CHORDBALL: A ROTATION TECHNIQUE FOR 3D VIRTUAL ENVIRONMENTS

M. Raees, S. Ullah and S. U. Rahman

Department of Computer Science and Information Technology, University of Malakand, Pakistan
Corresponding Author's email: sehatullah@uom.edu.pk

ABSTRACT: Rotation is one of the basic interaction tasks performed in 3D Virtual Environments (VEs). Chordball; a Rotation technique for touch-based, desktop and mouse-based 3D virtual environments was designed and evaluated. Index finger of a single hand was used for rotation about x, y and z axes. Interaction points, distance and direction of the finger's movement was traced for rotation about an axis. Torque-effect and rotation speed with respect to finger's move made the technique more realistic. The technique was implemented in Visual Stadio-10 using the OpenGl library and its performance was compared with other well-accepted rotation techniques. The technique was tested 288 times for rotation about different axes by twelve users. Statistical analysis revealed that the technique had better completion time with least loss of accuracy. It was concluded that the technique may improve rotation based interaction and can be used in a wide range of 3D virtual environments.

Keywords: 3D Rotation Technique, Interaction with Virtual Environments, ChordBall and Gesture based Rotation.

(Received 04-05-2016 Accepted 10-03-2017)

INTRODUCTION

Human-computer interaction in a 3D Virtual Space is to perform a task in virtual environment, metaphorically analogous to performing a real world action. The quality of a virtual environment depends on the level of interaction it provides between user and the designed environment (Stuart and Moran, 1986).

In 3D interaction, selection is to specify target for the desired interaction (Frank *et al.*, 2004). Navigation is a kind of interaction task representing movement of user from one location to a new target location in the virtual world (Doug *et al.*, 2004). Manipulation is scaling, translating and/or rotating a virtual object (Alex *et al*., 2004).

As specified by Hrimech (Hamid *et al.*, 2011), Rotation is a fundamental interaction task often required in VEs to orient a virtual object for viewing, inspection or for further 3D manipulation. Despite being a fundamental task, there is no established standard for rotating a 3D object (Doug *et al*., 2013). Rotation techniques are broadly categorized into view-based techniques, controller-based techniques, multiple DOF techniques and Virtual trackball techniques.

With View based techniques, different views of the same object are presented for rotation (Knud *et al*., 2004). In the Controller based techniques, object is rotated with a controller for each dimension (Vald, 2012) while with Multiple DOF technique, input devices with additional DOFs are required. The Virtual trackball projects motion of a 2D mouse onto a 3D sphere.

ChordBall is a general interaction technique designed specifically for rotation and is equally applicable for both desktop based and touch based systems. Inspired from the popular Virtual Trackball, ChordBall is object-centric as it allows a user to grab and rotate. After selecting a virtual object, index finger of the dominant hand is used to rotate the object about either axis by keep pressing and moving index finger. The direction of the finger's move identifies rotation about a specific axis. If the move is parallel to x-axis, rotation about y-axis is performed while for rotation about x-axis, the finger move should be parallel to y-axis. Rotation about z-axis is accomplished by making a linear trajectory on xy-plane. This is done by moving the finger either up- down (for Clockwise) or from down towards up (for Anti-Clockwise). In either case, the finger should move on x-axis in addition to its move along y-axis, Fig.1

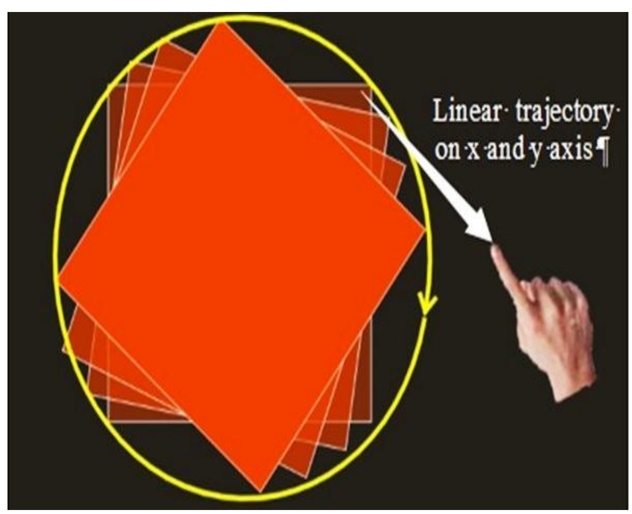

Figure 1: Finger move forming a linear trajectory on x and y axis simultaneously for rotation about z-axis.

MATERIALS AND METHODS

The proposed technique was to rotate a 3D object about one of the three axes (x, y and z) at a time. All in compliance with ergonomics' principles, rotation speed of ChordBall was based on the motion of finger. Quicker the move, speedier was the rotation and vice versa. Furthermore, naturalism was achieved by its support for torque effect.

As mobile phones were becoming ubiquitous, research about interaction with limited screen of mobiles was increasing. One key aim of ChordBall was to overcome the inherent constraint of occlusion by suggesting single finger based interaction. Facets of an object to be rotated were assumed to be inscribed by three virtual circles on xz, yz and xy planes. The virtual circles' diameters were taken equal to the corresponding facet length of the object. Thus, a circle on x-axis covered width of the object on x-axis, circle over y-axis the height of object on y-axis while the circle formed on z-axis was according to the depth or z-face of the object on z-axis, Fig.2.

The system tracked index finger move to set initial point (Press Point) and final point (Stop/Release point) dynamically for calculating angle of rotation. The axis of rotation was determined from the direction of the finger move on 2D plane. The distance between initial and final points of 2D screen was mapped using a function to calculate chord over 3D image plane. The calculated chord of a circle was then used to find out the angle for rotation. Initial point (P_i) and final point (P_f) with chord (C) are shown in Fig.3 (a) and (b) respectively.

