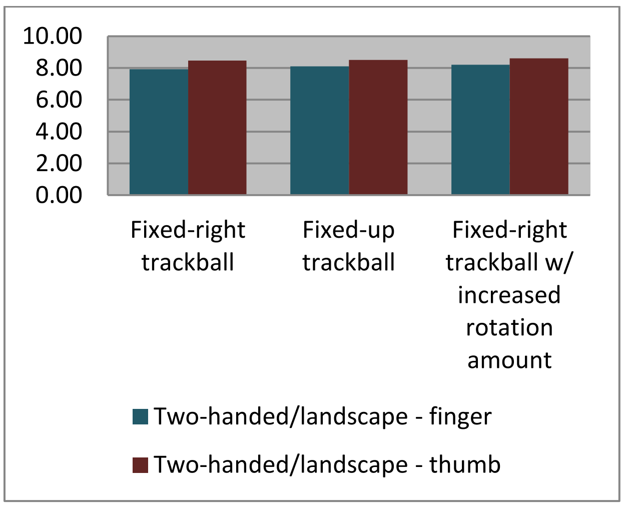

Figure 5.8 Average completion time for counting all emoticons using two-handed landscape interaction for all rotation techniques

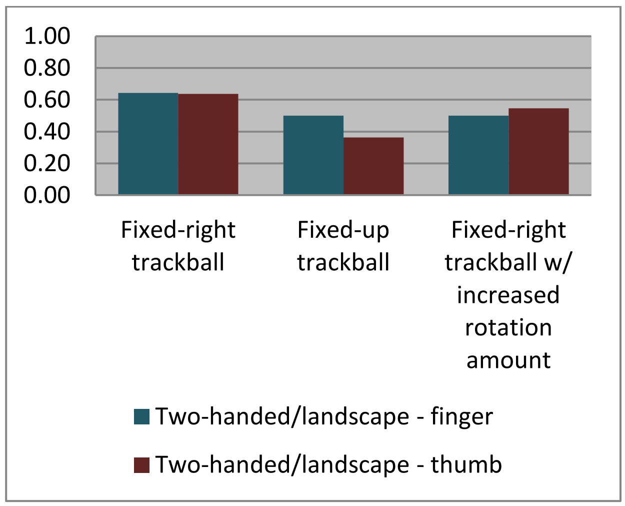

Figure 5.9 Average number of mistakes per participant for counting all emoticons using two-handed landscape interaction for all rotation techniques

5.1.4.3 Estimating distances

In terms of average completion time, for the task of estimating distances (Figure 5.10), there were no significant differences observed between the two interaction styles as indicated by a two-sample assuming unequal variances t-test, for the Fixed-right Trackball, ($p=0.8511 > 0.05$), Fixed-up Trackball, ($p=0.5552 > 0.05$) and Fixed-right Trackball with increased rotation amount, ($p=0.9183 > 0.05$). In general, the average number of mistakes made for estimating distances using the two-handed/landscape with thumb interaction style was lower than by using the two-handed/landscape with finger interaction style for all three rotation techniques (Figure 5.11).

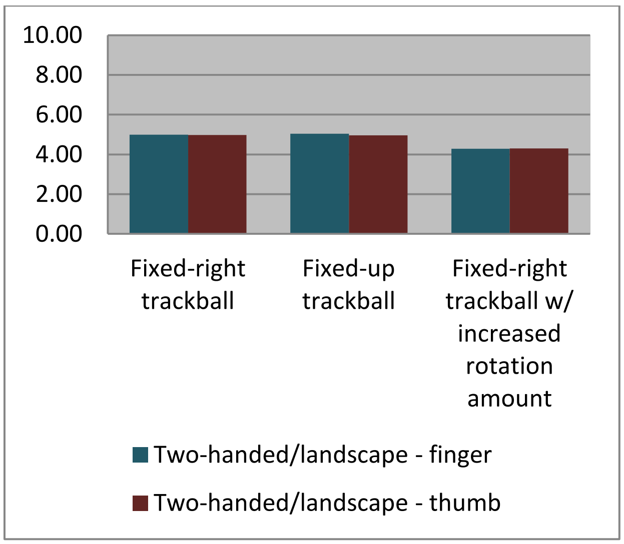

Figure 5.10 Average completion time for estimating distances using two-handed landscape interaction for all rotation techniques

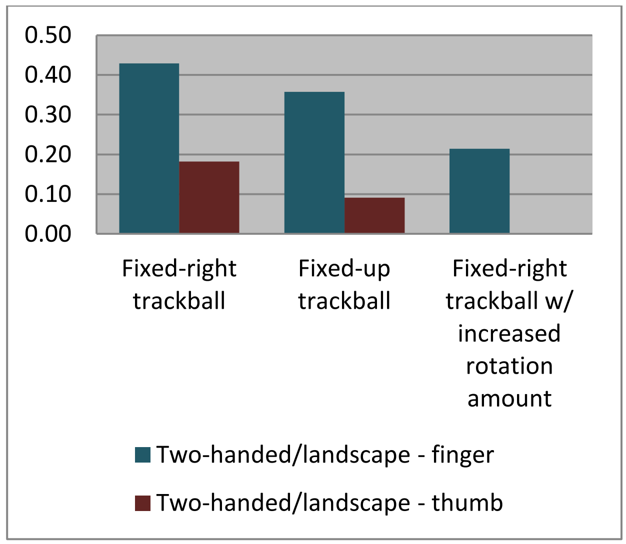

Figure 5.11 Average number of mistakes per participant for estimating distances using two-handed landscape interaction for all rotation techniques

5.1.5 Questionnaire

Participants rated the difficulty of each rotation technique for the three tasks using a five point rating scale from 1 to 5, with 1 being very hard, 2 hard, 3 moderate, 4 easy and 5 very easy (Figure 5.12). Participants also expressed their opinion on which of the three rotation techniques is more suitable and intuitive for rotating 3D objects and what are some of the additional options they think would improve 3D object interaction on mobile devices. In general, the Fixed-up Trackball and Fixed-right Trackball were considered to be the most difficult rotation techniques while the Fixed-right Trackball with increased rotation amount was considered to be the easiest. A paired two sample for means t-test revealed no significant difference between the average difficulty ratings of all rotation techniques for the three tasks. The only significant difference observed was between the Fixed-up Trackball and Fixed-right Trackball with increased rotation amount for the task of finding one emoticon ($p=0.004 < 0.05$). Participants considered the zoom, global rotation and flicking as the most important factors that might influence the overall usability for 3D object interaction on mobile devices (Figure 5.13).

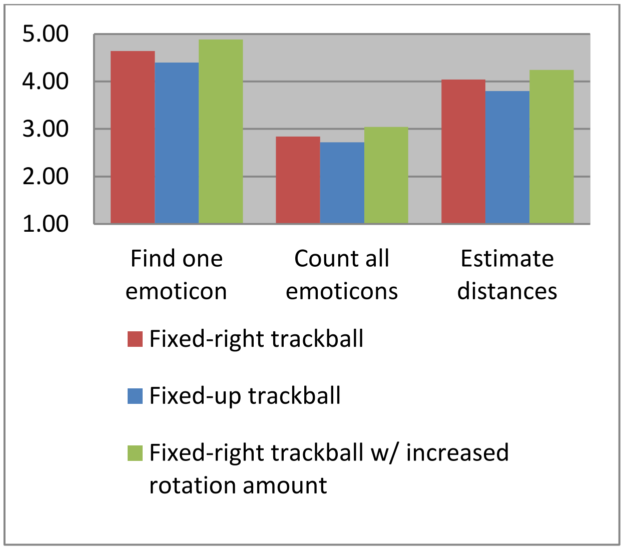

Figure 5.12 Average difficulty ratings per task for all rotation techniques

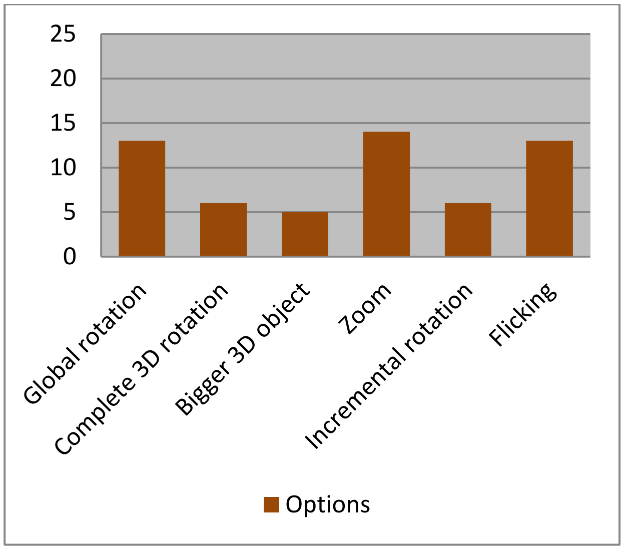

Figure 5.13 Influential options for 3D object interaction on mobile devices

5.2 Second experiment

In the second experiment of this project, two rotation techniques were investigated: the Fixed-up Trackball and the Two-axis Trackball, along with device interaction styles concerning one and two-handed interaction in landscape and portrait modes.

