Vol 8. No. 2 Issue 2 – August, 2015
African Journal of Computing & ICT

www.ajocict.net

An Object Selection Model for a 3D Gaming Environment

D. Allenotor, A. Ojugo & D. Oyemade
Department of Mathematics & Computer Science
Federal University of Petroleum Resources
Effurun, Delta State, Nigeria
Delta State, P.M.B. 1221, Nigeria
\allenotor.david@fupre.edu.ng, [2]ojugo.arnold@fupre.edu.ng, [3]oyemade.david@fupre.edu.ng

ABSTRACT

We present a design study for pointing movements to verify the effects of 3D visual environment on object selection using motor control systems theory. Our results support a strong mathematical relationship between object orientation, object size, and object speed and the degree of accuracy of object selection in a 3D environment. We use artificial application to capture data from participants and test our data using 100 subjects in three age groups; 10-25 years, 26-55 years; and 56 years and older. Using the results obtained, we a write a general expression for movement time for the selection techniques under study.

Keywords: Index of difficulty, reachability, movement time, human motor system, Fitts' Law.

African Journal of Computing & ICT Reference Format:
D. Allenotor, A. Ojugo & D. Oyemade (2015): An Object Selection Model for a 3D Gaming Environment
Afr J. of Comp & ICTs. Vol 8, No. 2, Issue 2. Pp 115-124.

1. INTRODUCTION

A large body of research efforts in literature (for example, [2, 3, 5]) applied Fitts' law in varied perspectives to model human motor systems. Fitts' law [3] provides the relationship between an object and its target using the human motor system. It defines Movement Time (MT) as a function of the distance between the target and object A, the width of the region which the object must move to get the target (W), and some empirically determined constants (a and b). MT as presented in Fitts model is given as:

$$M = a + b \times IoD \qquad (1)$$

where, MT is movement time, a, b are regression coefficients (experimentally derived constants), and IoD is the index of difficulty (bits). The index of difficulty measures the degree of difficulty encountered when a target is selected. IoD's measurement is determined by the target's size and its distance. Fitts [2] compute the index of difficulty as:

$$IoD = log_2\left(\frac{A}{W} + 1\right) \qquad (2)$$

where A is called the amplitude or distance to the target and

W is called the width of the target. By varying values for the IoD, the constant values for a and b can be computed using Equation (1). Obtaining the optimum value for A and W is one of the challenges for manipulating MT. Fitts' Law recognizes the need to further optimize MT by either reducing the distance A or increasing the target width W. While this approach may help to optimize target reachable time in the physical world, studies [6] have shown that changes in A and W affects target reachability. For example, in the virtual world it means changing the size as well as the position of the icons displayed on the screen. This may effectively result to larger screen displays as against current trends in miniaturization of digital circuits. Wobbrock et al. [8] show that applying Fitts' law alone can only achieve movement time prediction while hiding inherent errors if movement is associated with changes in direction as well. In a related study, and in several literature Fitts' law determines the suitability of a selection technique given a specific task. In many applications, only movement in the x and y axis are captured.

However, in real-life scenarios (gamming and targeting objects), movements occur in the z-plane also. In this paper we modeled the mouse which is a typically produces an x-y movements and adapted it to function as a though it were in a 3D gamming environment. In the current study, we examine the effect of environment on type of selection techniques such as the point cursor, area cursor, and volume cursor to select objects of various sizes, speeds, distracter densities, and orientation.

Vol 8. No. 2 Issue 2 – August, 2015
African Journal of Computing & ICT

www.ajocict.net

1.1 Effects of Manipulating Parameters of Fitts' Law

Pointing and selecting a desired target is improved by changing some parameters of the Fitts' law. For example, W and A may be varied by increasing (W) and decreasing (A). The aim is to bring the target closer for selection as much as possible. One of the techniques for achieving this objective is to modify the cursor activation area to obtained a faster object selection time in x-y direction. However, the error rate in their object selection approach increased with a large number of multiple objects in the selection environment. Balakrishnan [1] and Grossman and Balakrishnan [9] proposed three design techniques to enhance pointing by reducing target distance.

