

Using Kinect for 2D and 3D Pointing Tasks: Performance Evaluation

Alexandros Pino¹, Evangelos Tzemis^{2,3},
Nikolaos Ioannou², and Georgios Kouroupetroglou^{1,2}

¹ National and Kapodistrian University of Athens,
Accessibility Unit for Students with Disabilities, Athens, Greece

² National and Kapodistrian University of Athens,
Department of Informatics and Telecommunications, Athens, Greece
{pino, sdi0700116, koupe}@di.uoa.gr

³ University of Copenhagen, Department of Computer Science, Copenhagen, Denmark
khh107@alumni.ku.dk

Abstract. We present a study to comparatively evaluate the performance of computer-based 2D and 3D pointing tasks. In our experiments, based on the ISO 9241-9 standard methodology, a Microsoft Kinect device and a mouse were used by seven participants. For the 3D experiments we introduced a novel experiment layout, supplementing the ISO. We examine the pointing devices' conformance to Fitts' law and we measure a number of extra parameters that describe more accurately the cursor movement trajectories. Throughput, measured in bits per second is the most important performance measure. For the 2D tasks using Microsoft Kinect, Throughput is almost 39% lower than using the mouse, Target Re-Entry is 10 times up and Missed Clicks count is almost 50% higher. However, for the 3D tasks the mouse has a 9% lower Throughput than the Kinect, while Target Re-Entry and Missed Clicks are almost identical. Our results are also compared to older studies, and we finally show that the Kinect, operated by the user's hand and voice, is a suitable and effective input method for pointing and clicking, especially in 3D tasks.

Keywords: Fitts' law, 3D pointing, ISO 9241-9, Microsoft Kinect, Gesture User Interface.

1 Introduction

Nowadays, low-cost handheld devices introduced along with widespread game consoles can also be used as input devices in general purpose Personal Computers (PCs). Kinect [1] is a motion sensing input device by Microsoft for the Xbox 360 video game console and Windows PCs. Based around a webcam-style add-on peripheral, it enables users to control and interact with the Xbox 360 or the PC without the need to touch a game controller, through a natural user interface using gestures and spoken commands. The Kinect sensor is a horizontal bar connected to a small base with a motorized pivot and is designed to be positioned lengthwise above or below the video

display. The device features an RGB camera, depth sensor and multi-array microphone, running proprietary software, which provide full-body 3D motion capture, facial recognition and voice recognition capabilities. The Microsoft Kinect sensor, in opposition to most other accelerometer-based devices of this domain, uses the depth camera to recognize dynamic gestures, and this is the reason that the user does not need to use any kind of remote control, apart from his hands.

Although accelerometer-based recognition of dynamic gestures has been investigated in numerous studies (for examples see [2-3]), there is not an extensive research field of vision-based devices [4]. A promising software implementation that tracks the 3D position, orientation, and full articulation of a human hand from marker-less visual observations was developed by Oikonomidis, Kyriazis, and Argyros [5-6].

The abbreviations 2D and 3D in our case, where the Graphical User Interface (GUI) is displayed on a two-dimensional monitor in both cases, would more accurately be described by the terms two-directional and three-directional, rather than two-dimensional and three-dimensional.

The point-and-click metaphor usually referred to as pointing or tapping, constitutes a fundamental task for most 2D and 3D GUIs in order users to perform an object selection operation. Typing, resizing, dragging, scrolling, as well as other GUI operations require pointing. In order to develop better pointing techniques we need to understand the human pointing behavior and motor control. Fitts' Law [7] can be used to:

- Model the way users perform target selection
- Measure the user's performance,
- Compare the user's performance amongst various input devices or the change of performance over time.

Fitts' law has been applied to 3D pointing tasks [8] as well as to the design of gesture-based pointing interactions [9]. The most common evaluation measures related to Fitts' law are speed and accuracy, which are both incorporated in throughput.

2 Methodology

Fitts [7] proposed a model for the tradeoff between accuracy and speed in human motor movements. He proposed to quantify a movement task's difficulty using information theory by the metric of "bits". According to Fitts, the *Movement Time (MT)* needed to hit a target must be linearly related to the *Index of Difficulty (ID)* of the task:

$$MT = a + (b \times ID) , \quad (1)$$

where a and b are constants determined through linear regression,

$$ID = \log_2 \left(\frac{D}{W} + 1 \right) , \quad (2)$$

and D and W are the target's *Distance* and *Width* respectively.

