Interaction with large displays in a public space using the Kinect sensor

Igor Duarte Cardoso
DETI / IEETA
Universidade de Aveiro
idc@ua.pt

Paulo Dias
DETI / IEETA
Universidade de Aveiro
paulo.dias@ua.pt

Beatriz Sousa Santos
DETI / IEETA
Universidade de Aveiro
bss@ua.pt

Resumo

Com a evolução da tecnologia, monitores de computador e televisores têm-se tornado cada vez mais finos e baratos, o que permitiu a sua colocação em muitos espaços públicos tais como aeroportos, halls de entrada de edifícios, etc. para apresentação de informação a quem frequente esses espaços. No entanto, a utilização destes ecrãs apresenta alguns desafios sobretudo quanto à forma como a informação é apresentada e como se pode interagir com a mesma. Em trabalhos anteriores estudámos a possibilidade de utilizar telemóveis como dispositivos de interação com estes ecrãs; neste artigo, apresentamos uma evolução que utiliza o Kinect como dispositivo de entrada permitindo um controlo da informação apresentada no ecrã sem recurso a hardware adicional. São descritas as modificações ao protótipo original (desenhado para telemóveis) e duas propostas de interação diferentes: "Gesture-Based Interaction" e "Pointer-Based Interaction". Finalmente, apresentam-se os resultados dos testes efectuados para avaliar as escolhas realizadas e testar a viabilidade do sistema para interagir com ecrãs em espaços públicos.

Abstract

As technology evolves, computer and television screens have become increasingly thinner and cheaper. This has allowed these screens to be used in public locations such as airports, lobbies of buildings, etc., presenting relevant information to the people passing by. The use of these screens, however, presents a challenge in how information is displayed and how people can interact with it. Often, they do not provide any possible interaction and displayed information cannot be controlled at all by the users. In previous work, we studied the possibility to use mobile phones as interaction devices for these large public displays. In this paper we present an evolution using Kinect to allow a device free interaction avoiding the need to use any additional hardware. We present the modifications to the original prototype (for mobile phones) and two different interaction proposals: "Gesture-Based Interaction" and "Pointer-Based Interaction". Finally, we evaluated our choices and tested the viability of the system to interact with large displays in a public space.

Keywords

Kinect, large displays, gesture recognition, natural interaction, user interfaces

1. INTRODUCTION

With the reduction of size, weight and cost of computer and television screens, it is now common to find large displays available in public spaces, such as airports, train stations, waiting rooms, museums, etc. These displays often do not provide any interactivity and users cannot control in any way the displayed information. At the same time, major new products in the field of human-computer interaction have started emerging, allowing more natural user interfaces. Examples of such are innovative controllers, like the Wii Remote, Playstation Move, Kinect and other less known devices, that have been widely accepted by the community [Takala12]. Moreover, many types of sensors deeply integrated into smartphones, tablets and the like, enable richer user experiences and interaction.

In previous work, we studied the possibility of using mobile phones to allow user to interact with the content of large passive public displays using their own devices and with no need to buy additional hardware [Duarte11]. This solution, although interesting presents some limitations since it requires the user to download an application on their mobile devices and implies initial pairing steps for communication. A prototype called "DETI-Interact" was developed and is currently working in the lobby of the Department of Electronics, Telecommunications and Informatics of the University of Aveiro (see Figure 1). However, the number of users in this real scenario is reduced probably due to the initial configuration steps that most of them will not perform due to lack of trust (the application runs in their devices) and the time it takes to make the first configuration.

Figure 1: User interacting with DETI-Interact

To overcome the initial configuration problem, we propose to use an alternative interaction device that does not belong to the user and allows natural gesture interaction. Thus, DETI-Interact has been re-imagined to work with the Kinect sensor. The user interface and the architecture have been adapted to this new device, maintaining compatibility with Android Bluetooth-enabled devices (though only one the two: Kinect or Android, can be active at a given time).