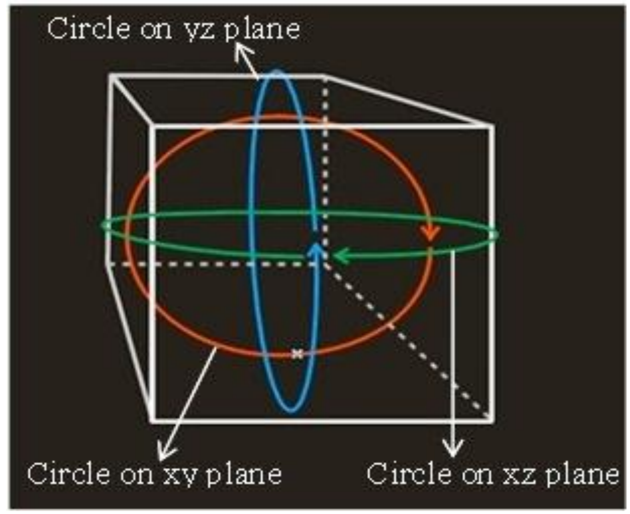

Figure 2: Virtual circles of a 3D Cube.

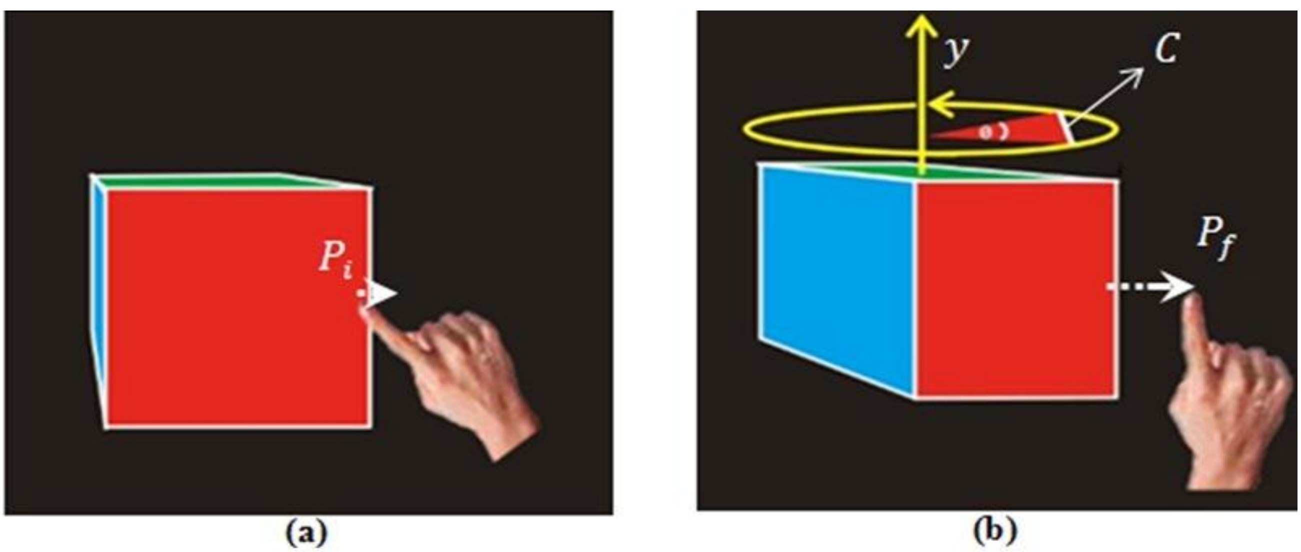

Figure 3: (a) Starting of rotation around y-axis (b) Cube Rotated by angle θ.

Intuitally predictable, if the finger move was horizontal to the viewport, rotation about y-axis was produced Fig.4(a). Rotation around x-axis was generated if movement of the finger was vertical to the viewport Fig.4(b).

Turning object about the look-vector (z-axis) was conditioned to both horizontal and vertical movements of the finger at the same time. The technique supported clockwise and counter-clock rotation based on y-coordinate of initial and final points. If the direction of the finger move was downward on right side of the point of rotation (1st and 4th quadrants) or upward at left side (2nd and 3rd quadrants) clockwise rotation was performed. The reverse of finger's move, was assumed to rotate the object anti-clockwise.

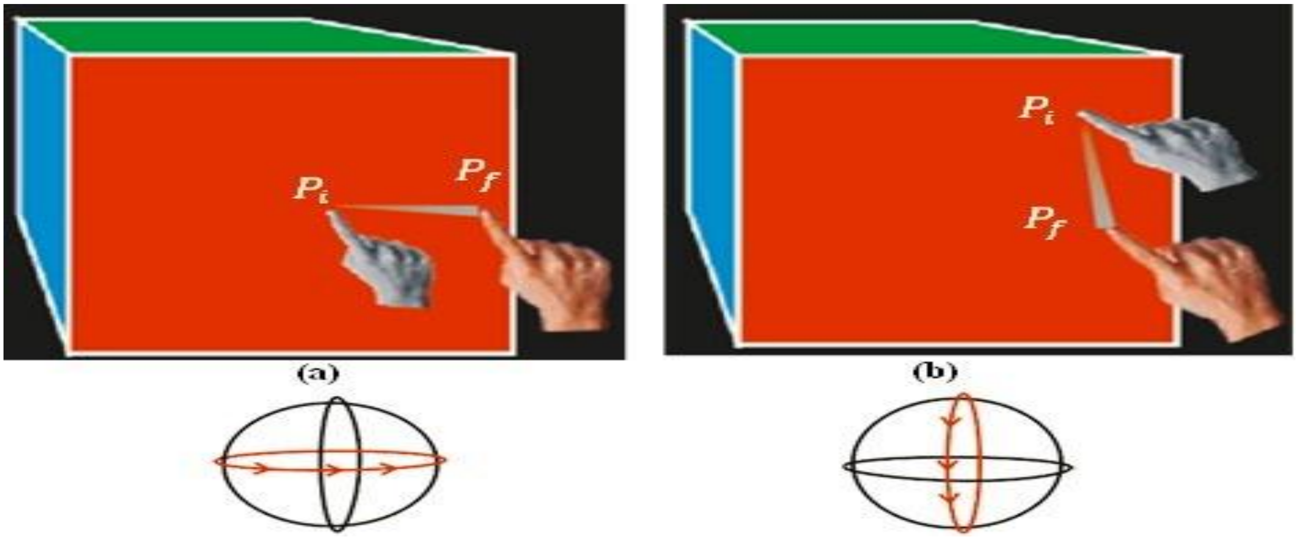

Figure 4: (a) Horizontal and (b) Vertical finger moves for rotation about y-axis and x-axis.