Fitts proposed to quantify the human rate of information processing in aimed movements using “bits per second” as units. Fitts named the measure “index of performance”; today it is more commonly known as *Throughput* (TP , in bits/s). Although different methods of calculating *Throughput* exist in the literature, the preferred method is that proposed by Fitts in 1954 [7]. The calculation involves a direct division of means: dividing ID (bits) by the mean MT (seconds), computed over a block of trials:

$$TP = \frac{ID_e}{MT} \quad (4)$$

The subscript e in ID_e reflects a small but important adjustment, which Fitts endorsed in a follow-up paper [10]. The “adjustment for accuracy” involves first computing the *effective target Width* (W_e) as

$$W_e = 4,133 \times SD_x \quad (5)$$

where SD_x is the observed standard deviation in a participant's selection coordinates over repeated trials with a particular D - W condition. Computed in this manner, W_e includes the spatial variability, or accuracy, in responses. In essence, it captures what a participant actually did, rather than what he or she was asked to do. This adjustment necessitates a similar adjustment to ID , yielding an *effective Index of Difficulty*:

$$ID_e = \log_2 \left(\frac{D}{W_e} + 1 \right) \quad (6)$$

Calculated using the adjustment for accuracy, TP is a human performance measure that embeds both the speed and accuracy of responses. TP is most useful as a dependent variable in factorial experiments using pointing devices or pointing techniques as independent variables.

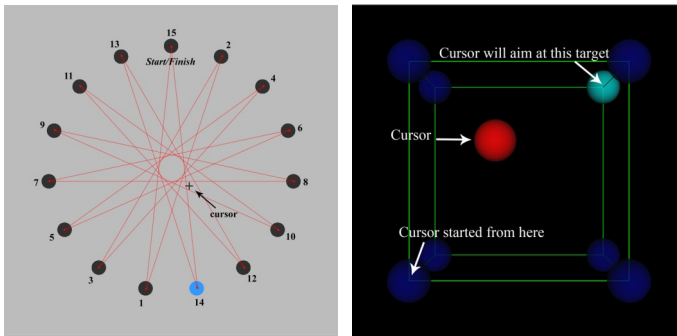
In order to evaluate the Kinect's conformance to Fitts' law as an input device, we used experimental software that we had previously designed and implemented [11], based on the ISO 9241-9 standard [12-13] that covers unidirectional and multidirectional user interaction. The ISO 9241-9 standard, describes a standardized procedure to evaluate the performance, comfort, and effort in using computer pointing devices; the use of the international standard grants the ability to better understand the experimental results, and to undertake comparisons between studies.

The GUIs of our 2D and 3D experiments are shown in Figure 1 left and right respectively. Users were asked to perform pointing tasks for 5 combinations of target distances and widths, hence 5 different ID s in increasing difficulty. For each ID they hit 15 targets in the 2D and 8 targets in the 3D experiment. Table 1 lists the ID s that were used for each of the 5 sessions. ID s were at a lower range in the 3D experiment because higher ID s resulted in non-displayable layouts either because the target distances were larger than the screen dimensions or because targets would be too small to see.

Table 1. Indexes of Difficulty (IDs) used for the 2D and the 3D experiments

Session	2D IDs	3D IDs
1 st	2.69	1.91
2 nd	3.19	2.35
3 rd	3.69	2.81
4 th	4.19	3.28
5 th	4.69	3.76

In the 2D case, the 15 circular targets are arranged in a circular layout (Figure 1, left). Initially the cursor is locked and unmovable on the center of the first target; the subject has to click in order to free the cursor and begin the experiment. Then the participant must move the cursor directly to the highlighted opposite target and click on it and so on clockwise. Each test block is complete when all 15 targets have been selected for 5 sessions giving a total of 75 trials per user. Circular targets and a cross-shaped cursor were used. The path the subject follows begins and ends at the top target. The lines in Figure 1 (left) indicate the ideal task path to alternating targets around the circle. Numbers indicate the succession of the targets to be hit. Software that captures the subjects' *MTs* and trajectory data, also graphically indicates which target the subject should proceed to (the lighter color target in the Figure). Figure 1 (left) illustrates the 4th ID experiment ($ID=4.19$)