5.2.1 Fixed-up Trackball vs. Two-axis Trackball

The Two-axis Trackball proved to be faster (Figure 5.14) than the Fixed-up Trackball as a paired two sample for means t-test revealed significant differences between the average task completion times of the two rotation techniques for finding one emoticon ($p=2.56\times10^{-7} < 0.05$), counting all emoticons ($p=2.28\times10^{-9} < 0.05$), estimating distances ($p=0.0033 < 0.05$), counting particular emoticons based on color and type ($p=4.20\times10^{-9} < 0.05$), connecting emoticon pairs, ($p=1.35\times10^{-12} < 0.05$) and counting particular emoticons by type ($p=1.21\times10^{-8} < 0.05$).

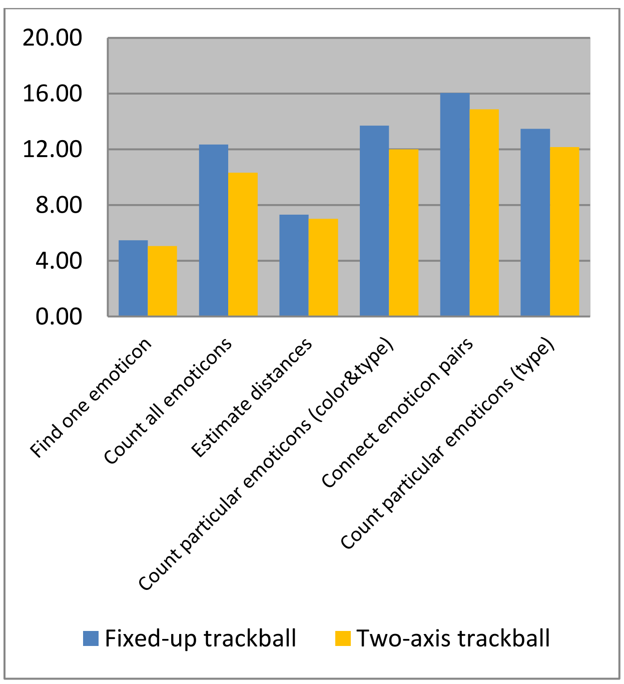

Figure 5.14 Average task completion time (in seconds) for the Fixed-up Trackball and Two-axis Trackball

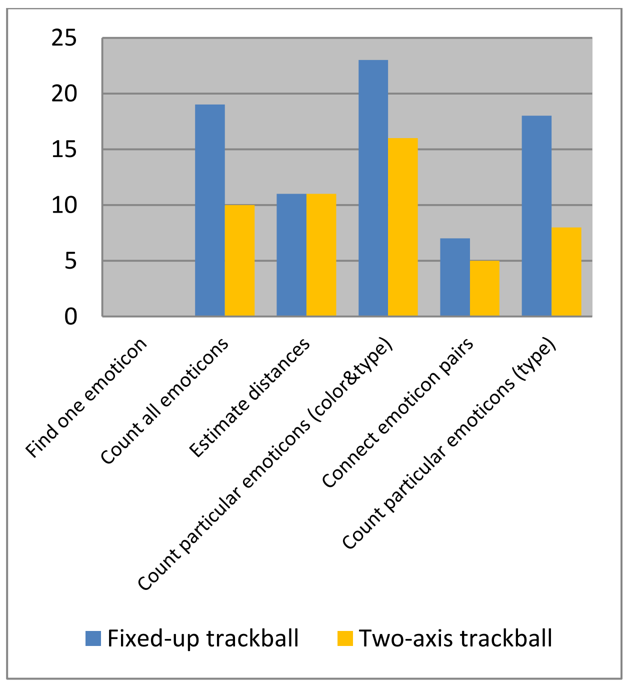

Figure 5.15 Total number of mistakes per task for the Fixed-up Trackball and Two-axis Trackball

The total number of mistakes made with the Two-axis Trackball was lower than the total number of mistakes made with the Fixed-up Trackball for all tasks with the exception of estimating distances which had an equal number of mistakes (Figure 5.15). The accuracy rates indicate that, overall, the Two-axis Trackball was the most efficient rotation technique (Table 5-4).

Table 5-4 Accuracy rates per task (in percentages) for the Fixed-up Trackball and the Two-axis Trackball

Rotation techniques / Tasks	**Fixed-up Trackball**	**Two-axis Trackball**
Find one emoticon	100%	100%
Count all emoticons	24%	60%
Estimate distances	56%	56%
Count particular emoticons (color & type)	8%	36%
Connect emoticon pairs	72%	80%
Count particular emoticons (type)	28%	68%

5.2.2 Device-interaction styles

In addition to the two-handed/landscape with thumb and the two-handed/landscape with finger interaction styles that were part of the first experiment, two more device-interaction styles were added in the second experiment, namely the two-handed/portrait with finger and the one-handed/portrait with thumb which leads to a total of four interaction styles. The results from the task of finding one emoticon were not included in the data analysis since the task was used to get users accustomed to the landscape and portrait orientation modes.

5.2.2.1 Count all emoticons

For the task of counting all emoticons, a two-sample assuming unequal variances t-test showed no significant differences between the average completion times (Figure 5.16) of all interaction styles. However, in terms of average number of mistakes, the two-handed/landscape with thumb was the most accurate interaction style for the Fixed-up Trackball while the two-handed/landscape with finger was the most accurate interaction style for the Two-axis Trackball (Figure 5.17).

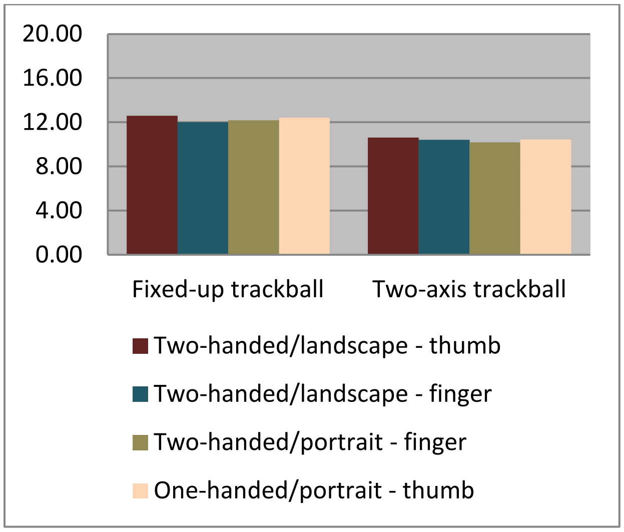

Figure 5.16 Average completion times per device interaction style with both rotation techniques for counting all emoticons

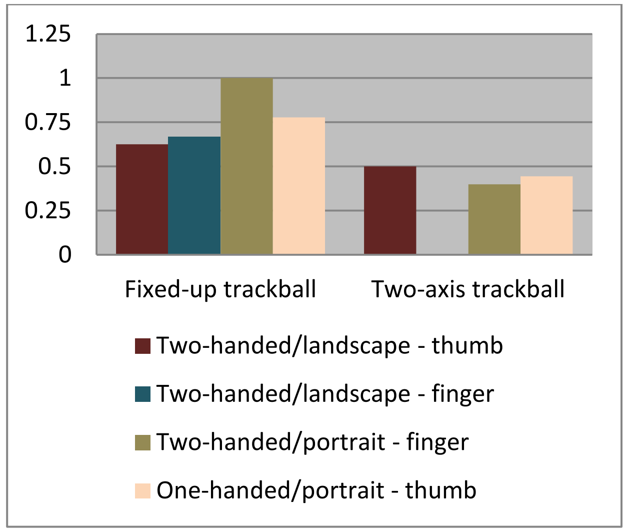

Figure 5.17 Average number of mistakes per device interaction style with both rotation techniques for counting all emoticons

5.2.2.2 Estimate distances

For estimating distances, no significant differences in average completion times were found to confirm the effectiveness of a particular device interaction style over the others (Figure 5.18). The two-handed/portrait with finger proved to be more accurate for the Fixed-up Trackball and the one-handed/portrait with thumb was more accurate for the Two-axis Trackball (Figure 5.19).