2. RELATED WORK

Kinect is a motion sensing input device developed by Microsoft and launched on November 2010. It allows tracking of physical objects and full body skeletons in a 3-dimensional space. The Kinect sensor includes: an RGB camera, an infrared laser projector, a monochrome CMOS sensor, a microphone array, a 3-axis accelerometer and other components less relevant for the interaction. Initially launched for the game industry on the Xbox 360 platform, some software development kits (SDK) and several toolkits appeared in the community, allowing development of Kinect-powered applications for the computer. Some SDKs are: OpenNI [Ope10b], OpenKinect [Ope10a] and the CL NUI Platform [CLN10]. More recently, Microsoft itself launched the final version of the Kinect for Windows SDK [Kin11a].

Besides being used the game industry and in some research projects, for instance in aiding the visually impaired [Zöllner11], modernization of user interaction classes [Villaroman11] and several hobbyist projects, Kinect has also spawned interest in important consumer electronics brands such as Samsung, which has recently released new solutions using similar sensors to control TV sets [Sam12].

Previous works, more aimed at interaction with large displays or surfaces, have tested how appropriate some gestures can be using Kinect [Hespanhol12] [Reis12]. Another work has joined surface-based interaction with space-based interaction into one, as a continuum of interactivity where one can start and stop gestures or movements in any part of the space including the touch surface of the tabletop system used [Marquardt11].

3. DETI-INTERACT

The basic functionalities of DETI-Interact are navigation on the Faculty list, department schedules, and the world map. In the previous version [Duarte11], the only interaction method with the content was done by using a Bluetooth-enabled mobile device with touchscreen and accelerometer (running Android operating system).

The technologies used were based on .NET, with C# as the programming language, along with Windows Presentation Foundation (WPF) for a modern user interface, the Google Earth plugin to provide navigation in the world map (plus JavaScript for interacting with it) and Android SDK to develop the mobile application. The application also made use of internal web services from the Department (specifically, DSD) in order to obtain Faculty and schedules information as can be seen in Figure 2, depicting the general architecture.

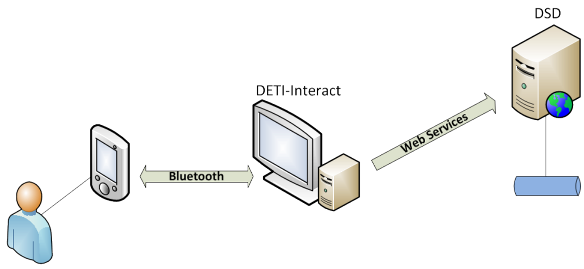

Figure 2: Previous general architecture

Now, DETI-Interact provides two new interaction methods that have been carefully developed, tweaked and finally submitted to usability tests. The goal is to find out which is the best interaction method and/or whether they can be merged into a single ”best of both worlds” interaction method that will finally be deployed in our Department.

3.1. Adaptation to Kinect

To provide support for Kinect interaction, the Kinect for Windows SDK was chosen, integrating more properly with the DETI-Interact implementation (based on Microsoft technology) and being capable and generally recommended for full body tracking [Takala12]. Also, a library was forked from the Kinect Toolbox [Kin11b] gesture recognition and improved to provide more actions and better suit the needs of DETI-Interact.

Using the library to recognize gestures, a list of the last captured frames is stored in what we call the gesture window. It has a limited number of frames and is used to calculate the variation of user movements. The only gesture detected, quickly moving the hand in one direction, is identified by comparing every element in the gesture window. The element is configurable to be any of the 20 joints detected by Kinect for Windows SDK. The parameters used to compare these elements and decide if a gesture has been found are: the minimum distance of the gesture, whether it was between a predefined interval of time, was it stable in its path, i.e, the movement approximately followed a straight line, and whether it did not invert direction. The direction axes supported by this gesture are the horizontal, vertical and longitudinal ones, relatively to the Kinect sensor.

Besides gesture detection, more types of actions have been added to this Kinect library, including one that periodically triggers when a user's joint is within a predefined range.

This adaptation to the Kinect sensor turned the architecture into what can be seen in Figure 3. Support for Kinect sensor was added without having to remove the Android one, made possible by developing the application in a way that generically supports any interaction device, having a middle layer to properly convert device-specific actions to UI-specific actions (enter, exit, zoom, and others), making the device abstract to DETI-Interact.