**Fig. 1.** 2D (left) and 3D (right) pointing task screenshots

For the 3D experiment we have introduced a new layout, as the ISO 9241-9 did not contemplate for 3D interaction on the computer screen. We used 8 spherical targets placed at the vertices of a cube (Figure 1, right). Each task begins again with a click on the center of the first upper right target. Then the participant must move the cursor directly to the diagonally opposite target and click on it. After a successful trial the cursor teleports to another target that will become the beginning of the next route. The active target to be hit is highlighted every time. Each test block ends when all 8 equi-distance diagonal routes are successfully done (8 trials) and 5 sessions are run for different target radius and distance combinations (in total 5 different IDs) giving a total of 40 trials per user. The cursor was also a sphere (the red one). This way the user could perceive where the cursor is by its size in relation to the inner and outer

targets. The smaller the cursor the deeper it is inside the screen. When it is the same size as a target it means it is in the same z-axis level with it. Figure 1 (right) illustrates the experiment with $ID=3.28$.

We have to note that we disabled all windows cursor acceleration and accuracy enhancement options for the mouse, we used a 40×30 cm mouse pad, and the left mouse button for clicking (all users were right-handed). With the Kinect, for the 2D tests we acquired cursor movement coordinates by the depth camera data, after identifying the user's hand and taking as x and y the mean value of his hand's detected points. Movement in the z axis was not taken into account in 2D tests.

For the 3D experiments the difference was in that we used the mouse's scrolling wheel in order to move on the z axis, scrolling up to "go inwards" the screen and scrolling down in order to "come outwards". Regarding the Kinect, for 3D tests we used the z-axis data that were acquired through the Kinect's depth camera. We should also point out that in both experiments clicking was achieved using Kinect's microphones. The user had to produce a very short vowel phoneme in order for the click operation to be committed. He or she just had to say "Ah" for example in order to click.

Our analysis is based on the theory proposed by Fitts [7, 10] and MacKenzie et al. [14]. Specifically, we measured the following parameters (detailed definitions and formulas can be found in [11], [14], and [15]):

- *Throughput (TP)* in bits per second.
- Missed Clicks (MCL) scalar,
- Target Re-Entries (TRE) scalar,
- Task Axis Crossings (TAC) scalar,
- Movement Direction Changes (MDC) scalar,
- Orthogonal Direction Changes (ODC) scalar,
- Movement Variability (MV) in pixels,
- Movement Error (ME) in pixels,
- *Movement Offset (MO)* in pixels, and

Moreover, we have introduced a novel parameter **Distance Travelled (DT)**, defined as the distance travelled in pixels from the starting point to the successful click point inside the active target in each trial. It gives a sense of how close to the ideal path was the actual one. In a perfect trial where the cursor starts from the center of the starting target and the user clicks on the center of the active target, the *Distance Travelled* would be equal to the target's *Distance (D)*. Keates and al. [15]v also introduced a series of relative to the *DT* measures.

We developed the experimental application [16] as a Virtual Instrument using the LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) graphical programming environment by National Instruments [17]. We have tested Microsoft Kinect sensor as a gesture input device, and we developed appropriate software in LabVIEW for getting data in real time from its microphone array and the depth-camera.

The Microsoft Kinect was connected to a high-end PC using USB 2.0 communication. The computer was an Intel Core i7, 3.50 GHZ desktop with 8 GB of RAM

running MS-Windows 7 Professional and LabVIEW 2011. We used a 19" TFT monitor with 1280×1024 resolution, and a 1600 dpi wireless mouse.

Seven (7) participants, volunteered for the study. Their age range was from 22 to 55 years. No one had any kind of disability and they had no experience with the Microsoft Kinect sensor.

Participants were instructed to try to hit the active target in each trial as fast and as closer to the center they could. In both tests users were instructed not to stop on erroneous clicks and an auditory feedback was given in that case. Visual feedback was also given when the cursor was in the target, and both auditory and visual feedback was given on successful clicks. In order for the users to achieve the best possible results they were instructed to wear a headset in order to reduce background noise and listen better to the audio feedback. Moreover, the experiments were taking place in a dark environment (no other source of light except for the TFT screen brightness), to avoid distraction caused by other objects in the environment and also for the screen to be more visible. For the mouse experiment users were sitting on a chair having a desk with the mouse and mouse pad in front of them. For the Kinect experiment they were standing up with their right hand outstretched in front at their chest level. In both experiments participants were situated 2m away from the screen. Each task was explained and demonstrated to participants and a warm up session at the medium *ID* was given for each device and each mode (2D, 3D).