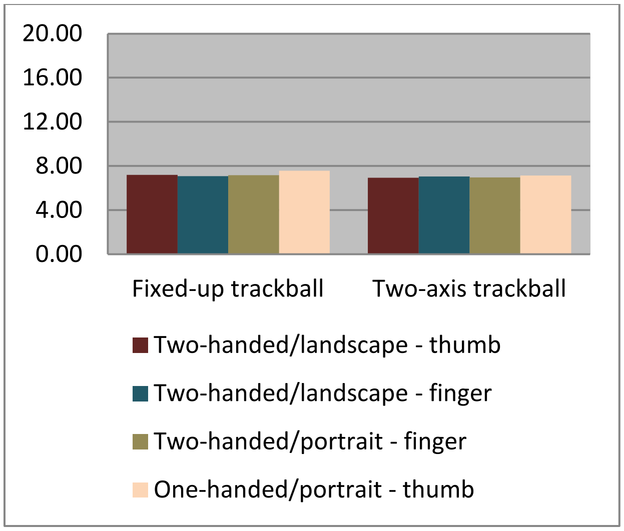

Figure 5.18 Average completion times per device interaction style with both rotation techniques for estimating distances

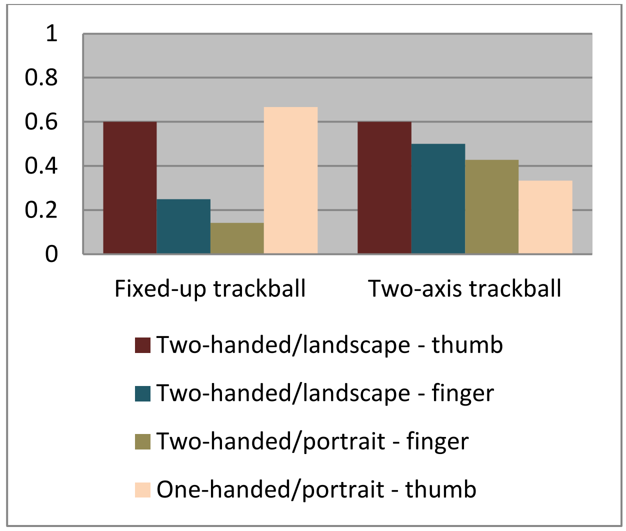

Figure 5.19 Average number of mistakes per device interaction style with both rotation techniques for estimating distances

5.2.2.3 Count particular emoticons by color and type

For the task of counting particular emoticons by color and type, there was no evidence to support a significant difference in terms of average completion time between the different interaction styles (Figure 5.20). The one-handed/portrait with thumb and the two-handed/portrait with thumb were the most accurate interaction styles for the Fixed-up Trackball and Two-axis Trackball respectively (Figure 5.21).

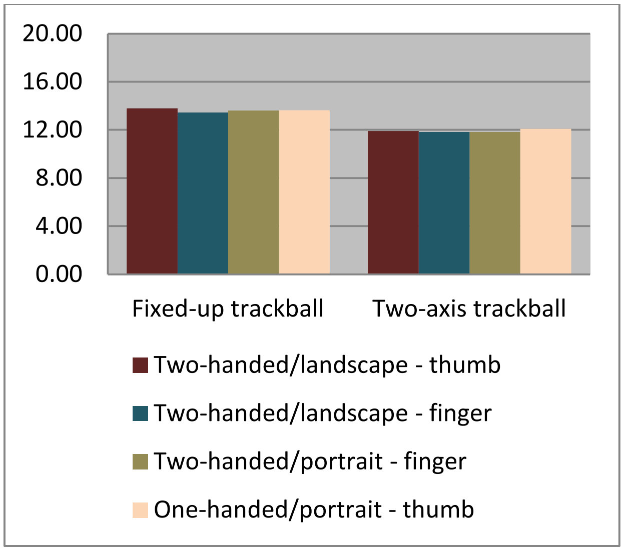

Figure 5.20 Average completion times per device interaction style with both rotation techniques for counting particular emoticons based on color and type

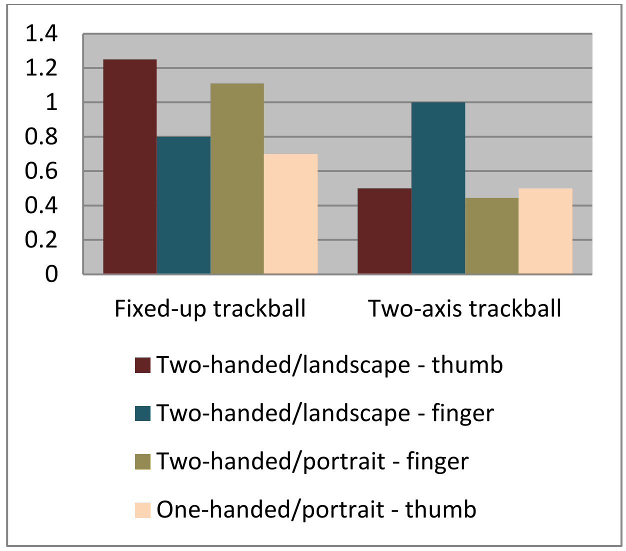

Figure 5.21 Average number of mistakes per device interaction style with both rotation techniques for counting particular emoticons based on color and type

5.2.2.4 Connect emoticon pairs

For connecting emoticon pairs (Figure 5.22), a significant difference was observed between the two-handed/landscape with finger and the one-handed/portrait with thumb ($p=0.0096 < 0.05$) for the Fixed-up Trackball but not between other interaction styles. The average number of mistakes shown in Figure 5.23 indicate that, for both rotation techniques, the two-handed/landscape with finger was the most accurate interaction style.

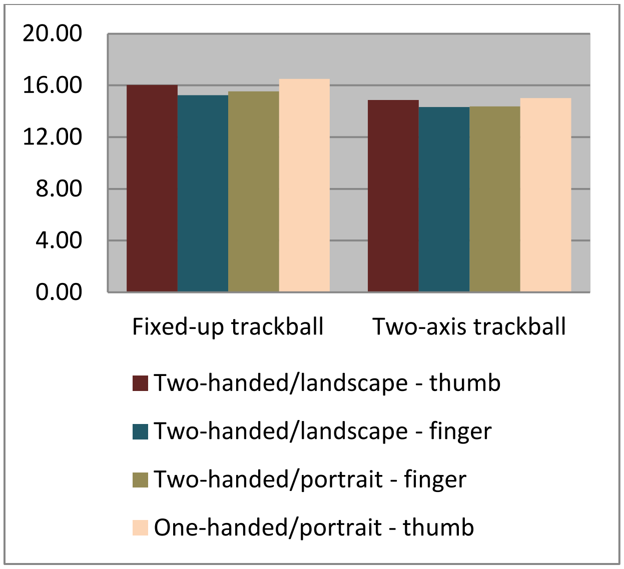

Figure 5.22 Average completion times per device interaction style with both rotation techniques for connecting emoticon pairs

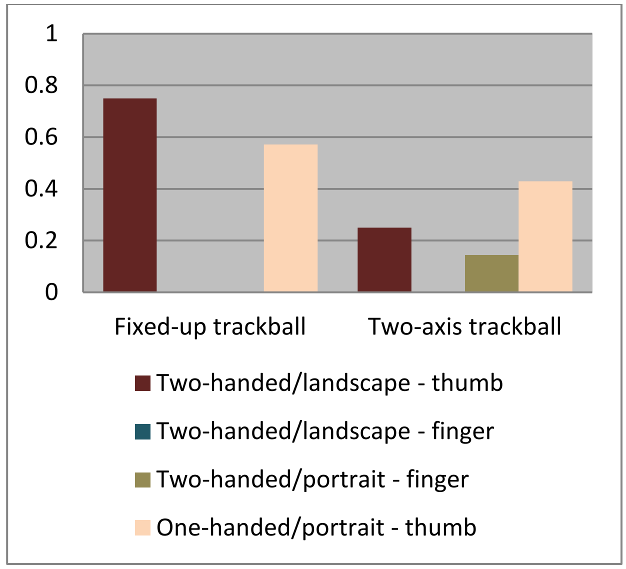

Figure 5.23 Average number of mistakes per device interaction style with both rotation techniques for connecting emoticon pairs

5.2.2.5 Count particular emoticons by type

The two-handed/portrait with finger interaction style was significantly faster than the one-handed portrait with thumb ($p=0.0044 < 0.05$) for the task of counting particular emoticons by type using the Fixed-up Trackball rotation technique (Figure 5.24). Overall, the two-handed/portrait with finger was the most accurate interaction style (Figure 5.25).