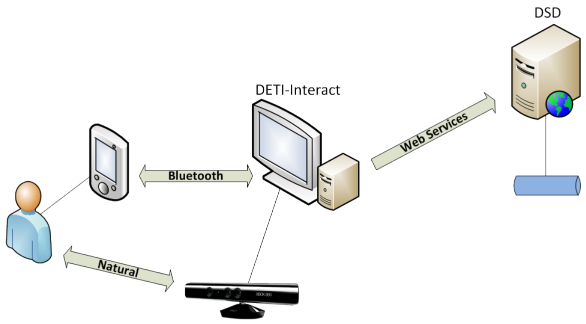

Figure 3: Current general architecture

3.2. User Interface

The overall appearance of DETI-Interact user interface has not changed much since the previous version, though some details have been tweaked specifically to improve interaction using the Kinect sensor. Overall, space has been optimized by removing padding and borders, font size has been increased on critical areas as well as the width of page tabs.

The user interface is based on pages that can be seen one at a time. They are arranged in an horizontal layout with tabs ("page tabs") at the right and left side which show the names of the pages they refer to. The major change from the previous version is the Gesture Tracker: a small "visor" that shows the relative position of the user's "navigation" hand (see Figure 4). This allows users to easily have feedback about their hand position while using the Gesture-Based Interaction without disturbing them too much. The last detected positions of the hand are shown in the Gesture Tracker as small circles.

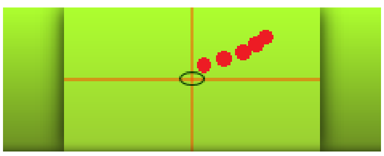

Figure 4: Gesture Tracker

The three basic functionalities are contained in three different pages. The title bar, at the top of every page, shows from left to right, the title, the Gesture Tracker, the current time and date. The user help information is shown at the bottom as in the previous version.

The first page, the Faculty viewer, presents a scrollable grid-based view on Faculty members, each shown in a business card fashion, containing the name, photo, office number, phone extension and webpage URL (Figure 5). For interaction and feedback purposes, four arrows are displayed around the grid.

Figure 5: Faculty viewer

The application allows the selection of a Faculty member to further open his/her webpage (Figure 6), using all the available space on the screen page. Depending on the interaction, an arrow is placed on the title bar (see Figure 9) providing the action to go back.

Figure 6: Faculty member webpage

Another different page is the Schedule viewer (see Figure 7), which allows navigating the Department schedule showing a full view of one timetable at a time.

The classes that are supposed to occur, as provided by the web services, are then arranged in the timetable with days of the week as rows and hours of the day as columns. At the top of the timetable are the Study Program and corresponding year selected.

As in the Faculty viewer, arrows are shown for interactivity and feedback purposes.

The final page is the world map (Figure 8). Although it has not been subject to formal usability tests because our main purpose was testing GBI against PBI in simple navigation contexts, some users actually interacted with it off the record. According to their feedback, interaction with the map seems to be an interesting experience. They have successfully found multiple locations on the map. In this version of DETI-Interact, the map has been max-

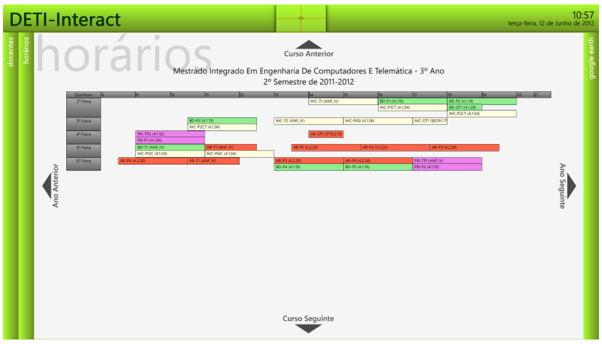

Figure 7: Schedule viewer

imized to fit all the space available on the screen page. The move around the map only one hand is required. Unlike [Stannus11], which provides a more intuitive yet complex way of interacting with the globe by having the users first grab it via closing their hands, we have a simplified implementation where latitude, longitude and altitude variation speeds are linearly controlled by the hand's position in the space. For instance, the farther away from the sensor, the faster the altitude increases (or the map zooms out). The same applies for latitude and longitude where moving the hand to the right rotates the earth towards east and moving the hand to the top rotates the earth towards north, and vice-versa. There is no need to first grab the globe nor distinguish between panning and zooming in an explicit way, in terms of individual gestures used.