3 Results

Measurements of the *Movement Time (MT)* as a function of the *Index of Difficulty (ID)* for all the participants in 2D and 3D experiments using the mouse and Microsoft Kinect sensor are presented in Figure 2.

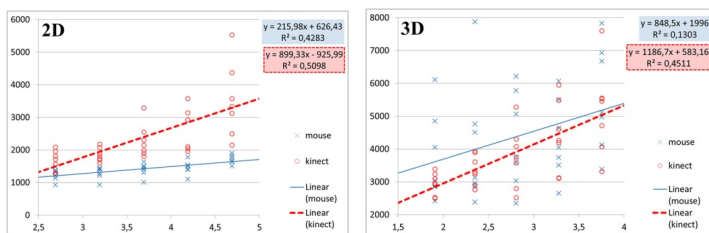


Fig. 2. Plots of the mean values of *Movement Time* for all trials (*MT* in milliseconds) as a function of the *Index of Difficulty (ID)* in bits) for all participants in 2D (left) and 3D (right) experiments using a Microsoft Kinect sensor (dashed lines) and the mouse (solid lines). Note: Value ranges in both axes differ between plots (axes are fitted to value ranged for better depiction).

The quantity *R*, called the *linear correlation coefficient*, measures the strength and the direction of the linear relationship between the two variables *ID* and *MT*. The value of the *correlation coefficient* is such that $-1 < R < +1$. Positive values indicate a relationship between the variables such that as values for *ID* increase, values for *MT*

also increase. Negative values indicate a relationship such that as values for *ID* increase, values for *MT* decrease. In our case *R* values were always positive. If there is no linear correlation or a weak linear correlation, *R* is close to 0. A value near zero means that there is a random, nonlinear relationship between the two variables. A perfect correlation of ± 1 occurs only when the data points all lie exactly on a straight line. A correlation greater than 0.8 is generally described as strong, whereas a correlation less than 0.5 is generally described as weak. In 2D experiments *R* was 0.65 for the mouse, and 0.71 for the Kinect. In 3D experiments *R* was 0.36 for the mouse and 0.67 for the Kinect.

The *coefficient of determination*, R^2 , is useful because it gives the proportion of the variance (fluctuation) of one variable that is predictable from the other variable. It is a measure that allows us to determine how certain one can be in making predictions from a certain model/graph, in our case Fitts' Law. The *coefficient of determination* is such that $0 < R^2 < 1$, and denotes the strength of the linear association between *ID* and *MT*. It represents the percent of the data that is the closest to the line of best fit. For example, the 2D Kinect line has $R^2=0.51$, which means that 51% of the total variation in *MT* can be explained by the linear relationship between *ID* and *MT* (as described by the regression equation). The other 49% of the total variation in *MT* remains unexplained. R^2 is a measure of how well the regression line represents the data, or how well our experiments' tasks comply with Fitts' Law. If the regression line passes exactly through every point on the scatter plot, Fitts' Law would be able to explain all of the variation. The further the line is away from the points, the less Fitts' Law is able to explain.

Table 2. Calculated parameters (means) of the cursor trajectory generated by the two input devices in 2D and 3D experiments.

		<i>TP</i>	<i>MCL</i>	<i>TRE</i>	<i>TAC</i>	<i>MDC</i>	<i>ODC</i>	<i>MV</i>	<i>ME</i>	<i>MO</i>	<i>DT</i>
2D	Mouse	3,45	0,30	0,07	2,62	31,50	1,24	11,39	12,95	3,73	569
	Kinect	2,10	0,28	0,69	4,85	18,82	5,28	16,97	15,80	3,41	905
3D	Mouse	0,96	0,53	0,18	-	41,10	12,37	78,30	109,10	-	1285
	Kinect	1,06	0,28	0,66	-	28,08	25,07	80,83	106,53	-	1563

Table 2 presents the results of the statistical analysis of all data from all users 2. We note that in 3D experiment *Task Axis Crossing* has no meaning because even if there is an axis connecting the two sphere targets, there is a too low possibility to actually cross it, and also that *Movement Error* is identical to *Movement Offset*.

Figure 3 illustrates typical sessions observed for the mouse (left) and the Kinect (right). It is obvious that for 2D tasks the mouse shows smoother and more accurate behavior. Kinect introduces tremor that typically appears in most people's hands; even when trembling is unnoticeable by eye, the plots of the pointing tasks reveal it.