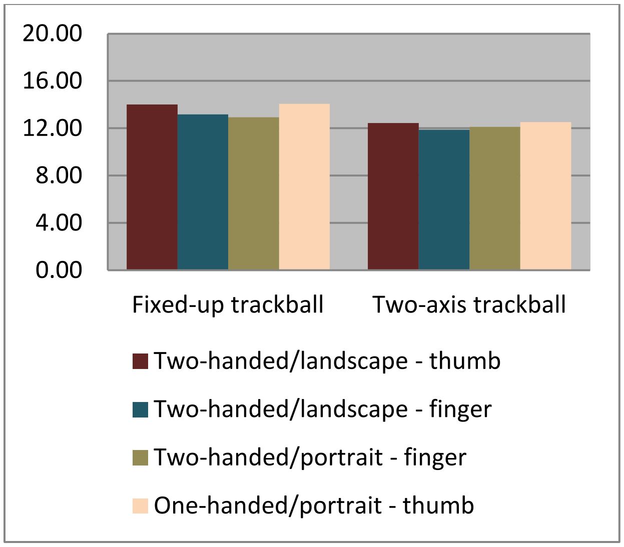

Figure 5.24 Average completion times per device interaction style with both rotation techniques for counting particular emoticons based on type

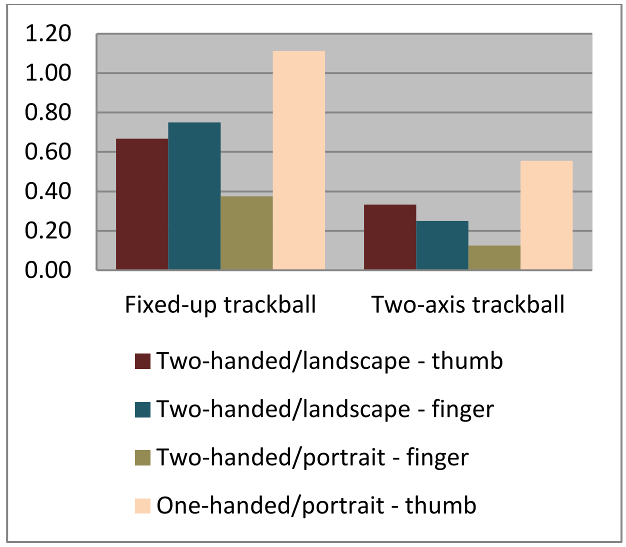

Figure 5.25 Average number of mistakes per device interaction style with both rotation techniques for counting particular emoticons based on type

5.2.4 Questionnaire

A five point rating scale was used to evaluate the difficulty of rotation techniques per task as perceived by participants. The five point rating scale registers the following values: 1- very hard, 2 – hard, 3 – moderate, 4 – easy, 5 – very easy. Figure 5.26 shows that, in general, the Fixed-up Trackball was perceived as being more difficult than the Two-axis Trackball. A paired two sample for means t-test revealed no significant differences, in terms of average difficulty ratings, between the tasks of finding one emoticon ($p=0.25958 > 0.05$), connecting emoticon pairs ($p=0.5425 > 0.05$) and counting particular emoticons by type ($p=0.1058 > 0.05$) but found significant differences between the tasks of counting all emoticons ($p=0.0384 < 0.05$), estimating distances ($p=0.0449 < 0.05$) and counting particular emoticons by color and type ($p=0.0081 < 0.05$).

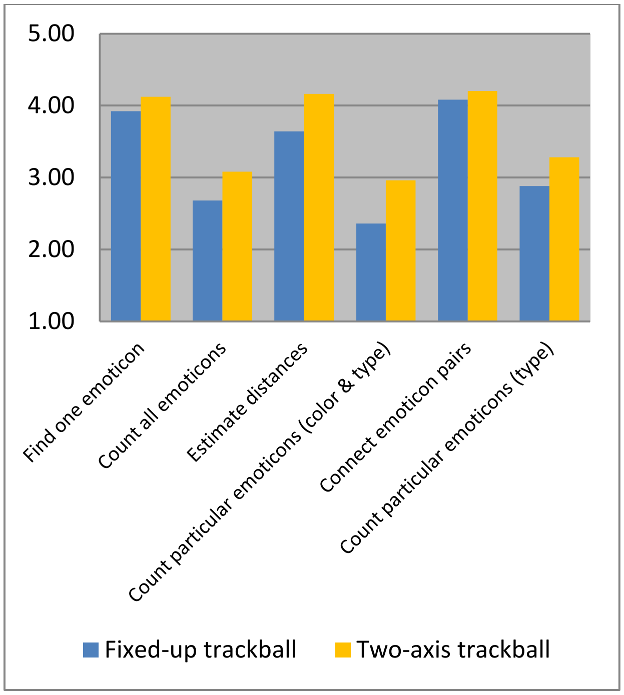

Figure 5.26 Average difficulty ratings per task for all rotation techniques

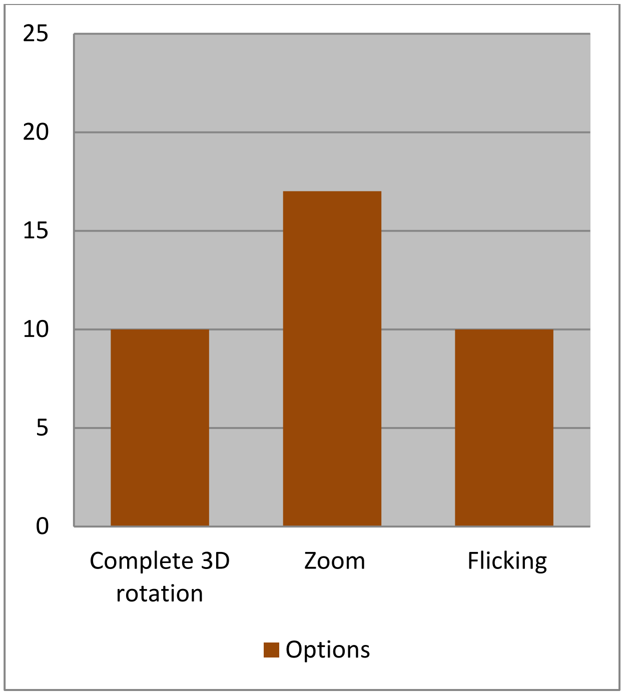

Figure 5.27 Influential options for 3D object interaction on mobile devices

Most participants considered the Two-axis Trackball to be a better rotation technique than the Fixed-up Trackball (Figure 5.28). In portrait mode, participants preferred using the one-handed/portrait with thumb interaction style while in landscape mode, they preferred the two-handed/landscape with finger.

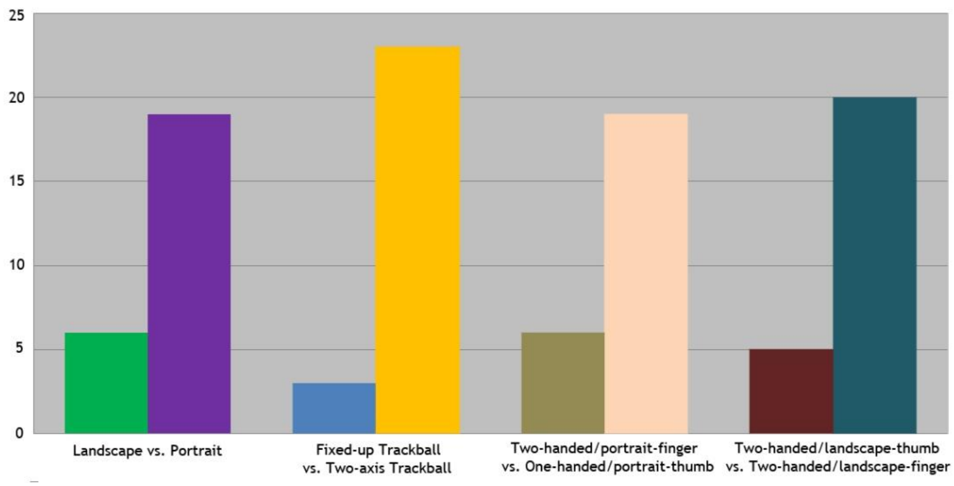

Figure 5.28 User preference for different rotation techniques and device interaction styles

Chapter 6 - Discussion

This chapter discusses the results of the project and presents the relationships between the observed facts and the research objectives. The discussion is focused on the rotation techniques and the device interaction styles that were part of the experimental comparison.

6.1 Rotation techniques

6.1.1 Fixed-right trackball vs. Fixed-up trackball

While the difference in average task completion time between the Fixed-right Trackball and Fixed-up Trackball for all three tasks of the first experiment is minimal and not statistically significant, participants struggled more with the Fixed-right Trackball, as evidenced by the higher number of mistakes made per task and the lower accuracy rates. This difference in accuracy regarding mistakes might be caused by the different rotation mappings which influence the steps following the initial starting rotation and subsequently the entire rotation process. Participants reported that they are aware of the difference between the underlying mapping controls for the Fixed-right Trackball and Fixed-up Trackball but that it was very difficult to accurately identify which mapping control is more preferable or effective. This uncertainty is further confirmed by the lack of significant differences between the average difficulty ratings of these two techniques. Also, when asked to specify which of these rotation techniques is more intuitive and useful for 3D interaction, seven participants chose the Fixed-right Trackball and another seven participants chose the Fixed-up Trackball. Moreover, the similarity in average completion times between the two techniques seems to suggest that participants can adapt effortlessly to the rotation mappings and that accuracy rates are not a major factor in establishing which of these two rotation techniques is more suited for rotation of 3D objects on mobile devices.