Fig.5 (a) Finger's move for Clockwise and (b) Anti-clockwise rotation

The algorithm followed for Clockwise and Counter-Clockwise rotations was as under,

If (Quadrant=1st OR Quadrant=4th)
If (P_i.y<P_f.y AND ABS(P_i.x-P_f.x)>1)
Clockwise Rotation
If(P_i.y>P_f.y ABS(P_i.x-P_f.x)>1)
Counter-Clock Rotation
If (Quadrant=2nd OR Quadrant=3rd)
If (P_i.y>P_f.y ABS(P_i.x-P_f.x)>1)
Clockwise Rotation
If (P_i.y<P_f.y ABS(P_i.x-P_f.x)>1)
Counter-Clock Rotation

Clockwise rotation as a result of the finger's downward move was:

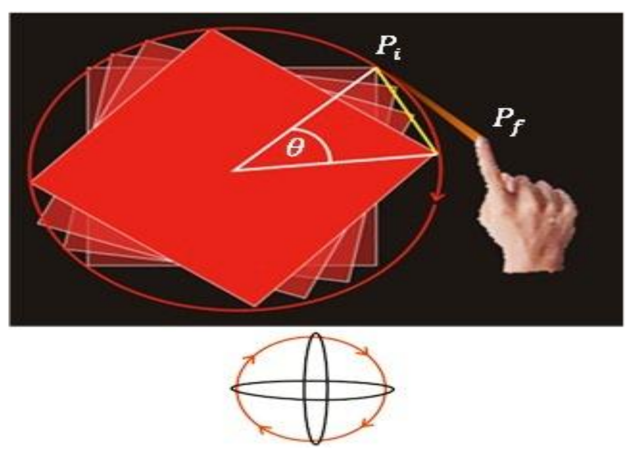

Figure 6: Downward finger's move for Clockwise rotation.

For torque, the arm of movement was calculated dynamically between starting point of interaction and point of rotation. If the distance between interaction point and point of rotation is greater, rotation by a larger angle is produced. Similarly, if both the points coincide, no rotation is performed. Rotation speed was based on the average time spent during the interaction for a single rotation about an axis.

***Mathematical Model*:** The 2D screen coordinates (x_p, y_p) were used for interacting with 3D image plane (x_p, y_p, z_p). The mapping from a 2D screen coordinates to the 3D image plane was performed by a function m_1. The amount of rotation was determined from the distance covered by the finger pressed at initial point $P_i(x_i, y_i)$ move to and released at final point $P_f(x_f, y_f)$. The corresponding initial point (*PI*) and final point (*PF*) on a circle were calculated as,

$$PI = m_1(P_i)$$
$$PF = m_1(P_f)$$

A rotation '*Rt*' about an axis '*A*' by amount 'θ' was made in the plane perpendicular to '*A*'. Let 'd' be the distance, $P_i(x_1, y_1)$ be the initial point and $P_f(x_2, y_2)$ be the final point of user's finger which directed move on world coordinates.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

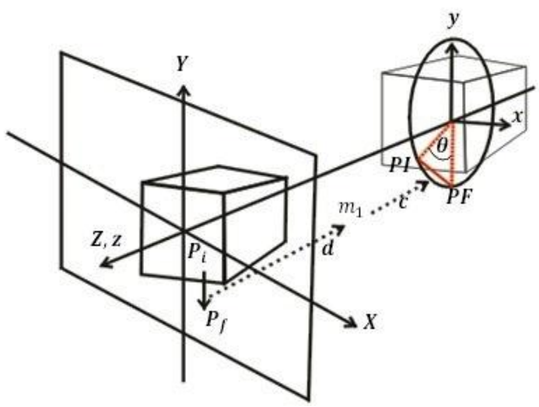

Fig. 7. Mapping of 2D screen coordinates to 3D image plane

Chord 'C' was obtained using function m_2 where actual distance 'd_a' between PI and PF was added with distance of PI from pivot point (M_a).

$$C = m_2(d_a, M_a)$$

Function m_3 finds out the angle for rotation as below,

$$\theta = m_3(C, R)$$
$$m_3 = 2 * sin^{-1}(C/_{2R})$$

Where C was the chord of a circle on x,y or z plane.

Rotation Speed: For rotation speed 'ω', the system was to trace finger's move with respect to time. Initial time (start time) and final time (stop time) of the finger for covering 'd' were taken as 't_i' and 't_f' to calculate ω,

$$\omega = \frac{\Delta\theta}{\Delta t}$$

where, $\Delta\theta = \Delta d$ and $\Delta t = t_f - t_i$

Torque Effect: For linear displacement $T = F.d.sin\,\theta$, but as *Force* was irrelevant while interacting with computer, the formula was not applicable. The moment-arm M_a between initial point P_i and point of rotation P_R was added with actual distance d_a covered by the finger.

$$d = d_a + M_a$$

Where, $M_a = (P_R - P_i)$

The faraway P_i from P_R, the greater the resultant d and hence greater was the rotation. Fig.8 illustrates rotation for smaller θ (a) and rotation for greater θ (b) for the same d_a.

After implementing the ChordBall, Arcball and Virtual Sphere techniques in Visual Studio with OpenGL, their performance was tested using four different rotation tasks. Although ChordBall was equally applicable for touch-based interaction, the technique was tested using mouse. Twelve participants, all males, ten right-handed and two left-handed, having ages between 25 to 40 years, evaluated the techniques using a corei5 laptop with 1366x768 true colors display. They were asked to perform different rotation tasks as quick as possible with least amount of mouse-clicks. The participants were allowed to take a short break between each task and trial.