For the specific sessions illustrated in Figure 3 we also give the mean values of the most important measures for comparison:

- Mouse: $MT = 1829$ ms, $TP = 3.27$ bits/s, $DT = 783.75$ pixels
- Kinect: $MT = 2148$ ms, $TP = 2.60$ bits/s, $DT = 865.61$ pixels

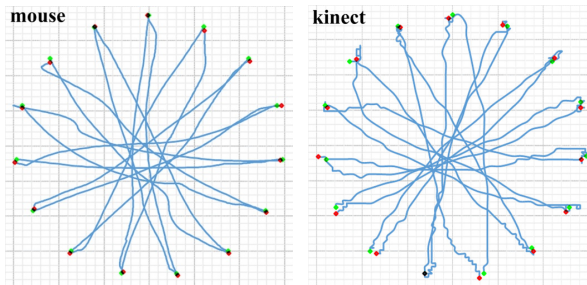


Fig. 3. Plots of the most difficult 2d session with $ID=4.69$: mouse (left), and Kinect (right)

We will briefly compare the results between this experiment and our previous one, which comprised the mouse and the Wiimote [16]. For the mouse, the basic difference is that in our previous experiments the mouse was enhanced by enabling the relevant windows settings; in Windows 7 the setting is “Enhance pointer precision”, and it accelerates the cursor when the user moves it fast (meaning he is “travelling” towards a target that is still far away), while it decelerates the cursor when the user is moving it slower (meaning he has reached close to the target and tries to click on the right spot). This way, in our previous experiments, users had some virtual aid when they were trying to hit the targets with the mouse. However, in order for the comparison between devices to be fair, we decided in the current experiment to use the mouse without any enhancements. We also adjusted the mouse speed in order to be comparable with the other device’s speed. Making ergonomic measurements we set up the experiment so that the whole range of the mouse’s movement (depending on the users’ arm and hand movement capabilities on the table) corresponds to the whole range of the Kinect movement (depending on the users’ arm and hand movement in the air) and both ranges have the same effect on the cursor’s movement on the screen.

In the previous experiment the mouse had a *Throughput* of 5.05 for the 2D and 1.71 for the 3D experiment, which were much better measurements than the 3.45 and 0.96 we respectively got now. The previous accuracy enhancement settings that were disabled now fully justify this difference.

Nevertheless, what is of higher interest and importance is the comparison of the Wiimote and Kinect performances: In our past experiments Wiimote gave a *Throughput* of 2.97, and 0.75 in 2D, and 3D tasks respectively. In the current experiment Kinect gave 2.10, and 1.06. This means that the Wiimote is better in 2D tasks and Kinect is better in 3D pointing.

4 Conclusion

Based on the ISO 9241-9 standard, we have presented an experimental study for the comparatively evaluation of computer-based 2D and 3D performance pointing tasks. A Microsoft Kinect device and a mouse were used by seven participants. For the 3D experiments we introduced a novel experiment layout, supplementing the ISO as well as the novel evaluation parameter **Distance Travelled (DT)**.

We conclude that for the 2D tasks using the Microsoft Kinect sensor, *Throughput*, is 39% lower than using the mouse, *Missed Click* count almost the same. However, for the 3D tasks using the Microsoft Kinect sensor, *Throughput* is 9.7% higher than using the mouse, while *Missed Clicks* are considerably lower.

Furthermore, Figure 2 shows that the fitting line *coefficient of determination (R^2)*, which reflects the reliability of the linear relationship between *MT* and *ID* values and, therefore, the compliance to Fitts' law, is slightly higher for the Kinect device than the mouse for the 2D experiment. When it comes to the 3D experiment, we can definitely observe that the compliance with Fitts' law is now much higher for the Kinect than the mouse.

Finally, we round off that the Microsoft Kinect device was proven to be a slower and harder to use input device for the 2D pointing task compared to the mouse. However 3D tests show that the condition is reversed as far as speed and accuracy of the Kinect are concerned. Kinect works better than the mouse in 3D. However, we can argue that both the mouse and the Kinect had too low *TP* as 3D pointing devices, which is partly justified by the fact that all users had no previous experience of any kind of 3D interaction.

Future work will include the involvement of more users in the experiments, also disabled users, research on how performance changes over time (i.e., familiarization with the Kinect and performance improvement), and introduction of new trajectory measures for 3D tasks in spherical coordinates.