In summary, the outcome of the experiment demonstrates that the Fixed-right Trackball performs equally well as the Fixed-up Trackball. This confirms the apparent lack of discrimination regarding 3D mobile and desktop applications that utilize both these techniques mostly based on designers' choice. However, an interesting observation is that 3D modeling and visualization applications for the desktop platform use the Fixed-up Trackball technique while the same applications for the mobile platform use the Fixed-right Trackball. Also, most research studies that deal with rotation techniques based on the virtual trackball metaphor deal with the Fixed-up Trackball and do not consider the Fixed-right Trackball. This might be due to the smaller and less known 3D desktop applications that employ the Fixed-right Trackball compared to the wide spread commercial applications which utilize the Fixed-up Trackball.

To accurately distinguish between the Fixed-right Trackball and the Fixed-up Trackball technique, orientation tasks might be more suited. Although these types of tasks have less practical value than rotation tasks they can reveal potential weaknesses in the overall rotation process.

6.1.2 Fixed-right Trackball vs. Fixed-right Trackball w/ increased rotation amount

The Fixed-right Trackball proved to be significantly slower than the Fixed-right Trackball with increased rotation amount for identification and distance estimation tasks where a considerable difference in average completion time was registered between the two techniques. However, in the case of counting tasks, no significant differences were found between the two rotation techniques. Overall, the Fixed-right Trackball with increased rotation amount was more efficient, participants making fewer mistakes with this technique. It appears that the added rotation amount helps in speeding up the rotation process and it also makes it easier to quickly correct mistakes on the fly and the results seem to confirm the significance of these claims. Nevertheless, in some cases, the added rotation amount is disruptive and less effective than without the added rotation amount as was the case with the counting task.

Participants rated the Fixed-right Trackball with increased rotation amount to be less difficult than the Fixed-right Trackball and also chose it to be the most intuitive and suitable option for 3D interaction. However, participants did mention that dealing with an increased rotation amount requires more mental resources and is more challenging than the default rotation amount of the other techniques. Given the high level of expertise and the familiarity with 3D rotation on mobile devices of the participants from the first experiment, it would appear normal to expect such an improved performance with the Fixed-right Trackball with increased rotation amount. In the case of less experienced users, the considerable differences between using the default rotation amount and the increased rotation amount could be less impressive. Also, the rotation tasks of the first experiment might be more suited for this type of increased rotation amount than, for example, orientation tasks which might prove to be harder to complete given the precise rotation strokes needed. Therefore, an appropriate design consideration would be to integrate an additional rotation amount parameter that can be manipulated while rotating the 3D object such as flicking.

6.1.3 Fixed-up Trackball vs. Fixed-right Trackball w/ increased rotation amount

In general, the Fixed-up Trackball performed worse than the Fixed-right Trackball with increased rotation amount, in terms of average completion time and accuracy rates, for finding one emoticon and estimating distances. Better results were obtained with the Fixed-up Trackball for counting all emoticons. Simple tasks such as finding one emoticon and estimating distances benefit from the increased rotation amount because of the need for bigger rotation strokes. In contrast, this added rotation amount is less effective for counting tasks where small and precise rotation strokes are required. These findings further confirm that the Fixed-right Trackball with increased rotation amount is not suitable for precise rotation. The effectiveness of this technique might reach a bigger potential on mobile devices with smaller touchscreens, where a single continuous rotation stroke can eliminate the need for several smaller rotation strokes. Since the Fixed-right Trackball with increased rotation amount is more demanding, in terms of mental resources, than the Fixed-up Trackball, it should not be used for complex tasks that require considerable time to complete. Also, there is the possibility that important information about a

3D object can be missed since in a real-life situation users want to visualize and interact with a 3D object to observe the structure and also the small details.

6.1.4 Fixed-up Trackball vs. Two-axis Trackball

Generally, the Two-axis Trackball technique proved to be significantly faster, in terms of average completion time, and more efficient, in terms of accuracy rates, than the Fixed-up Trackball technique. Participants managed to complete all tasks in a shorter period of time and with fewer mistakes (the only exception being the estimating distances task) with the Two-axis Trackball. While the Fixed-up Trackball generates rotations that do not always match finger movements, the Two-axis Trackball has a very predictable behavior and all finger movements match the intended rotations. However, participants complained about the lack of transitive rotations regarding the Two-axis Trackball which led to a loss in perceived precision since it is very hard to reposition the 3D object into its original position once it has been rotated. Moreover, due to the inability to control rotation about the look-vector, the exact angles of rotations about the up and right vectors are hard to specify. This unnatural motion gives the appearance that a more complex 3D object such as a human head is rotated to the side instead of an upright position. Nevertheless, even with all the disadvantages presented above, the Two-axis Trackball, which behaves as expected, outperforms the Fixed-up Trackball. The results are similar to the ones obtained by [Bade, Ritter, & Preim, 2005] where the Two-axis Trackball performed significantly faster than the Fixed-up Trackball. However, the same study did not show any differences between accuracy rates.

6.2 Device interaction styles

6.2.1 Two-handed/landscape with thumb

The results of the first experiment suggest that, in terms of average completion time, the two-handed/landscape with thumb performs worse than the two-handed/landscape with finger, although the differences are minimal. However, in terms of accuracy rates, the two-handed/landscape with thumb outperforms the two-handed/landscape with finger. These findings can be attributed to the anatomy of the thumb and the more difficult learning curve of the two-handed/landscape with thumb interaction style. The limited range of motion of the thumb makes it impossible for long continuous rotation strokes to be performed while, by using the finger, participants can utilize the entire display screen to input finger movements. The limitation of the two-handed/landscape with thumb is further exacerbated by the constant switching between the use of the left and right thumb as participants seem to fatigue when using one thumb. A consequence of this limited range of motion is that participants can observe the 3D object in more detail since only small motions can be made with the thumb.

The results of the second experiment show that the two-handed/landscape with thumb is slower in comparison to the two-handed/landscape with finger and the two-handed/portrait with finger but it is faster or in some cases as slow as the one-handed/portrait with thumb which seems to confirm the above statement regarding the limited thumb range of motion. Overall, the average numbers of mistakes per participant made with the two-handed/landscape with thumb are similar to the other interaction styles with some exceptions. It appears that for counting tasks, the two-handed/landscape is more accurate than the two-handed/landscape with finger but less accurate than the two-handed/portrait with finger. Despite its good results, only a small number of participants consider the two-handed/landscape with thumb to be an intuitive and natural interaction style. In summary, this interaction style is more suited for rotation tasks where small rotation strokes are needed.

6.2.2 Two-handed/landscape with finger

For the first experiment, the two-handed/landscape with finger was better in terms of average completion time when compared to the two-handed/landscape with thumb. The main reason for this difference is the larger range of motion available to the finger. Participants were easily able to reach all parts of the display screen without having to severely occlude the screen or to "overreach" as is the case with the thumb. However, the average number of mistakes made with the two-handed/landscape with finger is higher than the average number of mistakes made with the two-handed/landscape with thumb. This difference in accuracy is explained by the fact that, participants tend to pay less attention to the information displayed on the 3D object as the finger movements become easier. The results from the second experiment indicate that, in general, the two-handed/landscape with finger outperforms the two-handed/landscape with thumb and the one-handed/portrait with thumb and is on the same level with the two-handed/portrait with finger. A wide majority of participants consider the two-handed/landscape with finger to be a very intuitive and natural interaction style.

6.2.3 Two-handed/portrait with finger

Overall, the two-handed/portrait with finger interaction style is the most appropriate device interaction style being faster and more accurate than the two-handed/landscape with thumb and the one-handed/portrait with thumb and edging the two-handed/landscape with finger. The main reason for this performance is that the two-handed/portrait with finger allows users to firmly grasp the smartphone with one hand and smoothly perform different types of rotation strokes with the finger from the other hand. Surprisingly, only a small number of participants consider this interaction style to be intuitive and natural. Participants felt uncomfortable employing a two-handed approach in portrait mode and were more compelled to only use a single hand and the thumb from the same hand.

6.2.4 One-handed/portrait with thumb

Although a wide majority of participants considered the one-handed/portrait with thumb interaction style to be the most intuitive and natural device interaction style, it has the worst task completion times of all the interaction styles. While for the two-handed/landscape with thumb participants could stabilize the mobile device with two hands and use two thumbs in an alternating style to perform rotation strokes, for the one-handed/portrait with thumb, the use of a single hand coupled with the limited range of motion of the thumb makes it very difficult to stabilize the mobile device and perform rotation strokes in a fast manner. Moreover, the rotation strokes performed with the thumb are usually very short which leads to a further decrease in speed. However, similar to the two-handed/landscape with thumb, the one-handed/portrait with thumb has good accuracy rate compared with the device interaction styles that employ the finger for input. Participants enjoyed this interaction style because it mimics the normal way of interaction with a mobile device for purposes such as typing, calling, taking pictures, etc.