Figure 8: World map

3.3. Interaction Methods

As mentioned earlier, there are two new interaction methods in the current DETI-Interact: Gesture-Based Interaction (GBI) and Pointer-Based Interaction (PBI), both using the Kinect sensor.

The most important operations that can be done in DETI-Interact and its two main pages (Faculty viewer and Schedule viewer) are the following:

[Because there is always not much to do in a DI page, the gestures can be reused better without being ambiguous, and cite Hespanhol here]

- Alternate between pages
- Find a Faculty member, i.e, navigate until selection is possible
- Select Faculty member to open his/her webpage
- Go back from the webpage, i.e, return to the Faculty member list
- Select a specific timetable

The actions detected by the developed Kinect library are:

- Swipe

  Consists in the gesture of quickly moving the hand from one side to the other with enough amplitude. Vertically, when the hand moves from top to bottom, it is named a "Swipe Down". In the opposite direction, it is a "Swipe Up". Horizontally, when the hand moves from left to right, it is named a "Swipe Right". In the opposite direction, it is a "Swipe Left".

  This action is highly configurable with a variety of attributes like: sleep time (duration of time the system will reject new gestures after a successful one), length of the gesture, maximum time of the gesture, tolerance for unstable gestures and minimum speed of the gesture (by the relative increase in distance).

- Push/Pull

  Consists in the gesture of quickly moving the hand towards or away from the sensor. The concept of "Push" is the same as in [Hespanhol12] "Pushing" but we also provide the opposite, called "Pull". When the hand moves towards, it is named "Push". In the opposite direction (away from), it is "Pull". Push and Pull are exactly like a Swipe but in the longitudinal direction.

- Stay

  This action just acquires the relative user's hand position with a great number of samples per second, very useful for showing the position in Gesture Tracker or for scrolling inside a page. Although it isn't exactly a gesture, but rather a constant refresh of the user's hand position, it is normalized and highly configurable. For example, it is possible to set the amount of samples per second and a virtual central point in the Z axis of the space, which is currently useful for zooming in Google Earth.

- Timed Stay

  Partly based on the "dwelling" gesture by [Hespanhol12], "Timed Stay" works by hovering a zone for a specified amount of time to activate the corresponding action. It differs from "dwelling" because if the hand is within a specified range near each of the four edges of the field of view, the action will be periodically triggered (for instance, each second) and coupled with the actual direction, instead of triggering the action in any part of the screen.

These actions can be mapped to any hand and are the base for both methods of interaction.

3.3.1. Gesture-Based Interaction

The Gesture-Based Interaction (GBI) consists of simple hand gestures and positions, all done with only one hand. In the usability tests, the right hand was chosen only because the majority of users are right.

For the operation of alternating between pages, a Swipe action is done. When doing Swipe Right, the next page at the right expands and replaces the current one. In the opposite direction (”Swipe left”), the contrary occurs.

In order to find a Faculty member, users first need to use the Push action to activate the selection mode. In this mode, users navigate through the list using the Timed Stay action. This selects the next Faculty member in any of the directions (one per second, as configured for the usability tests) until the one sought is found. The Faculty member that is currently selected will have his/her ”business card” color changed to provide feedback. Also, the correct arrow (there are four arrows around the list) will change color each time the action triggers. Users can also make use of the Stay action to quickly scroll the list, only doing a Push in the end.

For the operation of viewing the webpage of a Faculty member, first users should already have a Faculty member selected. Moreover, they just need to make another Push and the webpage will automatically be shown in the user interface.

To go back from a webpage, users do a Pull action (the opposite of Push). When users pull from the webpage, they will be taken back to the Faculty member's list. Another Pull and the Faculty member's list is unfocused, again allowing to switch among DETI-Interact pages.