Acknowledgments. The work described in this paper has been funded by the Special Account for Research Grants of the National and Kapodistrian University of Athens.

References

1. Microsoft: Kinect - Xbox.com. Xbox 360+Kinect homepage, <http://www.xbox.com/en-US/kinect>
2. Kela, J., Korpipää, P., Mäntyjärvi, J., Kallio, S., Savino, G., Jozzo, L., Marca, D.: Accelerometer-Based Gesture Control for a Design Environment. *Pers. Ubiquit. Comput.* 10(5), 285–299 (2006)
3. Kratz, S., Rohs, M.: The \$3 recognizer: Simple 3D Gesture Recognition on Mobile Devices. In: *IUI 2010, 15th International Conference on Intelligent User Interfaces*, pp. 419–420. ACM Press, New York (2010)
4. Marvel, J.A., Franaszek, M., Wilson, J., Hong, T.H.: Performance Evaluation of Consumer-Grade 3D Sensors for Static 6DOF Pose Estimation Systems. In: Tescher, A.G. (ed.) *Applications of Digital Image Processing XXXV, SPIE Optical Engineering and Applications Conference*, vol. 8499, article 849905. SPIE, Bellingham (2012)

5. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Efficient Model-Based 3D Tracking of Hand Articulations Using Kinect. In: Hoey, J., McKenna, S., Trucco, E. (eds.) 22nd British Machine Vision Conference, pp. 101.1–101.11. BMVA Press, Manchester (2011)
6. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Tracking the Articulated Motion of Two Strongly Interacting Hands. In: CVPR 2012, IEEE Conference on Computer Vision and Pattern Recognition, pp. 1862–1869. IEEE (2012)
7. Fitts, P.M.: The information capacity of the human motor system in controlling the amplitude of movement. *J. Exp. Psychol.* 47(6), 381–391 (1954)
8. Murata, A., Iwase, H.: Extending Fitts' law to a three-dimensional pointing task. *Hum. Mov. Sci.* 20, 791–805 (2001)
9. Foehrenbach, S., König, W.A., Gerken, J., Reiterer, H.: Tactile Feedback Enhanced Hand Gesture Interaction at Large, High-resolution Displays. *J. Visual Lang. Comput.* 20(5), 341–351 (2009)
10. Fitts, P.M., Peterson, J.R.: Information capacity of discrete motor responses. *J. Exp. Psychol.* 67(2), 103–112 (1964)
11. Pino, A., Kalogeros, E., Salemis, I., Kouroupetroglou, G.: Brain Computer Interface Cursor Measures for Motion-impaired and Able-bodied Users. In: Stephanidis, C. (ed.) *HCI International 2003. Universal Access in HCI: Inclusive Design in the Information Society*, vol. 4, pp. 1462–1466. Lawrence Erlbaum Associates, Mahwah (2003)
12. ISO 9241-9:2000: Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) - Part 9: Requirements for Non-keyboard Input Devices. ISO Standard (2000)
13. Soukoreff, W.R., MacKenzie, I.S.: Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI. *Int. J. Hum.-Comput. St.* 61(6), 751–789 (2004)
14. MacKenzie, I.S., Kauppinen, T., Silfverberg, M.: Accuracy measures for evaluating computer pointing devices. In: *CHI 2001, SIGCHI Conference on Human Factors in Computing Systems*, pp. 9–16. ACM Press, New York (2001)
15. Keates, S., Hwang, F., Langdon, P., Clarkson, J.P.: Cursor measures for motion-impaired computer users. In: *ASSETS 2002, the 5th International ACM Conference on Assistive Technologies*, pp. 135–142. ACM Press, New York (2002)
16. Kouroupetroglou, G., Pino, A., Balmpakakis, A., Chalastanis, D., Golematis, V., Ioannou, N., Koutsoumpas, I.: Using Wiimote for 2D and 3D Pointing Tasks: Gesture Performance Evaluation. In: Efthimiou, E., Kouroupetroglou, G., Fotinea, S.-E. (eds.) *GW 2011. LNCS (LNAI)*, vol. 7206, pp. 13–23. Springer, Heidelberg (2012)
17. National Instruments: NI LabVIEW – Improving the Productivity of Engineers and Scientists. LabVIEW System Design Software homepage, <http://www.ni.com/labview/>