Chapter 7 - Conclusion and future work

This project offers a comprehensive view on adapting rotation techniques to mobile device constraints based on touchscreen gestures as well as the multitude of device-interaction styles that can be employed to further enhance user involvement for 3D object interaction. These rotation techniques and interaction styles were evaluated based on design principles and an experimental comparison. The results indicate that 3D object interaction on mobile devices can greatly benefit from the use of appropriate rotation techniques.

The rapid development of computing technology for high-end mobile devices such as tablets and smartphones has paved the way for 3D mobile applications with impressive graphics and visual effects. However, appropriate interaction techniques that enhance the performance and user satisfaction of these applications have been in part neglected. Interaction on mobile devices has two main purposes, namely device interaction, which refers to the overall controls and input methods of a mobile device, and application interaction, which deals with the representation of virtual applications on the mobile device. While newer methods of input have been developed as a result of inertial sensors, touchscreen based input remains the standard. In 3D object interaction, users perform transformation actions on a 3D object by ways of translation, scaling and most importantly rotation. Therefore, this project focused on rotation as the main interaction task.

Established rotation techniques that originate from the desktop platform are difficult to adapt on mobile devices because of constraints such as thumb occlusion, small screen size, etc., and specific functionalities such as direct manipulation and touchscreen gestures. These factors must be taken into account when considering potential solutions. The rotation techniques based on the virtual trackball metaphor are best suited for 3D object interaction on mobile devices since they do not require any graphical user interface or specialized viewport, and because the entire display screen of the mobile device can be used as input. The most important virtual trackball techniques were first reviewed to accurately determine their potential usefulness given the mobile device restrictions and functionalities and how they solve the 2D to 3D mapping problem. The remaining virtual trackball techniques were then evaluated based on several design principles, derived from related studies, for achieving pleasing and predictable rotation effects. The final rotation techniques considered appropriate for 3D object interaction on mobile devices were evaluated based on an experimental comparison.

Overall, the Two-axis Trackball is the most appropriate rotation technique for 3D object interaction on mobile devices. The ability to generate predictable rotations makes it ideal for rotating completely axisymmetrical objects such as cube and sphere. However, for orientation tasks and 3D editing and modeling actions, the Two-axis Trackball will probably not perform as well as the Fixed-up Trackball or Fixed-right Trackball due to the lack of transitive rotations. These findings are in line with those of [Bade, Ritter, & Preim, 2005] and [Decle & Hachet, 2009], who found that the Two-axis Trackball is the most appropriate rotation technique.

The Fixed-up and Fixed-right Trackballs have the same performance and preference results, regardless of their local and world vectors control mappings. The ability to generate transitive rotations makes them very valuable for complex 3D modeling applications where precision and exact alignment are required. Furthermore, structurally complex 3D objects such as human head, vase, etc., can easily be rotated with the Fixed-up and Fixed-right Trackballs because users can guide themselves using the overall structure of the object. This is not the case however, for completely axisymmetrical objects such as sphere and cube, with no orientation marks. By increasing the rotation amount of the Fixed-right Trackball, users were able to perform better, in terms of completion time and accuracy rates for simple tasks. The added rotation amount enhances the overall rotation process as well as mentally engaging users. The downside of this is that, for more complex tasks the added rotation amount can be disruptive and less effective and users fatigue more.

The two-handed/portrait with finger is the most appropriate device interaction style both in terms of accuracy and completion time. Unlike the two-handed/landscape with finger, which also had good performance, it allows users to firmly grasp the mobile device while smoothly performing rotation strokes. Interestingly, only a handful of participants felt comfortable using the two-handed/portrait with finger as it is not an intuitive and natural way of interacting with the mobile device. The two-handed/landscape with thumb and the one-handed/portrait with thumb are the slowest interaction styles. This is due to the limited range of motion of the thumb which prevents users from making long continuous rotation strokes. The one-handed/portrait with thumb was considered the most intuitive style of interaction. Participants enjoyed this interaction style because it emulates the normal way of interaction with a mobile device for purposes such as typing, calling, taking pictures, etc. This confirms to an extent the findings in related literature that state that two-handed interaction is better than one-handed interaction but is less preferred because of the amount of mental, visual and time resources necessary.

The results of this project can be used to design appropriate 3D object interaction techniques for use in 3D mobile applications. Potential uses for these rotation techniques include examining a 3D object in a museum exhibition and virtual catalogue, studying 3D objects for scientific visualization and games that require users to inspect 3D objects.

Unfortunately, this project only considers the smartphone as a testing device and does not take into account bigger mobile devices such as tablets. A continuation to this study should include testing the rotation techniques on a tablet. Given the larger display screen of tablets (usually twice the size of a smartphone), rotation techniques that have not been considered for this experimental comparison, such as the Virtual Sphere, Shoemake's Arcball and Hanson's Rolling Ball, could also be part of the testing. As indicated by the results, increasing the rotation amount of a technique improves completion time and accuracy for simple tasks. A future study could include an automation of this rotation amount by using flicking (rapidly sliding a finger on the touchscreen). Flicking is a popular method in mobile applications for quickly browsing between different options but little is known of how this factor could influence rotation for 3D object interaction.

The major unsolved problem of all virtual trackball techniques is the mapping of controls from the two dimensional degrees of freedom of the input device (mouse, finger gestures) to the three dimensional rotation of the object. Multi-touch gestures, where users can employ one finger for controlling two axes of rotation and another finger for the third axis, represent a potential solution to this problem. Mobile devices with larger screen sizes such as tablets are the most appropriate devices for this approach. It would be interesting to see how much the occlusion generated by the use of multiple fingers would influence the overall performance and user satisfaction. Is the performance related to the size of the display screen and if so, to what degree?

Different scenarios for mobile device interaction might yield different results. Since users will most likely utilize mobile devices in different mobility conditions such as seated, slow walking and normal walking, it makes sense to test them accordingly. Moreover, there is the question of how demanding these mobility conditions are on users as well as what are the relationships between mobility conditions and device-interaction styles. For example, what device interaction styles are employed by users in a seated mobility condition and what are the resulting performances?

3D object interaction deals not only with rotation of 3D objects but also with other transformation effects such as translation and scaling. This study does not take translation and scaling into consideration and the question of how users would perform all these interaction tasks in conjunction to reach a goal is still unsolved. In addition, what are the influences of these interaction tasks on 3D objects that do not have a fixed pivot point? To accommodate for the rotation, translation and scaling transformations and the pivot point location, 3D modeling systems have employed widgets [Cohé, Decle, & Hachet, 2011], [Wu & Malheiros, 2002]. These widgets are visual cues which help users distinghuish the different tranformation states that 3D objects present. What are the implications of widgets regarding rotation on mobile devices?

In addition to the rotation techniques mentioned in this project, there are other interesting research directions for controlling 3D object rotation on mobile devices. Double-sided multi-touch input [Shen, Tsai, Chu, Hsu, & Chen, 2009] is based on a mobile device that receives simultaneous multi-touch input from both the front and the back of the device, enabling intuitive finger gestures for manipulating 3D objects. [Gallo, Minutolo, & De Pietro, 2010] present a geometry based rotation technique, where the rotation direction changes according to the depth of the picked point of the geometry. These ideas represent a good starting point for alleviating some of the problems associated with 3D rotation on mobile devices.

References

Ancillotti. (2011, April 25). *http://ancillotti.hubpages.com*. Retrieved May 2012, from http://ancillotti.hubpages.com/hub/Understanding-the-hardware-architecture-of-smartphones

Autodesk. (2012). *Autodesk Software*. Retrieved June 2012, from Autodesk: http://usa.autodesk.com/products/

Autodesk. (2012). *Mobile Apps - Design, Engineering, Effects Apps - Autodesk*. Retrieved June 2012, from Autodesk - 3D Design and Engineering Software for Architecture, Manufacturing and Entertainment: http://usa.autodesk.com/adsk/servlet/pc/index?siteID=123112&id=16953811

Bade, R., Ritter, F., & Preim, B. (2005). Usability Comparison of Mouse-Based Interaction Techniques for Predictable 3D Rotation. *SG'05 Proceedings of the 5th International Conference on Smart Graphics* (pp. 138-150). Springer-Verlag.

Bell, G. (1988). *Bell's Trackball*. Retrieved April 2012, from http://www.opensource.apple.com/source/X11server/X11server-85/mesa/Mesa-7.2/progs/util/trackball.c?txt

Buda, V. (2012). *Rotation Techniques for 3D Object Interaction on Smartphones.* Department of Information and Computing Sciences. Utrecht University.

Card, S., & Moran, T. (1986). User Technology - From Pointing to Pondering. *HPW '86 Proceedings of the ACM Conference on the History of Personal Workstations* (pp. 183-198). ACM.

Chen, M. (1991). *Patent No. 5,019,809.* United States of America.

Chen, M., Mountford, J. S., & Sellen, A. (1988). A Study in Interactive 3D Rotation Using 2D Control Devices. In R. J. Beach (Ed.), *SIGGRAPH '88 Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques. 22,* pp. 121-129. ACM.