Finally, to select a timetable, users must first use Push to initiate the selection mode (the same as the Faculty member's list), and then Swipe in any of the four directions until the wanted schedule is found. A Swipe Right will slide the timetable to the next on the right (next year). A Swipe Left will do the opposite (previous year). A Swipe Up will slide the timetable to the next above (previous Study Program). A Swipe Down will do the opposite (next Study Program). Each time a timetable is changed, the proper arrow around will change its color to further emphasize the direction taken. To exit selection and be able to change pages again, another Pull is required.

The Push/Pull actions are always used to enter and exit pages. Because the main pages of DETI-Interact can be swapped using the Swipe gesture, there needs to be a way of avoiding accidental changes of those pages. So, when users are interested in interacting with content, they should first activate that content using the Push action. When finished, they use the Pull action to go back to page selection.

3.3.2. Pointer-Based Interaction

The Pointer-Based Interaction (PBI) is a different approach that controls a screen cursor (pointer) with one hand and selects/enters interface elements with the other hand. We chose to use different hands for selection and navigation because using a single hand for both has precision problems during the Push action. The right hand was chosen for controlling the cursor given that the majority of computer users is trained to use the mouse with this hand.

First, the user interface provides a noticeable cursor (a large green circle) so users know exactly where they are, even when a few meters away from the screen.

For the operation of alternating between pages, users need to hover the page tab desired (the tabs have the name of the page on it) and then select that tab using the other hand by quickly pressing it towards the sensor, through the Push action, simulating a mouse click.

In order to find a Faculty member in the list, users first scroll until the zone where the Faculty member is, made possible with the implicit Stay action. When users find out where the Faculty member is, they can hover it and the UI will provide feedback by changing the color of the Faculty member's ”business card”.

For the operation of viewing the webpage of a Faculty member, users simply select (Push) the Faculty member that is below the cursor. The webpage will automatically be displayed.

To go back from a webpage, users can use a specific interface element that only appears in Pointer-Based Interaction, which is the back button (see Figure 9). By selecting it with the hand, the Faculty viewer will be visible again. Alternatively, users can select another page if they are not interested in the Faculty viewer again, though when coming back to the viewer the webpage will still be visible.

Figure 9: Back button

Finally, to select a timetable, the arrow for the desired direction needs to be hovered and selected with a Push. The left arrow shows the timetable for the next year, while the right arrow shows the previous year. The up arrow shows the timetable for the previous Study Program, while the down arrow shows the next Study Program year.

4. USABILITY TESTS

With both interaction methods ready and sufficiently robust, usability tests were performed to evaluate the whole interaction, as can be seen in the Figure 10.

Unlike previous works that usually test their systems in isolated spaces in order to focus exclusively in the study of interaction methods, like [Hespanhol12], we have proceeded to usability tests in a public space. However, the tests occurred when the flow of people actually present in the public space, our department, were low but noticeable. Thus, we avoid having the comparison of the interaction methods efficiency compromised due to too many distractions.

Always less than half a dozen people passed by during the approximate 15 minutes of usability tests.

The tests consisted in four specific and realistic tasks to be carried out by the user:

1. Find the phone extension of a given Faculty member
2. Consult the classroom of a given class at a given time
3. Check the address of a given Faculty member
4. Find a specific timetable of a Study Program and finish the test

Figure 10: User interacting with the new DETI-Interact

The configuration parameters of the Kinect actions, especially for the gestures, were tweaked empirically by the developers to be at least acceptable in the usability tests.

Although every user tested both interaction methods, approximately half of them started with the Gesture-Based Interaction and the other half with the Pointer-Based Interaction to minimize learning effects.

The possible actions and operations were explained in general before the start of the test and some hints were given during the test in a way that users got to know what they were supposed to do without knowing the feeling of doing so.

In total, there were 17 participants in the usability tests, 8 of them started with the Gesture-Based Interaction while the other 9 started with the Pointer-based Interaction. These participants were all students of informatics-related courses so they were already trained with standard computer interaction. Of all participants, 15 had used touchscreen devices (like smartphones) before, while the other 2 didn't. Finally, 15 were male while 2 of them were female.