First; by moving target to the nearest position of the cursor e.g., drag and drop and drag and pick. Second; by bringing potential target to the cursor by moving it temporary instead of changing the size and position, and third; by removing the empty spaces in between cursor and target. We test . These techniques have the inherent tendency for reducing the efficiency of accurate pointing and a limitation for 3D gamming application. The rest of this paper is organized as follows. In Section 2 we provide a review of related work. The experimental design and setup is provided in Section 3. In Section 4 we present and discuss the results of our experiments. The paper is concluded in Section 5 with some future directions.

2. RELATED WORK

Many research efforts attempt to beat Fitts' law. Balakrishnan [1] suggests three schemes that could facilitate virtual enhancements for pointing operations which when justified may beat Fitts' law. The categories include reducing, increasing W, and decreasing and increasing at the same time. However, reducing will involve widgets design capable which could actually lead to minimizing. The scheme presented in Balakrishnan [1] brings the target closer to cursor. For example, a navigation of a contextual linear pop-up menu and a pie-menu show that the pie- menu shows that all objects are locatable equidistance distance always equal the pie radius. However, in the liner pop-up menu, the objects farthest from the cursor is located at the bottom of the menu.

The known drawback of the pie menu lies in the design overhead and how to manage sub-menus. Another technique suggests a scheme where potential targets are located close to the cursor temporarily. The Balakrishnan [1] approach achieves reduction however; it does not necessarily result to re-engineering the underlying layout. This is because the cursor reacts to unidirectional sensitivity while it predicts potential targets. This scheme is found to circumvent the inadequacies of the technique – that is interface unchanged, user manipulates objects at closer range – reduces D without re-design). However, with crowded objects, it may be hard to implicitly select an object form a cluster of objects.

Therefore, this scheme will need further design in the selection a target from a group of other possible likely targets. This selection technique therefore is not interference-free which could cause a user frustration. The article [1] also considered a method of empty spaces removal between the cursor and the targets. The removal of empty spaces between target and objects results to making D smaller which is another way to bring targets nearer objects in the virtual world. The author also argued that because all the spaces around the object and target will not be used to 100% during a selection, keeping them would result to selecting unimportant objects. Guiard et. al [13] provides a variant of the scheme in [1].

The variant scheme skips across empty spaces to an object. A disadvantage of the empty spaces removal technique is that a group of the tightly coupled objects under one menu may become selected similar to the approach that achieves a reduction in D. The area cursor applies a technique W reduction technique. This means that smaller objects could be selected. However, those smaller objects would have had a higher selection difficulty (Index of difficulty (IoD)) using the default cursor. Despite the advantage of the area cursors, studies in literature shows that as the area of the cursor gets larger, there is a possibility that it will obscure underlying area (this disadvantage is particularly evident in our experiments). Similarly, identical objects that are close together could be selected indiscriminately. Suggested remedy includes creating hot spots especially in a control environment, or an area cursor that transit into point cursor when the target gets closest.

Another technique that is presented in [13] is the concept of expanding targets when the cursor gets closer to the target (a scheme used by Apple MacOX "dock"). In this scheme dynamic sized widgets increases in size as a cursor moves over them. Similar to area cursors, tightly packed widgets may suffer from indiscriminate selection. A seem hybrid approach decreases and increase; described as control-display gain (C-D gain). This technique is based on the device movement speed and it does not consider the location and size of the potential target. The scheme performs with one target in the object space, however, with multiple objects, performance deeps because of the multiple targets tend to hinder movement along the desired target and it's a worse case in a 3D environment. This is the core challenge that we experiment in this paper.

Another challenge that motivated thoughts in the literature is the need to reduce error rates. Vogel and Baudish [12], propose a pointing technique designed to reduce the error rates since there is a difference between position of objects in real world compared to the virtual world. The technique proposed by Vogel and Baudish [12] supports the selection of object. However, as the objects gets smaller in size, the efficiency of selection diminishes. In another research carried out by Vogel and Baudish they studied the strengthens the weakness of the Offset Cursor approach provided by Potter et. al [10] and Sears and Shneiderman [11].