Choi, M., & Kim, G. J. (2009). TouchBall: A Design and Evaluation of a Hand-held Trackball Based Touch-Haptic Interface. *CHI '09 Proceedings of the 27th International Conference on Human Factors in Computing Systems* (pp. 1535-1538). ACM.

Cohé, A., Decle, F., & Hachet, M. (2011). Cohe 2011 - tBox A 3D Transformation Widget designed for Touch-screens. *CHI '11 Proceedings of the 2011 annual conference on Human factors in computing systems* (pp. 3005-3008). ACM.

Cutler, K. M. (2012, June 27). *TechCrunch Android*. Retrieved July 30, 2012, from TechCrunch: http://techcrunch.com/2012/06/27/android-reaches-400-million-devices-activations-1-million-per-day/

Debarba, H., Franz, J., Reus, V., Maciel, A., & Nedel, L. (2011). The Cube of Doom: a Bimanual Perceptual User Experience. *2011 IEEE Symposium on 3D User Interfaces (3DUI)* (pp. 131-132). Singapore: IEEE Computer Society.

Decle, F., & Hachet, M. (2009). A Study of Direct Versus Planned 3D Camera Manipulation on Touch-Based Mobile Phones. *MobileHCI '09 Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services. Article no. 32.* ACM.

DNA Learning Center. (2012). *G2C Online 3-D Brain*. Retrieved June 2012, from DNA Learning Center: http://dnalc.org/resources/3dbrain.html

Döring, T., Shirazi, A. S., & Schmidt, A. (2010). Exploring Gesture-Based Interaction Techniques in Multi Display Environments with Mobile Phones and a Multi-Touch Table. *AVI '10 Proceedings of the International Conference on Advanced Visual Interfaces* (pp. 419-419). ACM.

Edge, D., & Blackwell, A. F. (2009). Bimanual Tangible Interaction with Mobile Phones. *TEI '09 Proceedings of the 3rd International Conference on Tangible and Embedded Interaction* (pp. 131-136). ACM.

Ericson, F., & Olwal, A. (2011). Interaction and Rendering Techniques for Handheld Phantograms. *CHI EA '11 Proceedings of the 2011 Annual Conference Extended Abstracts on Human Factors in Computing Systems* (pp. 1339-1344). ACM.

Evans, K. B., Tanner, P. P., & Wein, M. (1981). Tablet-based Valuators that Provide One, Two or Three Degrees of Freedom. *SIGGRAPH '81 Proceedings of the 8th Annual Conference on Computer Graphics and Interactive Techniques* (pp. 91-97). ACM.

Export2Reality;. (2012, May 28). *Real Time 3D Modern Interior with Red Ape Studio*. Retrieved June 20, 2012, from Export 2 Reality | 3D Design Studio: http://www.export2reality.co.uk/portfolio/real-time-3d-modern-interior-with-red-ape-studio/

Fabien, C. (2011). Design of a Bio Inspired 3D Orientation Coordinate System and Application. In S. Goto, *Robot Arms* (pp. 175-210). InTech.

Fingas, J. (2012, May 24). *Engadget Mobile*. Retrieved July 30, 2012, from Engadget: http://www.engadget.com/2012/05/24/idc-q1-2012-world-smartphone-share/

Froehlich, B., Hochstrate, J., Skuk, V., & Huckauf, A. (2006). The GlobeFish and The GlobeMouse - Two New Six Degree of Freedom Input Devices for Graphics Applications. *CHI '06 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 191-199). ACM.

Gallo, L., Minutolo, A., & De Pietro, G. (2010). A User Interface for VR-Ready 3D Medical Imaging by Off-The-Shelf Input Devices. *Computers in Biology and Medicine , 40* (3), 350-358.

Gordongdgd. (2012). Retrieved June 2012, from Google Play - 3D Rubik Cube: https://play.google.com/store/apps/details?id=my.wca.rubik.ext&hl=nl

Grootelaar, R. (2011). *Perspective Correction for 3D Graphics on Mobile Devices.* Utrecht University, Department of Information and Computing Sciences. Utrecht University.

GSM Arena. (2011, April). *GSM Samsung*. Retrieved March 10, 2012, from GSM Arena: http://www.gsmarena.com/samsung_i9100_galaxy_s_ii-3621.php

Guiard, Y. (1987). Asymmetric Division of Labor in Human Skilled Bimanual Action The Kinematic Chain as a Model. *Journal of Motor Behavior , 19* (4), 486-517.

Gurney, J. (2010). *Color and Light: A Guide for the Realist Painter.* Andrews McMeel Publishing.

Hancock, M., Cate, T. t., & Carpendale, S. (2009). Sticky Tools Full 6DOF Force-Based Interaction for Multi-Touch Tables. *ITS '09 Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (pp. 133-140). ACM.

Hanson, A. J. (1992). The Rolling Ball. In D. Kirk, *Graphics Gems III* (pp. 51-60). Academic Press Professional.

Helder, M. (2011). *VR Environments on Handheld Devices with Applications for Gaming.* Department of Information and Computing Sciences. Utrecht University.

Henriksen, K., Sporring, J., & Hornbæk, K. (2004). Virtual Trackballs Revisited. *IEEE Transactions on Visualization and Computer Graphics* , 206-216.

Hinckley, K., Pausch, R., Proffitt, D., & Kassell, N. F. (1998). Two-Handed Virtual Manipulation. *ACM Transactions on Computer-Human Interaction (TOCHI) , 5* (3), 260-302.

Hinckley, K., Tullio, J., Pausch, R., Proffitt, D., & Kassell, N. (1997). Usability Analysis of 3D Rotation Techniques. *UIST '97 Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology* (pp. 1-10). ACM.

Jacob, I., & Oliver, J. (1995). Evaluation of Techniques for Specifying 3D Rotations with a 2D Input Device. *HCI '95 Proceedings of the HCI'95 Conference on People and Computers X* (pp. 63-76). Cambridge University.

Karlson, A. K., & Bederson, B. B. (2007). ThumbSpace Generalized One-Handed Input for Touchscreen-Based Mobile Devices. *INTERACT'07 Proceedings of the 11th International Conference on Human-Computer Interaction* (pp. 324-338). Springer-Verlag.

Karlson, A. K., & Bederson, B. B. (2006). *Understanding Single-Handed Mobile Device Interaction.* Technical Report, Computer Science Department, University of Maryland.

Katzakis, N., & Hori, M. (2009). Mobile Phones as 3-DOF Controllers: A Comparative Study. *DASC '09 Proceedings of the 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing* (pp. 345-349). IEEE Computer Society.

Kettner, L. (1995). *A Classification Scheme of 3D Interaction Techniques.* Technical Report, Freie Universität Berlin, Department of Mathematics and Computer Science.

Kim, M.-S., Seong, J.-K., Hyun, D.-E., Lee, K.-H., & Choi, Y.-J. (2001). A Physical 3D Trackball. *PG '01 Proceedings of the 9th Pacific Conference on Computer Graphics and Applications* (pp. 134-139). IEEE.

Kirkham, J. (2010). *Inertial Sensors for Visualization Control.* University of Manchester, Faculty of Engineering and Physical Sciences. University of Manchester.

Kiwi Viewer. (2011, August 23). *Kiwi Viewer*. Retrieved June 2012, from Kiwi Viewer: http://www.kiwiviewer.org/

Kratz, S., & Rohs, M. (2010). Extending the Virtual Trackball Metaphor to Rear Touch Input. *2010 IEEE Symposium on 3D User Interfaces* (pp. 111-114). IEEE Computer Society.

Kratz, S., & Rohs, M. (2009). HoverFlow: Expanding the Design Space of Around-Device Interaction. *MobileHCI '09 Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services. Article no. 4.* ACM.

Kratz, S., Rohs, M., Guse, D., Müller, J., Bailly, G., & Nischt, M. (2012). PalmSpace: Continuous Around-Device Gestures vs. Multitouch for 3D Rotation Tasks on Mobile Devices. *AVI '12 Proceedings of the International Working Conference on Advanced Visual Interfaces* (pp. 181-188). ACM.

Lanier, L. (2008). *Advanced Maya Texturing and Lighting.* Sybex.

Ma, Z.-Q. (2007). *Group Theory for Physicists.* World Scientific Publishing Company.

Maniar, N., Bennett, E., Hand, S., & Allan, G. (2008). The Effect of Mobile Phone Screen Size on Video Based Learning. *Journal of Software* , 51-61.

MCW, S. S. (2012). *ShapeSpin Interactive Catalogue*. Retrieved June 20, 2012, from MWC Software Studios Solutions: http://www.mcwsoftware.com/products/shapeSpin

Mirtich, B. (1996). Fast and Accurate Computation of Polyhedral Mass Properties. *Journal of Graphics Tools , 1* (2), 31-50.