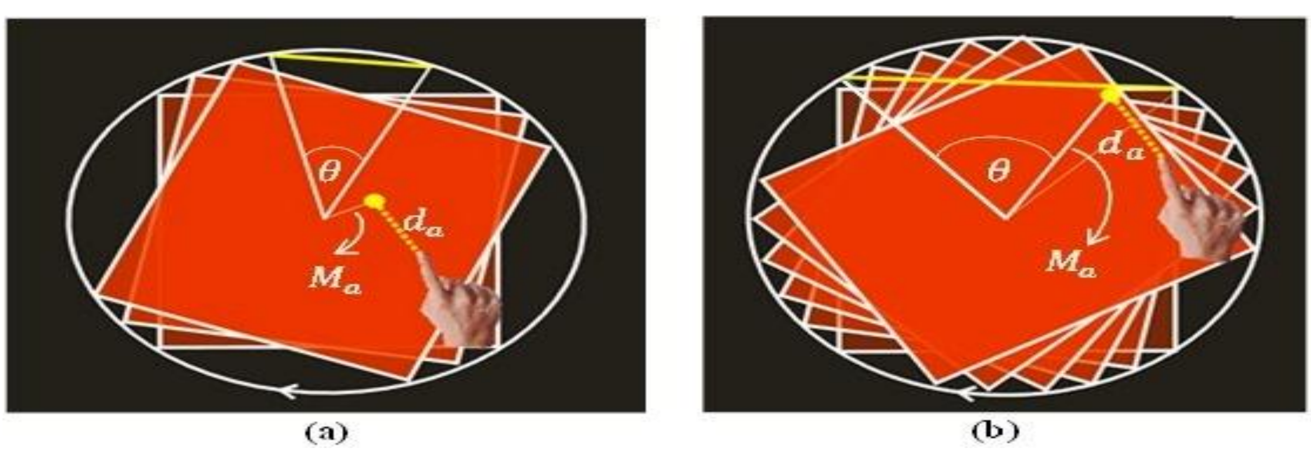

Fig.8 (a) Fewer rotation for a small M_a and (b)Greater rotation for a larger M_a

Task Description: Participants were asked to perform the following tasks on a 3D-cube rendered in a 600x600 pixel window (Fig. 9).

Task-1: Bring "Back" of the cube to front by rotating it about y-axis.

Task-2: Bring "Bottom" of the cube to front by rotating it about x-axis.

Task-3: Make the text on cube face upside down by perform rotation about z-axis.

Task-4: Bring "Top" of the cube to the position of "Bottom".

At the beginning of each trial, for each technique, only the "*Front*" face of the cube was visible, with zero rotation on either axes. To start timer for each task, tester had to click on "*Start*" button before performing the task. Completion time was displayed after clicking on "*Stop*" button.

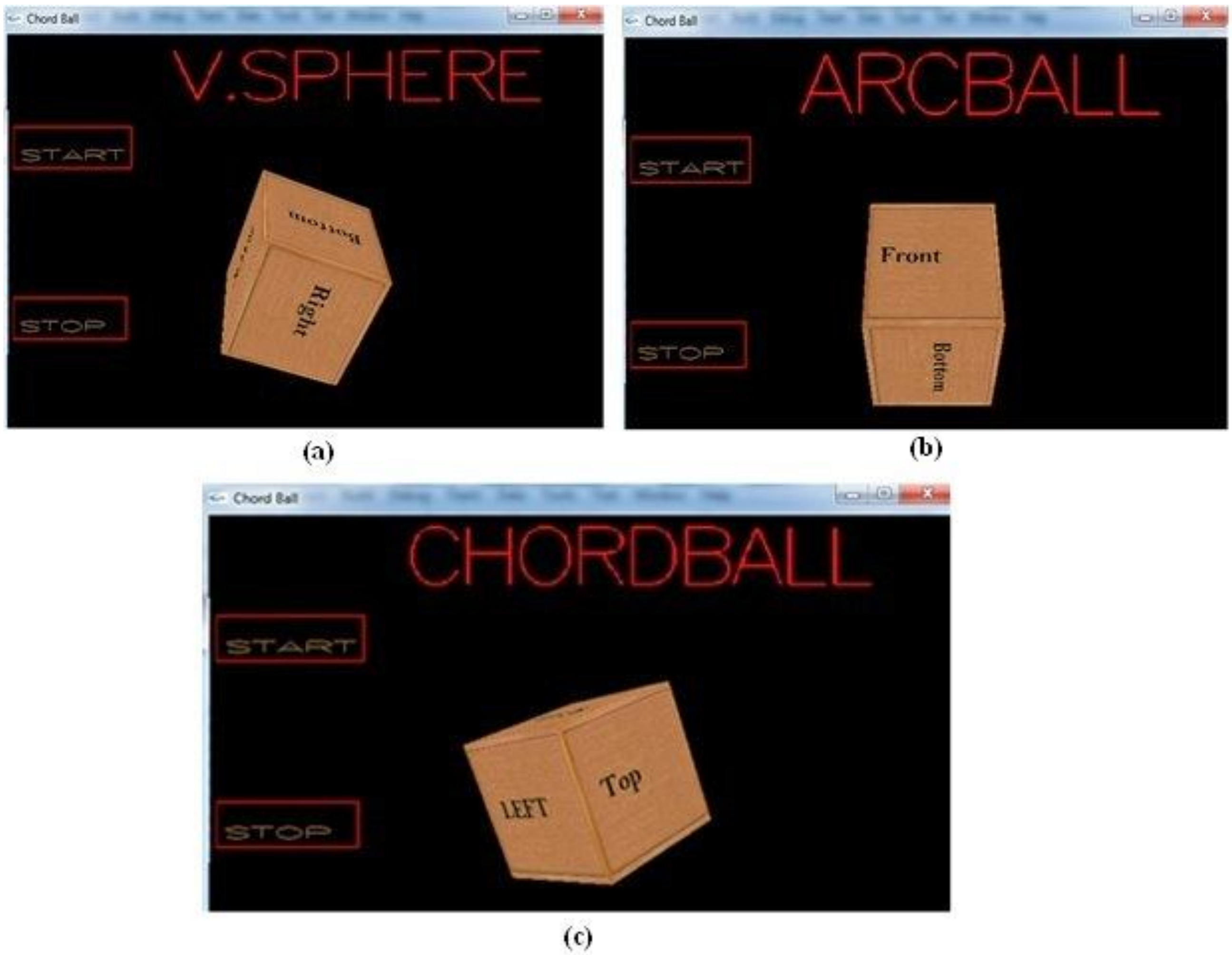

Fig.9 Testing environment of the algorithms of (a)Virtual Sphere (b) Arcball and (c) ChordBall techniques.