Vol 8. No. 2 Issue 2 – August, 2015
African Journal of Computing & ICT

www.ajocict.net

Vogel and Baudish argues that although the Offset Cursor achieved some successes, however, the Offset Cursor technique was characterized by some costly overheads, incapable of accessing the unreachable areas of the display, and does not make users aim for the target directly. Balakrishnan in [1] and in [9] proposed the bubble and volume cursors. These input techniques were improvements over the point cursor. These input techniques have ability to dynamically change their area and thus, enhance their selection capabilities. In these various studies, the proposed input techniques were efficient for 2D environments.

They will not function better in a 3D settings of a gamming environment. For example, the pint cursor must reach the target before a success section will be recorded. In that same line of difficulty of selection, the bubble and the area cursor will unnecessarily expand and select other objects, a process of clutching will occur. The main drawback of this approach is using this technique it is difficult to point a specific target when multiple targets are closely grouped to each other. One of the main drawback of area cursor is it takes more than one target and it becomes very difficult when multiple objects are close to each other. The drawback must be avoided in a 3D as we present in the current paper.

3. EXPERIMENTAL DESIGN AND SETUP

In our experiments 100 participants (50 females, 50 males) participated. Their ages ranged from between 10 years and 60 years old. All participants had normal vision. 15% of the participants used for our experiment are left-handed users and the rest 85% used their right hand to operate the mouse in a gamming environment. Staff and students of the Federal University of Petroleum Resources, Effurun Delta State, Nigeria participated in the experiment. They used a standard wireless Logitech performance MX mouse as the pointing device to run the experiment. Keyboard was not a required input device for this experiment.

3.1 Apparatus

The implementation for the 3D was done under the Microsoft tool for 3D animation, XNA Studio. Currently, XNA consists Microsoft's complete Game Development sections which also include the standard Xbox Development Kit. The experiment was conducted on a 3.6 GHz Core i7 6600U Processor (Dual Core), Intel HD graphics 520, 16.00 GB RAM system that runs the Windows 10 (64 bit) OS and a 15 inches LCD display at 1920×1080 resolutions. Mouse acceleration was set to 0, and the C–D gain was set to middle value of control panel of windows 10.

3.2 Input Techniques

We used 3 input techniques. The point cursor is a standard arrow pointer is displayed as cross hair cursor without virtual enhancements. All input techniques in the 3D environment use a sphere, however, behavior and shape size vary in the selection techniques.

3.3 Experiment Procedure

Prior to beginning the experiment, the experimental concepts as well as the 3 types of selection techniques were explained to the participants. The experimental setup consists of 100 participants, 3 input techniques, 3 target sizes, 2 target distances/speeds, 23 direction$(x \leftrightarrow x',\ y \leftrightarrow y',\ z \leftrightarrow z')$, and 10 repetitions. This gives a total of 36,000 individual trials. The expected average time to run the experiment is estimated for about 5 minutes for one trial. Experimental data were captured in real-time while participants ranking followed according to their number of hits or misses for all three object selection techniques. A hit is recorded as a successful selection of the moving target using the animated cursor. A fully crossed design resulted in 36 combinations of size, speed, and direction. The session was broken up by cursor type, with three blocks of trials completed for each cursor. In each block participants completed trial sets for each of the 36 combinations of size, speed, direction, and the three input techniques presented in random order. We write the function of the order of the experiment to compute the total number of trials (T_r) as:

$$T_r = f(p, i_t, S_z, S_v, S_w, n_r) \qquad (3)$$

where p defines the number of participants in the experiment, $i_t$is the numbers of input techniques, S_z is the target size, $S_v$is the target speed, S_w is the target direction, and n_r is the number of repetitions.