Miyaki, T., & Rekimoto, J. (2009). GraspZoom: Zooming and Scrolling Control Model for Single-Handed Mobile Interaction. *MobileHCI '09 Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services. Article No. 11.* ACM.

Neowin. (2012, May 26). *Guide to Smartphone Hardware: Displays*. (T. Schiesser, Editor) Retrieved March 11, 2012, from Neowin: http://www.neowin.net/news/guide-to-smartphone-hardware-47-displays

Nicolau, H., & Jorge, J. (2012). Touch Typing using Thumbs Understanding the Effect of Mobility and Hand Posture. *CHI '12 Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems* (pp. 2683-2686). ACM.

Otaduy, M. A., Igarashi, T., & LaViola, J. J. (2009). Interaction - Interfaces, Algorithms and Applications. *International Conference on Computer Graphics and Interactive Techniques. Article no. 14.* ACM.

Parhi, P., Karlson, A. K., & Bederson, B. B. (2006). Target Size Study for One-Handed Thumb Use on Small Touchscreen Devices. *MobileHCI '06 Proceedings of the 8th Conference on Human-Computer Interaction with Mobile Devices and Services* (pp. 203-210). ACM.

Partala, T. (1999). Controlling a Single 3D Object: Viewpoint Metaphors, Speed and Subjective Satisfaction. *Proceedings of INTERACT'99* (pp. 486-493). IOS Press.

Sasakura, M., Kotaki, A., & Inada, J. (2011). A 3D Molecular Visualization System with Mobile Devices. *IV '11 Proceedings of the 2011 15th International Conference on Information Visualisation* (pp. 429-433). London, UK: IEEE Computer Society.

Satyanarayanan, M. (2010). Mobile Computing: the Next Decade. *MCS '10 Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond. Article no. 5.* ACM.

Shen, E.-l. E., Tsai, S.-s. D., Chu, H.-h., Hsu, Y.-j. J., & Chen, C.-w. E. (2009). Double Side Multi-Touch Input for Mobile Devices. *CHI EA '09 Proceedings of the 27th International Conference Extended Abstracts on Human Factors in Computing Systems* (pp. 4339-4344). ACM.

Shoemake, K. (1992). Arcball - A User Interface for Specifying 3D Orientation Using a Mouse. *Proceedings of the Conference on Graphics Interface '92* (pp. 151-156). Morgan Kaufmann.

Shoemake, K. (1994). Arcball Rotation Control. In P. S. Heckbert, *Graphics Gems IV* (pp. 175-192). Academic Press Professional.

Thornton, R. W. (1979). The Number Wheel - A Tablet Based Valuator for Interactive Three-Dimensional Positioning. *SIGGRAPH '79 Proceedings of the 6th Annual Conference on Computer Graphics and Interactive Techniques* (pp. 102-107). ACM.

Wagner, J., Huot, S., & Mackay, W. (2012). BiTouch and BiPad - Designing Bimanual Interaction for Handheld Tablets. *CHI '12 Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems* (pp. 2317-2326). ACM.

Ware, C., & Rose, J. (1999). Rotating Virtual Objects with Real Handles. (J. Grudin, Ed.) *ACM Transactions on Computer-Human Interaction , 6* (2), 162-180.

Wu, S.-T., & Malheiros, M. G. (2002). Interactive 3D Geometric Modelers with 2D UI. *WSCG'02 Proceedings of International Conference in Computer Graphics, Visualization and Computer Vision* (pp. 559-559). ACM.

Zeleznik, R. C., Forsberg, A. S., & Strauss, P. S. (1997). Two Pointer Input For 3D Interaction. *I3D '97 Proceedings of the 1997 Symposium on Interactive 3D Graphics* (pp. 115-120). ACM.

Zhao, J. Y., Shuralyov, D., & Stuerzlinger, W. (2011). Comparison of Multiple 3D Rotation Methods. *2011 IEEE International Conference on Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS)* (pp. 1-5). IEEE Computer Society.

Appendix A - Questionnaire first experiment

Name:____

Age:____

Sex:____

Occupation:___________________________________

Dominant hand:_______________________

Questions asked before the experiment:

1. Do you have any vision/eye problems?

O… No
O… I use glasses
O… I use contacts
O… I am color blind

2. Do you own a smartphone?

O… Yes
O… No

3. How often do you play video games?

O… Never
O… Rarely
O… Regularly
O… Often

4. Do you play games on your mobile phone?

O… Never
O… Rarely
O… Regularly
O… Often

5. Have you used any games or apps on the mobile phone which included rotation of 3D objects?

O… Yes
O… No

Questions asked after the experiment:

6. Please rate the different parameters with respect to how difficult it was to perform each task:

Fixed-right Trackball					
Tasks	1 very hard	2 hard	3 moderate	4 easy	5 very easy
1) Find one emoticon	O	O	O	O	O
2) Count all emoticons	O	O	O	O	O
3) Estimate distances	O	O	O	O	O

Fixed-up Trackball					
Tasks	1 very hard	2 hard	3 moderate	4 easy	5 very easy
1) Find one emoticon	O	O	O	O	O
2) Count all emoticons	O	O	O	O	O
3) Estimate distances	O	O	O	O	O

Fixed-right Trackball with increased rotation amount					
Tasks	1 very hard	2 hard	3 moderate	4 easy	5 very easy
1) Find one emoticon	O	O	O	O	O
2) Count all emoticons	O	O	O	O	O
3) Estimate distances	O	O	O	O	O

7. Which one of these options seems to be more intuitive and suited for rotating 3D objects?

O… Fixed-right Trackball
O… Fixed-up Trackball
O… Fixed-right Trackball with increased rotation amount

8. What other options would you like to see for this application? Examples might include:

O… Global rotation
O… Complete 3D rotation
O… Bigger 3D objects
O… Zoom
O… Incremental rotation (45 degrees)
O… Flicking
O… Other (please mention them if you select this option)

Appendix B - Questionnaire second experiment

Name:____

Age:____

Sex:____

Occupation:_____________________________________

Dominant hand:________________________

Questions asked before the experiment:

1. Do you have any vision/eye problems?

O… No
O… I use glasses
O… I use contacts
O… I am color blind

2. Do you own a smartphone?

O… Yes
O… No

3. How often do you play video games?

O… Never
O… Rarely
O… Regularly
O… Often

4. How often do you play video games on computers?

O… Never
O… Rarely
O… Regularly
O… Often

5. How often do you play games on your mobile phone?

O… Never
O… Rarely
O… Regularly

6. Do you predominantly play 2D or 3D games on your mobile phone?

O… 2D
O… 3D

7. Are you familiar with 3D visualization applications such as DirectX Viewer, Autodesk products (Maya, 3D Studio Max, Mudbox), Cinema 4D, Zbrush, Marmoset Toolbag?

O… Yes, I am familiar with…
O… No

8. Have you ever used any 3D visualization applications for mobile phones?

O… Yes, I have used…
O… No

9. How often do you use your smartphone in landscape mode?

O… Never
O… Rarely
O… Regularly
O… Often

10. Do you prefer holding your mobile phone with one or two hands when writing messages or playing games?

O… One hand
O… Two hands

11. Do you prefer using the thumb or the index finger when writing messages or playing games?

O… Index finger
O… Thumb

Questions asked after the experiment:

12. Please rate the different parameters with respect to how difficult it was to perform each task:

Fixed-up Trackball					
Tasks	1 very hard	2 hard	3 moderate	4 easy	5 very easy
1) Find one emoticon	O	O	O	O	O
2) Count all emoticons	O	O	O	O	O
3) Estimate distances	O	O	O	O	O
4) Count particular emoticons (color & type)	O	O	O	O	O
5) Connect emoticon pairs	O	O	O	O	O
6) Count particular emoticons (type)	O	O	O	O	O

Two-axis Trackball					
Tasks	1 very hard	2 hard	3 moderate	4 easy	5 very easy
1) Find one emoticon	O	O	O	O	O
2) Count all emoticons	O	O	O	O	O
3) Estimate distances	O	O	O	O	O
4) Count particular emoticons (color & type)	O	O	O	O	O
5) Connect emoticon pairs	O	O	O	O	O
6) Count particular emoticons (type)	O	O	O	O	O

13. Which one of the two orientation modes do you prefer?

O… Portrait
O… Landscape

14. Which one of the two rotation modes do you prefer?

O… Fixed-up trackball
O… Two-axis trackball

15. In portrait mode, which one of these options do you think is more intuitive and natural?

O… Two-handed/portrait with finger
O… One-handed/landscape with thumb

16. In landscape mode, which one of these options do you think is more intuitive and natural?

O… Two-handed/landscape with thumb
O… Two-handed/landscape with finger

17. Would you use such an application to visualize 3D objects in a museum?

O… Yes
O… No

18. What other options would you like to see for this application? Examples might include:

O… Complete 3D rotation
O… Zoom
O… Flicking
O… Other (please mention them if you select this option)

19. Do you have any other comments/remarks to make about the experiment, overall procedure, application, etc?