Experimental Protocol: Before actual testing, participants were introduced to the rotation techniques via a dedicated presentation/demonstration. Three Pre-trials were performed by each tester for each task using each of the technique. In order to make evaluation process biasless, domain for Technique (*Th*), Tasks(*Ts*) and Trials (*Tr*) was set as,

$Th_iTs_jTr_k$ where $1 \leq i \leq 3,\ 1 \leq j \leq 4,\ 1 \leq k \leq 6$

Participants were categorized into 3 groups containing 4 testers in each group. To test different Algorithms for different tasks in different sequence, the following evaluation protocol was followed.

G1: Th_iTs_j $\forall i \in \{2,3,1\},\ \forall j \in \{1,\ldots,4\}$

G2: Th_iTs_j $\forall i \in \{1,2,3\},\ \forall j \in \{1,\ldots,4\}$

G3: Th_iTs_j $\forall i \in \{3,1,2\},\ \forall j \in \{1,\ldots,4\}$

A total of 288 trials were made for each technique where each of the 12 participants performed 6 trials for each of the 4 tasks using a single rotation technique in a session lasting 3 hours with a short break. Click with no rotation, wrong rotation and reverse rotation were counted as error.

RESULTS AND DISCUSSION

The completion time (in seconds) of each trial was calculated dynamically by the system and was then recorded for statistical evaluation. Average of completion time of each technique is shown (Fig.10).

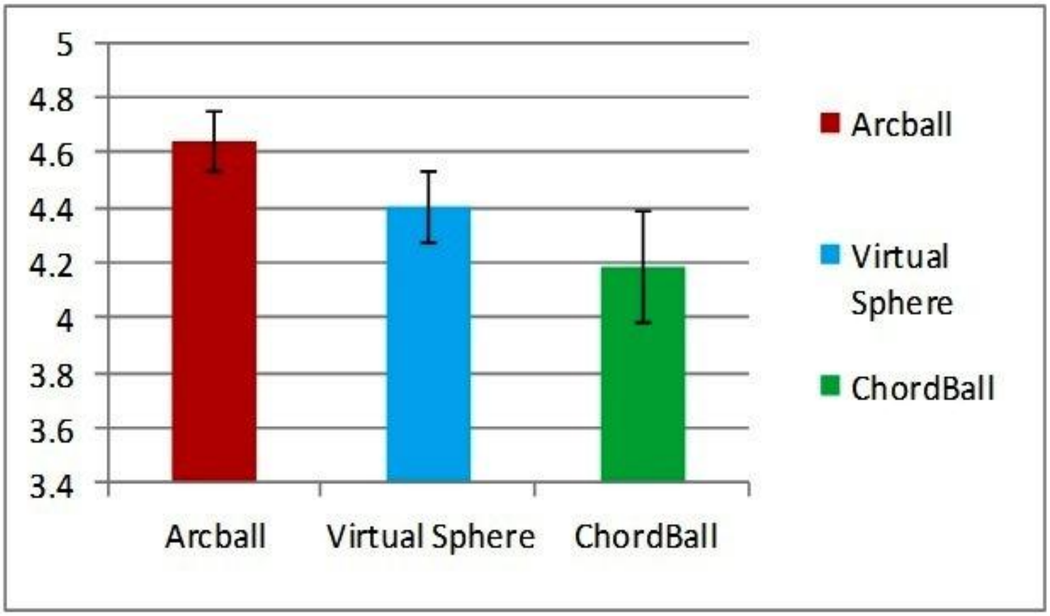

Figure: 10 Mean Completion time with Standard Deviation

To objectively test the performance of the designed technique, ANOVAs were conducted on completion time and errors. The first Single Factor F-test showed that means of all the four tasks of the three techniques were significantly different ($F_{2,15}$=6.16 , p<0.011). Repeated measure for comparison of means was also calculated in Ms. Excel. As Virtual Sphere and ChordBall had closer statistical results, the probability of Repeated Measure ANOVA was a bit higher ($F_{2,10}$=1.23,p<0.37).

The second F-test about error rate showed that the techniques were notably different in terms of errors ($F_{2,15}$ = 4.64 , p<0.026). Means and standard deviation of the errors was also calculated (Fig.11).