The individual 3D experiment consists of tasks. Each of the tasks have 54 selections. This totals up to 162 trials. Targets vary in size (pixels), speed (milliseconds), and direction or orientation in the z-axis. Each setting used one level of each target size small (20 pixels), medium (40 pixels) and large (60 pixels), two separate speeds; slow: (128 pixels/sec) and fast: (256 pixels/sec), and direction zz' where $z'$is the z-out. Figure 1 shows the interfaces' coordinate planes.

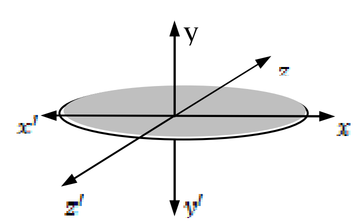

Figure 1.0: The Coordinate Planes of Experiment

Vol 8. No. 2 Issue 2 – August, 2015
African Journal of Computing & ICT

www.ajocict.net

The experiment environment also has a number of distracters that changes orientation and speed as the target object. The start experiment starts with the cursor–the spherical earth shape which can be moved using the mouse x – y directions. When a successful object selection is made the target turns green for a hit selection. To initiate a $z \leftrightarrow z'$ movement the right button is held down and the a left click. A missed target appears as red while for a hit, it appears as green. The target object appears as red spheres and the distracters (non-targets) are grey. The experiment also computes the statistics as the experiment progresses. The experiment boundary is enclosed in six walls. Five walls that enclose the movement area, four walls for the sides and one at the back.

There is also an invisible sixth wall in front of the camera. Figure 1 shows the various mouse movements in 3D gaming environment. A left and right mouse movement moves the cursor towards the x-axis and a vertical move of the mouse corresponds to the y-axis of the cursor movement. Pressing either the middle button or the right mouse button moves the object towards the z-axis going away from the camera. The reverse z-direction (z') is obtained by pressing the scroll button. In either case, selection is made by the left click of the mouse button. Prior to the start of the experiment, the 3D environment is setup by selecting various conditions of the experiment such as: Technique type, object size, orientation, number of distracter objects, and speed in a setup information file scene.xml.

3.4 Experimental Design

We applied a within-subjects design in our experiments and used input technique, target size, and target distance as independent variables. We applied a fully crossed design where each participant used one input technique to carry out 10 independent trials with each combination of distance, size, and width appearing in a random order. We balance the order of input. The technique used by the participants by means of a Latin square. From Equation (4), the entire experiment consists of a total of 36,000 trials for 100 participants.

$$Trials = \begin{cases} 100(prticipants) & \times \\ 3(input\ technique) & \times \\ 3(target\ size) & \times \\ 2(Distracter\ densities) & \times \\ 2(target\ distances) & \times \\ 10(repetitions) & \times \\ \quad = 36{,}000 & \end{cases} \qquad (4)$$

4. RESULTS AND DISCUSSIONS

In the test data analysis, the Randomized Block Design (RBD) was applied to measure the efficiency different input techniques (point cursor, area cursor, and volume cursor) in 3D environment. In the analysis we carried out, D_{ij} is used to simplify data analysis, where $i = 2$ for $2D$ and $i = 3$ for $3D$. Similarly, $j = 1,2,3$ means point cursor, area cursor, and volume cursor respectively. Hence, the data analysis results show $D21, D31, D32, D22, D23,$ and $D33$. For the RDB, techniques serve as treatments and participants' played the role of blocks or replication. The analysis of data presented is carried out using using SPSS 16.0. In order to examine the effectiveness of the three techniques, a paired t-test was used. The basic assumption of the paired t-test (specifically under the current experiments) is that samples are dependent. In this experiments, the same set of participants were engaged for both environments. First, 50 of the 100 participants did started with experiment I (2D) and ended up with experiment II (3D). The other 50 sets of participants started with experiment II and ended with experiment I. It was observed from the analysis that the efficiency of the point cursor in both environments is not significantly different (it is significant only at 89.8%). Figure 10 shows that the area cursor similarly indicated that it is significant only at 18.7%. However, the volume cursor is highly significant in both environments (2D and 3D). The regression analysis test was also carried out in these experiments. The Figures (2, 3, and 4) shows the regression analysis for the point cursor, area cursor, and volume cursor respectively. We obtained the empirical value for the intercept and the slope (log2((A/W) + 1)) in all cases. For the point cursor (technique 1), a = 0.3662 and slope 0.0009 and R2 = 0.6734; for the area cursor (technique 2), a = 0.2003 and slope = 0.0007 and R2 = 0.3262; and for the volume cursor (technique 3), a = 0.4366 and slope = 0.0008 and R2 = 0.7499. Based on the experiment carried out, and using the values for (a, b), we derived a general expression for computing MT in Equations (5), Equation (6) and Equation (7) written respectively for the point cursor, area cursor, and volume cursor for the 3D dimensional plane.