After they finished each task, they were asked to give a score for the difficulty, while their impression of the system was still fresh.

At the end of the test, each participant was given a questionnaire so they could evaluate the user interface and interaction methods in more detail, while also allowing them to write their own feedback. A copy can be accessed through the URL: http://goo.gl/uq67M (only available in portuguese).

The user questionnaire consisted in three types of questions: true or false, questions in the Likert scale and open ended questions.

In true or false questions, users were asked personal questions: if they had experience with touchscreen, accelerometer or digital compass powered devices, experience with devices like Kinect, Wii Remote or Playstation Move and what if they had already used the previous DETI-Interact. The results, as shown in Figure 11, indicate that nowadays almost every informatics student has experience with tactile or accelerometer-enabled devices. Also, it shows that more than half have had contact with the Kinect sensor or related devices.

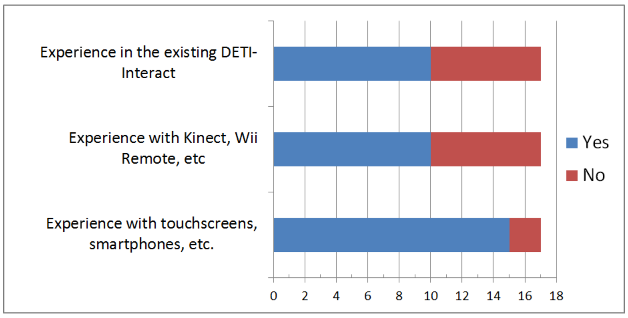

Figure 11: Personal true/false results

For the questions in Likert scale, they were asked general usability questions for each interaction method, for instance whether they felt the navigation was pleasant, intuitive or required much training. They were also asked about the difficulty of each action for each interaction method. Finally, they answered about the general satisfaction in each interaction method.

In the open ended questions, they were invited to comment on the set of gestures used, whether they liked them, and to propose new ideas.

It was observed that when a person is wearing a thick, cotton coats, the precision of skeleton detection decreases thus negatively affecting the interaction.

Some users commented that Pointer-Based Interaction was good, especially for selecting Faculty members, yet most of them took longer using this method of interaction and made negative comments due to limb occlusion from Kinect. When selecting something at the left side of the screen using the Pointer-Based Interaction, the right-arm would go in front of the left-arm, creating additional difficulty to the proper full skeleton detection necessary by the Microsoft Kinect SDK.

Another skeleton detection problem occurs when there are people passing by. As this project is supposed to be deployed in a public space, it is a real concern to improve the detection efficiency and robustness when more people are in front of the sensor.

Other common complaints and errors of interaction include: the non intentional input of gestures, when someone repositioned himself/herself more abruptly; the nonresponsive time right after a successful gesture, consequence of the sleep time implemented that prevents the movement of returning from a swipe to be detected as another swipe;

We also observed that 15 of the 17 users intuitively wanted the swipe to be in the opposite direction, i.e, instead of moving from left to right to show something that was at the right side, they wanted to virtually grab the page with their hand and slide it from right to left in order to view the page that was hidden at the right side. The direction we chose seemed appropriate due to the lack of a surface to grab the DETI-Interact pages. We think that the different direction users tended to take might be partly related to their previous experience with touch-devices, like smartphones. If our system was based on a continuum interaction space, as introduced by [Marquardt11], and the swipe started by touching the surface and swiping it to the desired direction (thus grabbing and dragging the page), we argue that we and every user would agree on the same direction.

In general, the Pointer-Based Interaction was worse, as indicated by the user satisfaction (see Figure 12), the median difficulty of actions (see Figure 13) and the median difficulty felt by the users in performing each task (see Figure 14); however, user performance was better with Pointer-Based Interaction in task 3: "Check the address of a given Faculty member" as can be seen in Figure 15. What we observed is in agreement with these charts. As it was expected, PBI is especially difficult when selecting areas of the screen that are close to the edges, especially at the edge opposite to the cursor hand (in our tests, the left edge of the screen) since the arms get interlaced and proper detection becomes difficult. However, when selecting something in the middle of the screen, PBI works well and is faster, the greatest difficulty being the successful detection of a push gesture that triggers a click. Such is the case of selecting a Faculty member to view the corresponding webpage and address, as in task 3. The feedback provided by the users also supports this. They presented some alternative ways of executing the clicks that do not have the problem of interlaced arms. One way would be to close the selection hand (left hand) to trigger a click, which cannot be done in Kinect for Windows SDK using skeleton frames due to its lack of precision. Another way would be to nod the head down for the same effect, thus only requiring one hand.