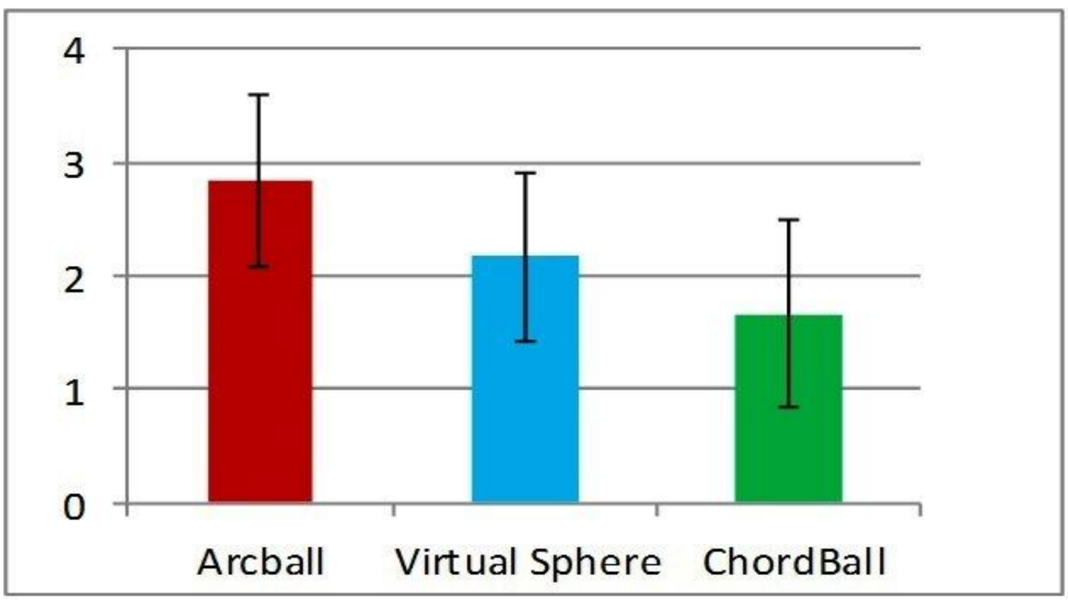

Figure 11: Mean and standard deviation of Errors

The learning effect of each technique was measured from the error occurrence rates and completion time. As shown in Fig.12, the number of error decreases significantly for ChordBall in the last three trials. The of task completion time for Virtual Sphere and ChordBall were decreased in subsequent trials as shown as secondary axis in the chart.

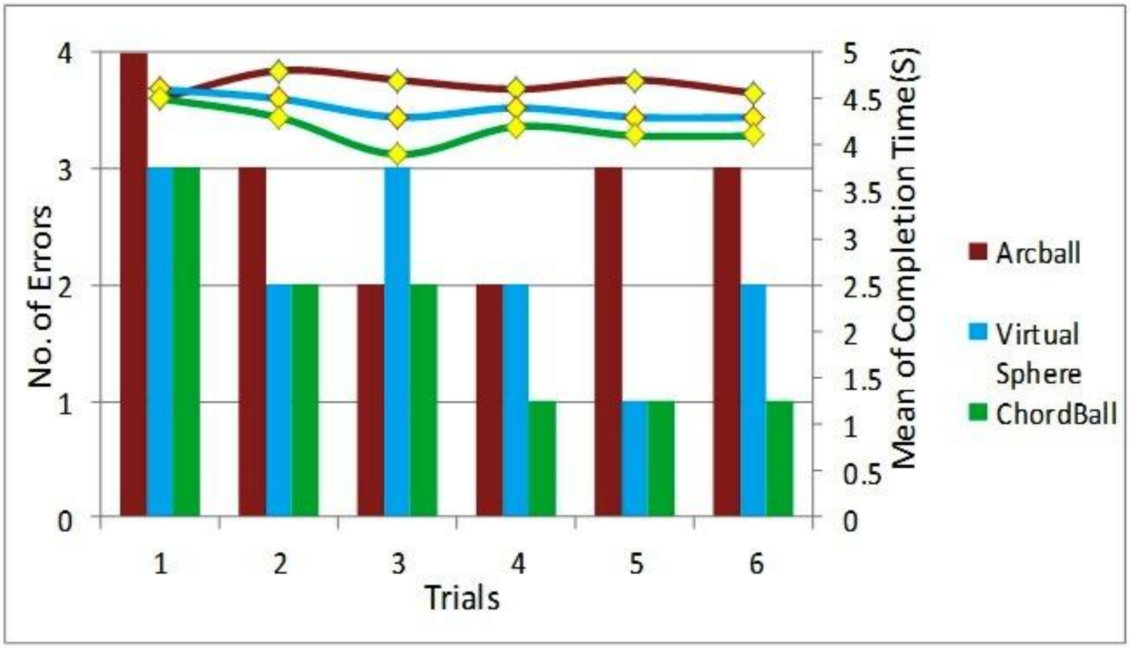

Figure 12: Errors and mean completion time of trials.

Subjective Analysis: At the end of the session we presented a questionnaire to the users for measuring the ease of use of the three techniques. Based on the subjective analysis of the techniques, ChordBall is the preferred rotation technique, as shown by the histogram in Fig. 13. The Chi-square test confirmed that it has significantly higher rating over the other two techniques ($\chi^2 = 18.1$, $p<0.001$).