African Journal of Computing & ICT

www.ajocict.net

Table 1: Extract of Data Collected from Experiment

Technique	Object amount	Object Size	Object Speed	Orientation (A/D)	Hit/Miss	Distance since last click(px)	Time since last click (ms)	Average Mouse Speed (px/s)	Miss Distance	O/U
PC	2	20	1	A	Miss	921	1864	494.1	8	U
PC	4	20	1	A	Miss	217	1194	181.74	2	U
PC	2	40	1	A	Hit	1295	1369	945.95	0	
PC	4	40	1	A	Hit	290	1050	276.19	0	
PC	2	60	1	A	Miss	1782	1736	1026.5	1	U
PC	4	60	1	A	Miss	323	1579	204.56	4	U
AC	2	20	1	A	Hit	891	1996	446.39	0	
AC	4	20	1	A	Hit	447	1257	355.61	0	
AC	2	40	1	A	Hit	239	1226	194.94	0	
AC	4	40	1	A	Hit	246	1061	231.86	0	
AC	2	60	1	A	Hit	465	1341	346.76	0	
AC	4	60	1	A	Hit	526	1727	304.57	0	
BC	2	20	1	A	Hit	345	1418	243.3	0	
BC	4	20	1	A	Hit	163	780	208.97	0	
BC	2	40	1	A	Hit	238	777	306.31	0	
BC	4	40	1	A	Hit	549	1976	277.83	0	
BC	2	60	1	A	Hit	374	806	464.02	0	
BC	4	60	1	A	Hit	155	941	164.72	0	

In Table 1, the following keys apply: PC, Point cursor; AC, Area cursor; BC, Bubble cursor; A/D, Advancing/distancing; O/U, Overshoot/undershoot. Table 1 shows the snippet of data captured during the experiment. The parameters of interest include the type of technique (the techniques are point cursor, area cursor, and bubble cursor), object amount, object sizes (measured in pixel), orientation (either approaching or distancing – this is the movement in the z-direction), hit/miss captures the hit or miss counts. Other captured parameters include distance since last click. This parameter indicates the distance the target has moved since the last successful hit. The time since last click provides information about the time it has taken since the last time a selection was made. The average mouse speed and the miss distance are also measured. The miss distance is always zero for a hit. Any click made is also capture and determined whether its undershoot or overshoot.

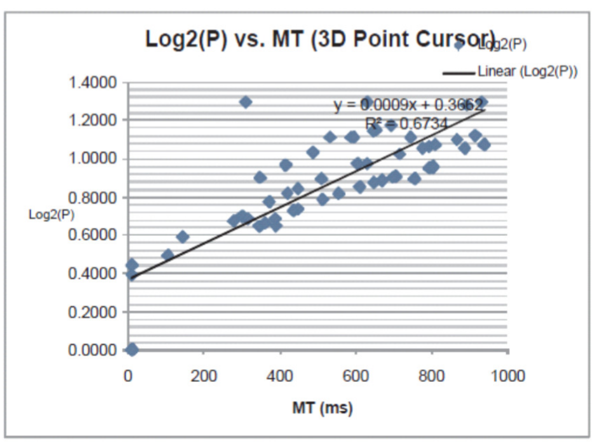

Figure 2: Regression Analysis for 3D point Cursor.