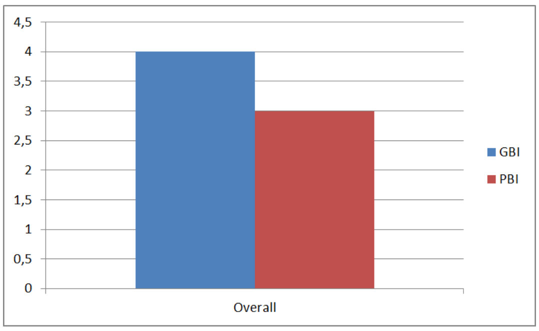

Figure 12: Overall satisfaction

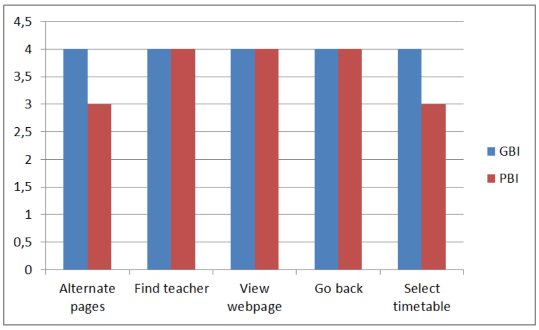

Figure 13: Median difficulty of actions

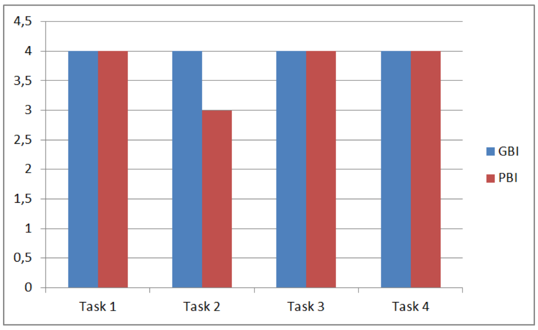

Figure 14: Median difficulty of tasks

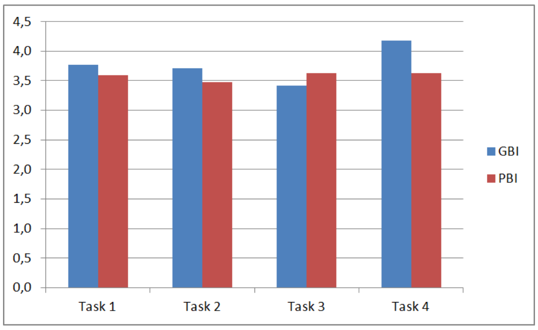

Figure 15: Average difficulty of tasks

5. CONCLUSIONS AND FUTURE WORK

Results have shown that for most of the UI elements, users preferred the Gesture-Based Interaction over the Pointer-Based Interaction. However, when users are presented with multiple selectable elements spread along the screen, they want to select just the one they intend to, without having to go through all the others. Therefore, Pointer-Based Interaction seems more appropriate in this case.

On the other hand, when presented with a simple screen with few selectable information, users prefer to swipe in order to select the next piece of information they are looking for, instead of "pushing" specific buttons to navigate. In this class, Gesture-Based Interaction simplifies and is overall better.

Given the mixed interaction advantages, a hybrid solution seems the most appropriate for a final implementation scenario, as we already expected.

During the usability tests, we noticed that most users needed some help in the beginning to interact with the system. However, after a short period of time, they were able to use the interface without major problems. Since normal users are not supposed to have any training, it is important to include help and other affordances.