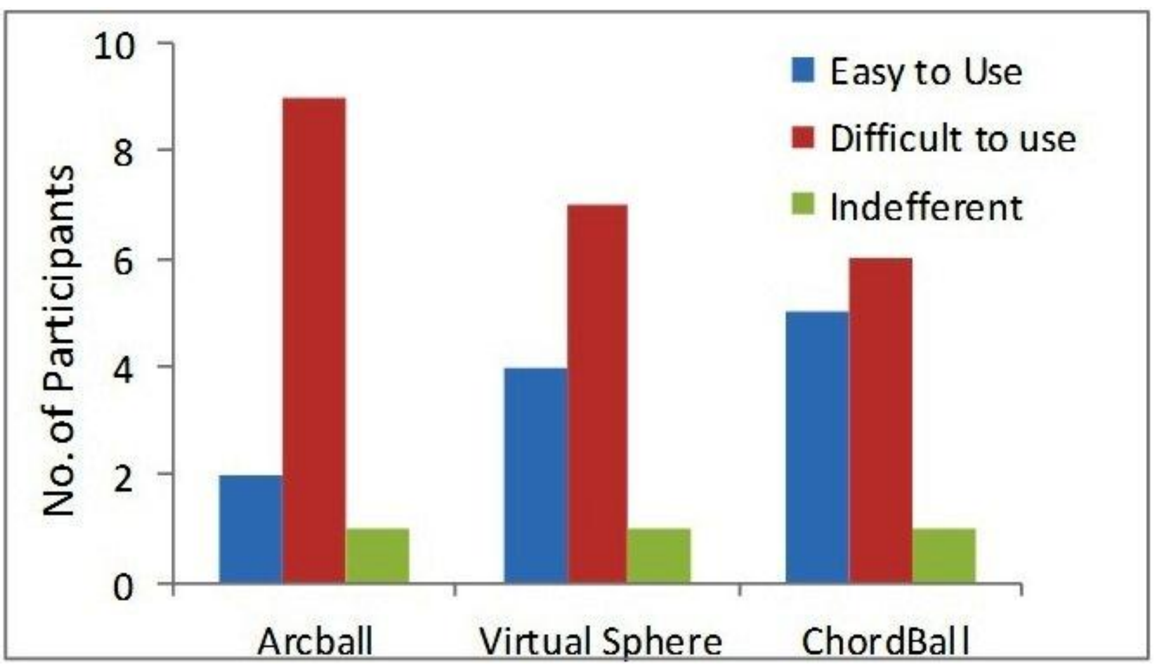

Figure 13: Testers' response about the three rotation techniques

To relate the findings of ChordBall with other techniques, the state-of-the-art research work of Zhao (Zhao *et al*., 2011) is pertinent to mention where the performance of three well-known rotation techniques were compared. As all the techniques have almost the same completion time and error rates, no significant learning effect on completion time ($F_{3,33}$ =1.5) and error ($F_{3,33}$ = 0.59) was reported. The statistical results obtained from the comparison of ChordBall with the two techniques showed that there was a significant difference in terms of completion time ($F_{2,15}$=6.16) and error ($F_{2,15}$ = 4.64). Furthermore, the greater standard deviation of ChordBall for task completion, its ease of use and the decrease in errors indicated that user can learn the technique more quickly. Gesture-based techniques were widely used for 3D interaction where postures of head or arm are traced, like in arm extension, ray casting and HOMER (Doug *et al*., 1999). The accuracy rates of such systems were subjected to the correct detection of gestures and were mostly remained variant to scale and shape of the body parts. The gesture based technique of Khalilpour (Khalilpour *et al*., 2013) used to extract static hand gesture for rotation using HSV color space. Although its accuracy was satisfactory (94%), the suffered badly from lighting condition and was suitable for controlled environment only. Similarly, In Multi-Finger interaction technique; reported by Mike and Balakrishna (2003) and Jason *et al*., (2009), two fingers were used for rotation where pivot finger was to identified the center of rotation and the second finger for the angle of rotation. The approaches did not work well in case of small objects rotation where the system failed to distinguish touching of the two fingers. Drawback of Virtual trackball technique (Vald, 2012) was that it failed to rotate object along an intended great circular arc. The well-known Arcball method (Shoemake, 1992) had to follow constraint arc using mouse where the half-length arc model was followed for rotation. The technique was suitable merely for desktop systems and remained inappropriate for 3D VEs and touch-screens. The rotation technique of Jason *et al.* (Jason *et al.,* 2009) though matched well for touch-screen systems but as both hands were used, therefore not applicable for mobile based interaction.

For interaction with mobile devices, the occlusion-free interaction technique was proposed by Daniel (*Daniel et al.,* 2007) but it necessitated the use of a touch-sensitive surface to the rear of the device. In the SideSight approach of (Alex *et al.*, 2008), IR distance sensors were used as responding multi- touch devices. The technique was applicable only when the device was placed on a flat surface. The Palm-Space technique (Sven *et al.*, 2012) designed specifically for rotation was workable only within the proximity of mobile devices.

The two-axes trackball, originally developed by (Thornton, 1979) and evaluated by (Chen *et al.*, 1988) was advantageous as it was predictable but its target was only two-axes rotation(x and y). Since there was no control over the look-vector, clock-wise or anti-clockwise rotation was not possible and was therefore not suitable for interacting with 3D VEs. The two handed technique of Zeleznik (Zeleznik *et al.,* 1997) was to rotate a 3D object with the help of two cursors, one for controlling orientation and another for controlling the cursor for certain degree of freedom. As for 3D interaction, input devices with more than two degree of freedoms are required (Shumin *et al.*, 1996) therefore the system falls short to be used in VEs. The technique of Laviola (Laviola *et al.*, 2007) developed for 6DOF non-isomorphic rotation was rarely affordable due to its requirement for costly devices like magnetic trackers and Wanda. Naturalism of the Leap motion based rotation system proposed by (Coelho *et al.*, 2014) Was high but its missed and false detection rate was higher than mouse, particularly for z-axis rotating. Most of the interaction techniques suffered from issues like temporal separation, tilting based interfaces and fat-finger problem (Sven *et al.*, 2010).

The ChordBall technique stays invariant to the mentioned problems and need no extra device for operation. Furthermore, as single finger is used for interaction; hence ChordBall minimizes the occlusion impasse. The rotation is proportional to the move of finger therefore the motor function and cognitive ability remained well harmonized in interaction process. Furthermore, ChordBall satisfies the four basic and standard principles (Table 1) as suggested by Rangar (Ragnar *et al.*, 2005) for real-like rotation. The principles are;

Principle-1: similar actions should provoke similar reaction (Replaceable by Principle-2 in case of non-desktop based systems).

Principle-2: Direction of rotation should match the direction of 2D pointing device.

Principle-3: 3D rotation should be transitive. Movement from A to B and then to C should be the same as from A to C.

Principle-4: The control to display ratio (C/D) should be customizable.

Table 1 Standard principals of 3D rotation and different Rotation techniques

Rotation Technique / Design Principles	TrackBall	ArcBall	Virtual Sphere	Virtual Trackball	Multi-Finger technique	ChordBall
Principle-2	×	✓	✓	✓	×/✓	✓
Principle-3	×	×	×	✓	✓	✓

The 4th principle is pursuable for desktop platform, due to 1:1 correspondence in case of mobile/touch-based systems C/D ratio is not pursuable (Amy et al., 2007).

Conclusion: The Chord Ball rotation was based on object coordinates rather than world-coordinates therefore not affected by the object size. Relevancy of the rotation technique for a broader range was proved statically, however, other interactions clearly remained to be done. We are determined to enhance the technique for selection and navigation as well in future.