African Journal of Computing & ICT

www.ajocict.net

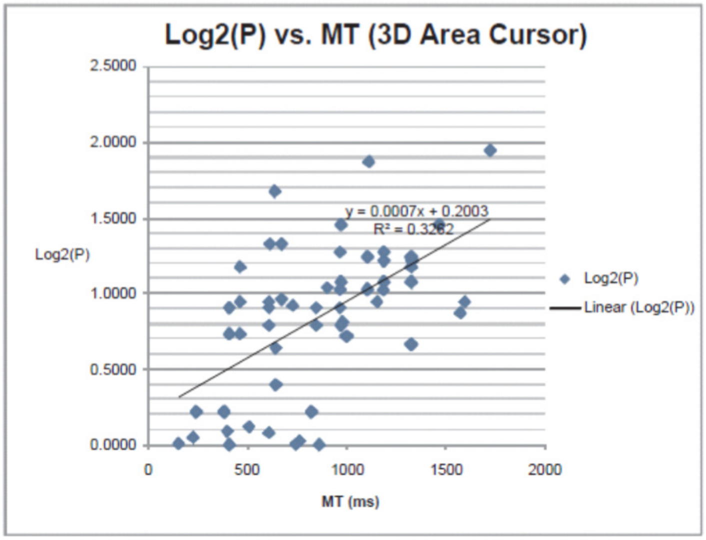

Figure 3.0: Regression Analysis for 3D Area Cursor

$$MT = 0.3662 + 0.0009\ log_2\left(\frac{A}{W} + 1\right) \qquad (5)$$

$$MT = 0.2003 + 0.0007\ log_2\left(\frac{A}{W} + 1\right) \qquad (6)$$

$$MT = 0.4366 + 0.0008\ log_2\left(\frac{A}{W} + 1\right) \qquad (7)$$

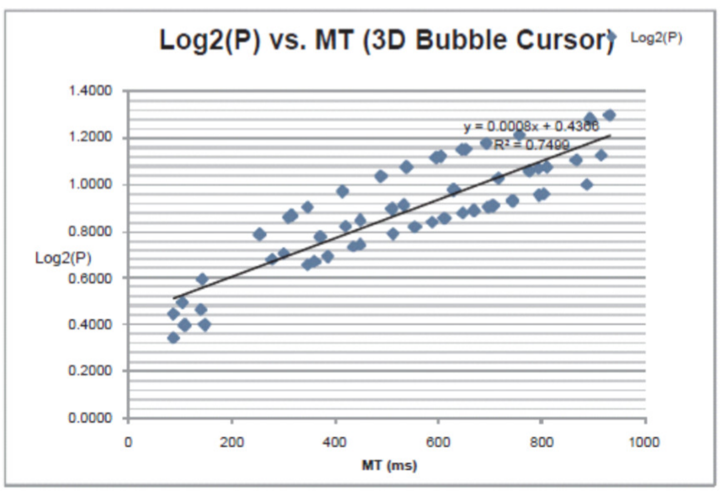

Figure 4.0: Regression Analysis for 3D Bubble Cursor

Vol 8. No. 2 Issue 2 – August, 2015
African Journal of Computing & ICT

4.1 Measurement of Dependent Variables

We measured two dependent variables from our experiment; (a) error rate and (b) trial completion time. Error rate is computed whenever a participant missed a target. Missing a target means a participant either clicked the wrong object or clicked an empty part of the screen. We compute the trial completion time by a measure of the time taken between to select one target to another. We carried out all analyzes of the dependent measures using repeated-measures analysis of variance.

4.2 Error Rate

The data that we captured during the experiments contain some outliers. We recorded 47 outliers in the captured datasets. This value represents more than three-inter-quartile ranges above the upper quartile in trial completion time. We removed these 47 outlier trials before we analyzed the error rates of the experiment. Figure 5.0 shows the total experiment completion time measure against the various techniques. In the three input techniques, results show that the total average completion time for point cursor techniques was heist at all time. Figure 6.0 shows the captured error rates in our experiment. We used the Index of Difficulty (IoD) parameter to determine the error index. At all times, IoD for the bubble cursor technique was less than 1.0.