Based on the results of this work, we plan to develop a new version of DETI-Interact with the hybrid solution, for deployment on the lobby of our Department and further logging/testing. Further improvements may include activation methods to ensure that the system will respond only to interested users. A possible solution may use a camera (already available in the Kinect sensor) and face recognition/tracking software to detect users staring at the monitor.

6. ACKNOWLEDGEMENTS

This Research was partly is funded by FEDER through the Operational Program Competitiveness Factors - COMPETE and by National Funds through FCT - Foundation for Science and Technology in the context of the project FCOMP-01-0124-FEDER-022682 (FCT reference PEst-C/EEI/UI0127/2011)

Our thanks also to all the students that participated in the evaluation providing valuable contributions.

7. REFERENCES

[CLN10] CL NUI Platform. `http://codelaboratories.com/nui`, 2010. [Online; accessed 15/06/2012].

[Duarte11] Filipe Duarte, Paulo Dias, and Beatriz Sousa Santos. Deti-interact: Interacting with public displays through mobile phones. In 6ª *Conferência Ibérica de Sistemas e Tecnologias de Informação (CISTI 2011), Workshop on Information Systems for Interactive Spaces*, pages 787–792. IEEE, 15-18 June 2011.

[Hespanhol12] Luke Hespanhol, Martin Tomitsch, Kazjon Grace, Anthony Collins, and Judy Kay. Investigating intuitiveness and effectiveness of gestures for free spatial interaction with large displays. In *Proceedings of the 2012 International Symposium on Pervasive Displays*, PerDis '12, pages 6:1–6:6, New York, NY, USA, 2012. ACM.

[Kin11a] Kinect for Windows SDK. `http://www.microsoft.com/en-us/kinectforwindows/`, 2011. [Online; accessed 15/06/2012].

[Kin11b] Kinect Toolbox. `http://kinecttoolbox.codeplex.com/`, 2011. [Online; accessed 15/06/2012].

[Marquardt11] Nicolai Marquardt, Ricardo Jota, Saul Greenberg, and Joaquim A. Jorge. The continuous interaction space: interaction techniques unifying touch and gesture on and above a digital surface. In *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part III*, INTERACT'11, pages 461–476, Berlin, Heidelberg, 2011. Springer-Verlag.

[Ope10a] OpenKinect. `http://openkinect.org/wiki/Main_Page`, 2010. [Online; accessed 15/06/2012].

[Ope10b] OpenNI. `http://openni.org/`, 2010. [Online; accessed 15/06/2012].

[Reis12] Bernardo Reis, João Marcelo Teixeira, Felipe Breyer, Luis Arthur Vasconcelos, Aline Cavalcanti, André Ferreira, and Judith Kelner. Increasing kinect application development productivity by an enhanced hardware abstraction. In *Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems*, EICS '12, pages 5–14, New York, NY, USA, 2012. ACM.

[Sam12] Samsung ES8000 LED HDTV is Smarter Than Most Phones. `http://goo.gl/Wi497`, 2012. [Online; accessed 15/06/2012].

[Stannus11] Simon Stannus, Daniel Rolf, Arko Lucieer, and Winyu Chinthammit. Gestural navigation in google earth. In *Proceedings of the 23rd Australian Computer-Human Interaction Conference*, OzCHI '11, pages 269–272, New York, NY, USA, 2011. ACM.

[Takala12] Tuukka M. Takala, Paivi Rauhamaa, and Tapio Takala. Survey of 3dui applications and development challenges. In Mark Billinghurst, Joseph J. LaViola Jr., and Anatole Lécuyer, editors, *3DUI*, pages 89–96. IEEE, 2012.

[Villaroman11] Norman Villaroman, Dale Rowe, and Bret Swan. Teaching natural user interaction using openni and the microsoft kinect sensor. In *Proceedings of the 2011 conference on Information technology education*, SIGITE '11, pages 227–232, New York, NY, USA, 2011. ACM.

[Zöllner11] Michael Zöllner, Stephan Huber, Hans-Christian Jetter, and Harald Reiterer. Navi: a proof-of-concept of a mobile navigational aid for visually impaired based on the microsoft kinect. In *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part IV*, INTERACT'11, pages 584–587, Berlin, Heidelberg, 2011. Springer-Verlag.