REFERENCES

Alex O. and S. Feiner (2004). Modular development of distributed interaction techniques for highly interactive user interfaces. Proc. of the 2nd Int. Conf. on Comput. Graph., Interactive Tech., Singapore: 131-138

Alex B., S. Izadi, and S. Hodges (2008). SideSight: Multi-touch interaction around small devices. Proc. of the 21st Annual ACM Symp. on User Interf. Softw., Technol. , Monterey, CA, USA: 201-204

Amy A.K., and B.B. Bederson (2007). Thumb Space Generalized One-Handed Input for Touchscreen-Based Mobile Devices. Int.l Conf. on HCI, Rio de Janeiro, Brazil: 324-338

Li C., S. Pion, and C.K. Yap (2005). Recent progress in exact geometric computation. The J. Log. Algebr. Program. (JLAP)64(1):85-111

Chen M., J. S. Mountford and Sellen, A. (1988). A Study in Interactive 3D Rotation Using 2D Control Devices. Proc. of the 15th Annu. Conf. on Compu. Graph. and Interact. Tech.: 121-129.

Coelho, Joanna C. and Fons J. V. (2014). Pointing Task Evaluation Of Gestural Interface Interaction In 3D Virtual Environment. Int. J. on WWW/Internet (IJWI),12(1): 113-130

Daniel W., C. Forlines, P. Baudisch, J. Barnwell and C. Shen (2007). LucidTouch: A see-through mobile device. Proc. of 2th Annu. Symp. on user Interface Softw., Technol., Newport Rhode Island, USA: 269-278

Doug, A. B., D. B. Johnson, and L. F. Hodges (1999). Testbed evaluation of virtual environment interaction techniques. Proc. of the ACM Symp. on Virtual Real. Softw., Techno., London United Kingdom : 26-33

Doug A. B., E. Kruijff, J. Joseph, LaViola and P. Ivan (2004). 3D user interfaces: theory and practice. Addison-Wesley Inc.; Redwood City, CA, USA. 124 p

Doug S. and W. Stuerzlinger (2013). A one-handed multi-touch mating method for 3D rotations. CHI '13 Extended Abstracts on Human Factors in Computing Systems, Paris, France: 1623-1628

Frank S., T. Ropinski and K. Hinrichs (2004). Object selection in virtual environments with an improved virtual pointer metaphor. Int. Conf. on Comput. Vision and Graph. (ICCV), Warsaw, Poland: 320-326

Graham B. (1988). Bell's trackball. Written as part of the" flip" demo to demonstrate the Silicon Graphics (now SGI) hardware.

Hamid H., L. Alem and F. Merienne (2011). How 3D Interaction Metaphors Affect User Experience in Collaborative Virtual Environment. Adv. Hum. Comput. Interact: 2011(1): 11-22

Jason L. R., L. Philip, Davidson, and J. Y. Han (2009). A screen-space formulation for 2D and 3D direct manipulation. Proc. of the 2nd Annu. ACM Symp. on User interface Softw. Techno., New York, USA: 69-78

Khalilpour, M. and N. Razmjooy (2013). A Hybrid Method for Gesture Recognition. J. of World's Elect. Eng. Techno. 4(2): 70-75

Knud H., J. Sporring, and K. Hornbaek (2004). Virtual Trackballs Revisited. IEEE Trans. Vis. Comput. Graph., 10(2):206-216

Laviola, J. J., & Katzourin, M. (2007, March). An exploration of non-isomorphic 3D rotation in surround screen virtual environments. In 3D User Interfaces, 2007. 3DUI'07. IEEE Symp. on. IEEE.

Mike W., R. Balakrishnan (2003). Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. Proc. of the 16th Annu. ACM Symp. on User interface Softw. Techno., Vancouver, Canada: 193-202

Ragnar B., F. Ritter and B. Preim (2005). Usability Comparison of Mouse-Based Interaction

Techniques for Predictable 3D Rotation. SG'05 Proc. of the 5th Int. Conf. on Smart Graph.:138-150

Shneiderman B. (1987). Direct Manipulation: A Step Beyond Programming Languages. Human-computer interaction: a multidisciplinary approach, Morgan Kaufmann Publishers Inc.; San Francisco, CA. 187 p

Shoemake K. (1992). Arcball - A User Interface for Specifying 3D Orientation Using a Mouse. Proc. of the Conf. on Graph. Interface, Vancouver, British Columbia:151-156

Shumin Z., P. Milgram and W. Buxton (1996). The Influence of Muscle Groups on Performance of Multiple Degree-of-Freedom Input. Proc. Comp. Human Interaction Conf., Vancouver,BC, Canada: 308-315

Stuart C. and T. Moran (1986). User technology: from pointing to pondering.

Proceedings of the ACM Conference on The history of personal workstations, Palo Alto, California, USA: 183-198

Sven K. and M. Rohs (2010). Extending the virtual trackball metaphor to rear

touch input. Proc. of the IEEE Symp. on 3D User Interfaces, Waltham, MA: 111-114

Sven K. , M. Rohs , D. Guse , J. Muller , G. Bailly and M. Nischt (2012).

PalmSpace: continuous around-device gestures vs. multitouch for 3D rotation tasks on mobile devices, Proc. of the Int. Working Conf. on Adv. Visual Interfaces, Capri Island, Italy: 181-188

Thomas G. Z., L. Jaron, B. Chuck, B. Steve and H. Young (1987). A Hand Gesture Interface. Proc. of the SIGCHI/GI Conf. on Human Factors in Compu. Sys. Graph. , Toronto, Canada: 189-192

Thornton, R. W. (1979). The Number Wheel: A Tablet Based Valuator for Interactive Three-Dimensional Positioning. Proc. of the 6th Annu. Conf. on Compu. Graph. and Interactive Techniques, Chicago, Illinois, United States: 102-107

Vlad B., (2012). Rotation techniques for 3D object interaction on mobile devices, Doctoral dissertation, Utrecht University.

Yao, J. Z., D. Shuralyov and W. Stuerzlinger (2011). Comparison of multiple 3D rotation methods. IEEE Int. Conf. on Virtual Env., Human-Computer Interfaces and Measurement Syst. Proc., Ottawa, Ontario: 1-5

Zeleznik, R. C., A. S. Forsberg, and P.S. Strauss (1997). Two pointer input for 3D interaction. Proc. of the 1997 Symp. on Interactive 3D graph., Providence, Rhode Island, USA: 115-120.