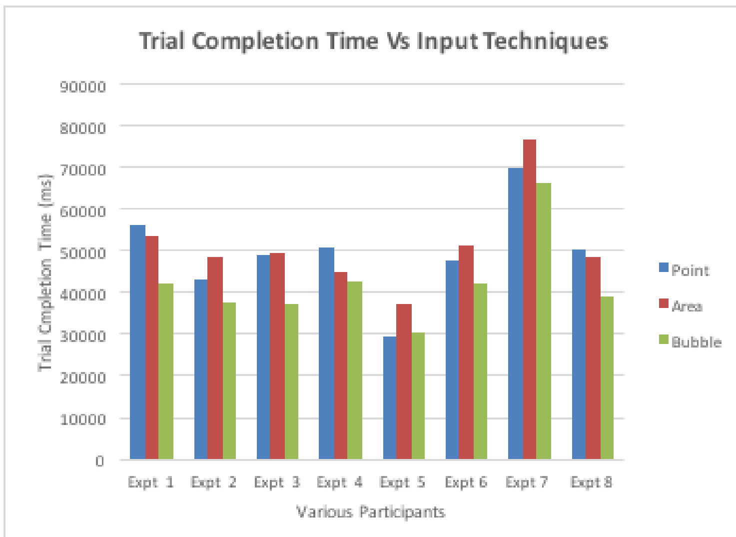

Figure 5.0: Experiment Completion Time Vs. Input Techniques

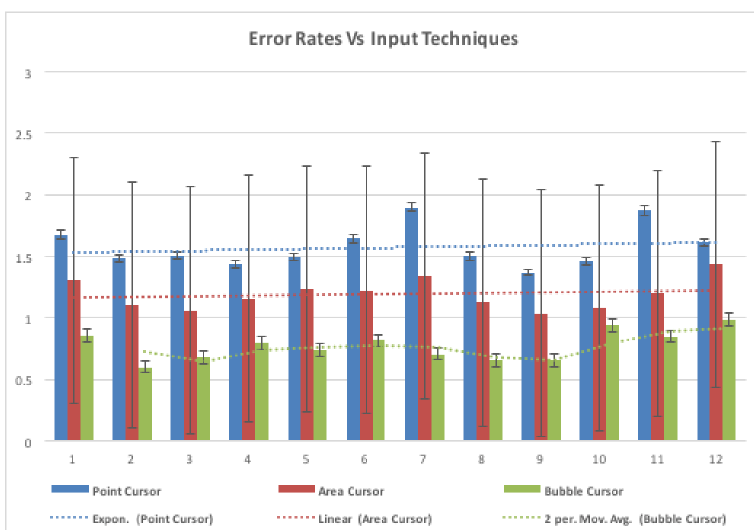

Figure 6.0: Error Rates Vs. Input Techniques

Vol 8. No. 2 Issue 2 – August, 2015
African Journal of Computing & ICT

 - ISSN 2006-1781
www.ajocict.net

Table 2: Paired Sample Test–Point Cursor (D33-D22).

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	D32	.8270	1000	.06684	.02114
	D22	.7510	1000	.12324	.03897

Paired Sample Correlation

		N	Correlation	Sig.
Pair 1	D32 & D22	1000	-.101	.782

Paired Samples Test

	Paired Differences							
				95% Confident Interval of the Difference				
	Mean	Std. Deviation	Std. Error Mean	Lower	Upper	t	df	Sig. (2-tailed)
Paired 1 D32-D22	-.06600	.14600	.04617	-.03844	.17044	1.430	999	.187

Table 4: Paired Sample–for Bubble Cursor (D33-D21).

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	D31	.8013	1000	.08045	.02734
	D21	.8055	1000	.07522	.02379

Paired Sample Correlation

		N	Correlation	Sig.
Pair 1	D31 & D21	1000	.08645	.02734

Paired Samples Test

	Paired Differences							
				95% Confident Interval of the Difference				
	Mean	Std. Deviation	Std. Error Mean	Lower	Upper	t	df	Sig. (2-tailed)
Paired 1 D31-D21	-.00371	.08872	.02806	-.06718	.05976	-.132	999	.898

African Journal of Computing & ICT

www.ajocict.net

We show statistical inference for the various input techniques in Tables 2, 3, and 4. There are significant differences between the techniques used in our experiments.

5.0 CONCLUSION

In this paper, we presented an environment based assessment for input techniques using the point cursor, the area cursor, and the bubble cursor. We reported that bubble cursor demonstrated a high response to changes in environment even at 0.1% significance. We obtained three mathematical equations for MT from our analysis. Equation (5), Equation (6), and Equation (7) for the three input techniques under study. Our results are comparable with the standard Fitts' law model for manipulating the human motor system. We have hence extended the Fitts' idea to select objects in 3D environment. More rigorous experiments are expected to yield results where error rate will be significantly negligible.

African Journal of Computing & ICT

www.ajocict.net

REFERENCES

[1] R. Balakrishnan. "Beating" Fitts' Law: Virtual Enhancements for Pointing Facilitation. International Journal of Human-Computer Studies, 67(6):857–874, 2004.

[2] P. M. Fitts. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. Journal of Experimental Psychology, 47:381– 391, June 1954.

[3] Y. Guiard. The problem of consistency in the design of Fitts' law experiments: consider either target distance and width or movement form and scale. In Proceedings of the 27th international conference on Human factors in computing systems, CHI '09, pages 1809–1818, New York, NY, USA, 2009. ACM.

[4] M. Hertzum and K. Hornbaek. Input Techniques that Dynamically Change their Cursor Activation Area: A Comparison of Bubble and Cell Cursors. Int. J. Human Computer Stud., 65:833–851, October 2007.

[5] S. C. Seow. Information Theoretic Models of HCI: A Comparison of the Hick-Hyman Law and Fitts' Law. Human Computer Interact., 20:315–352, September 2005.

[6] R. W. Soukoreff, J. Zhao, and X. Ren. The Entropy of a Rapid Aimed Movement: Fitts' Index of Difficulty Versus Shannon's Entropy. In Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part IV, INTERACT'11, pages 222–239, Berlin, Heidelberg, 2011. Springer-Verlag.

[7] S. Vetter, J. Bu¨tzler, N. Jochems, and C. M. Schlick. Fitts' Law in Bivariate Pointing on Large Touch Screens: Age-differentiated Analysis of Motion Angle Effects on Movement Times and Error Rates. In Proceedings of the 6th international conference on Universal access in human-computer interaction: Users diversity - Volume Part II, UAHCI'11, pages 620–628, Berlin, Heidelberg, 2011. Springer-Verlag.

[8] J. O. Wobbrock, E. Cutrell, S. Harada, and I. S. MacKenzie. An Error Model for Pointing Based on Fitts' Law. In Proceedings of ACM CHI 2008 Conference on Human Factors in Computing Systems, pages 1613–1622, 2008.

[9] T. Grossman and R. Balakrishnan. The Volume cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area. In CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems, pages 281–290, 2005.

[10] R. L. Potter, L. J. Weldon, and B. Shneiderman. Improving the Accuracy of Touch Screens: An Experimental Evaluation of Three Strategies. In CHI '88: Proceedings of the SIGCHI conference on Human factors in computing systems, pages 27–32, 1988.

[11] A. Sears and B. Shneiderman. High Precision Touchscreens: Design Strategies and Comparisons with a Mouse. International Journal of Man-Machine Studies, 34:593–613, 1991.

[12] D. Vogel and P. Baudisch. Shift: A Technique for Operating Pen-based Interfaces Using Touch. In CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems, pages 657–666, 2007.

[13] Y. Guiard, R. Blanch, and M. Beaudouin-Lafon, Object Pointing: A Complement to Bitmap Pointing in GUIs. In Proceedings of Graphics Interface, pp. 9–16, 2004.