Interactive Public Displays: A Gesture-Based Proposal using Kinect

Thiago Motta, Luciana Nedel
Institute of Informatics
Federal University of Rio Grande do Sul (UFRGS)
Porto Alegre, Brazil
E-mail: {tsmotta,nedel}@inf.ufrgs.br

***Abstract*—The use of interactive public displays in urban spaces is increasing very fast, and new setups to support these displays are still being explored. In this paper we introduce a gesture based input technique to allow the interaction with public displays avoiding the use of any device attached to the user body. The gestures supported provide navigation, selection and manipulation of objects, as well as panning and zooming on the screen. In order to evaluate how robust the system is in a real public scenario, criteria that could interfere on the interactive task are evaluated, as the amount of brightness in the environment, and the presence of other persons. The setup used to support the tests include a 55" LED TV, a Kinect for gestures capture, and a new algorithm to allow the identification of closing and opening hands. Three test scenarios are described in this paper: the interactive visualization of a graph representing the academic genealogical tree of our University; the selection and manipulation of simple objects; and the free interaction with the map of a building. Given the results of the performed tasks, we conclude that the system, although not behaving very accurately in all situations, has potential to be used on many applications.**

***Keywords*-interactive computing; interactive systems; large-screen displays; natural language interfaces**

I. Introduction

Public displays are displays that are available to the public, usually in uncontrolled environments. They can be easily found in airports, shopping malls, parks, restaurants, etc. They are used to present the following sessions in the cinema, the price of some product, the dish of the day, important news, and a lot of other information that may be useful to those who see it. These mentioned screens are static and, normally, non-interactive. However, increasingly interactive displays are been presented to the public [1] and are a tendency for the next few years [2], [3], [4]. A couple of works [5], [6] have been done, that describe situations in which people faced an interactive display on a public space. It is important to notice that, in these situations, the user does not have the comfort of sitting at a desk and use the conventional mouse and keyboard to interact. He needs to stand up and often move physically to be able to explore all the information.

Although there are works that explore the capabilities of an interactive public display, this is an area still poorly addressed, and consequently there are few indications of which would be the best methods of interaction to be used. In the works mentioned above, large touch-screens were used. However, other approaches can be taken, as will be seen in Section II. Among these approaches, one that draws attention is the one that can be used in any type of screen, and where the user does not need to hold any device to interact with the information displayed.

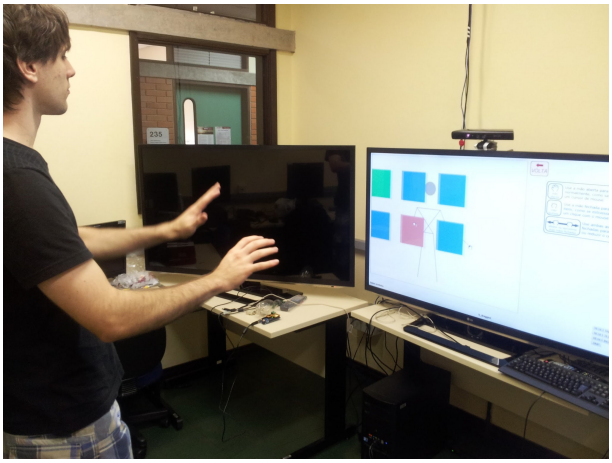

Figure 1. Snapshot of a user interacting with a public display.

In our work, a system is proposed in which the user only uses his hands to perform tasks on an interactive display (see Figure 1). We implemented case studies that explore natural gestures for selection and manipulation of virtual objects in 2-D, pan and zoom on an assorted amount of information. Such natural interaction can enhance performance and the overall user experience [7]. Aiming its applicability to public displays, several criteria that could interfere on the user interaction on uncontrolled environments were formally evaluated, such as the amount of illumination in a room and the presence of other people in the same space. Furthermore, the accuracy of the proposed technique was evaluated for different scenarios and goals.

After the analysis of the results obtained on the tests conducted, it is clear that the proposed model is good enough in certain scenarios, such as the selection and manipulation of large objects and panning and zooming the screen, but it lacks in others, such as the selection and manipulation of small objects. Based on the low number of studies on the

subject, one can say that there is not a method that allows user interaction without devices more accurately. However, while the technology does not advance to create robust devices for gesture recognition, the methodology proposed in this paper can be applied with acceptable performance and at a very low cost.

The remaining of this paper is organized as follows. Section II present related works on interaction with large displays, and gestures recognition using the Kinect, the hardware we are using for gestures capture. Section III explains our strategy for device less gestural interaction and Section IV details the design and implementation of this project. Section V details the user studies conducted and Section VI presents the results obtained. Some key observations are presented in the Section VII as well as conclusions and future works in Section VIII.

II. Related Work

In order to identify the criteria that must be taken into account, in this work, we covered the state of the art on interaction with large displays. Moreover, as the goal is to build a model that employs gesture recognition without any device manipulation by the user, it is also important to examine studies that employ the Microsoft Kinect, since it is currently the device with the better cost-benefit relation [8].

A. Interaction on large displays

Touchscreens have been studied for a long time. However, with the advent of the multi-touch sensitive screens, its use has been increasingly common, even applied to games [9]. Two studies were conducted on the same touchscreen system at the Helsinki Institute for Information Technology [5][6]. The architecture of the system proposed involves a semi-transparent back-projective screen and a camera sensitive to infrared emissions, positioned next to the projector. Touches are detected emitting infrared light on the screen. According to the authors, the system provides the capture of as many taps as possible to be made by the dimension of the screen.

A common technique used to perform the interaction with large displays is the use of mobile devices as interactive tools [10]. For example, the *Touch Projector* [11] uses a smartphone to record a big screen, identify it and receive its content. The work of Pears et al. [12] is very similar, using the camera of a smartphone to locate an object to be manipulated on a large screen and using the mobile device as a 3D mouse, being able to provide 4 DOF interaction.

The most employed approach is to create new devices or to adapt an existing device, such as the Nintendo Wii or a 3D mouse, especially in cases where a system whose cost is not prohibitive is needed. The VisionWand [13] is an example. It consists of a wand with colored tips that are read by a couple of cameras and provides the user the selection and manipulation of virtual objects in 5 DOF interaction. Differently, the LOP-cursor [14] uses the accelerometers embedded on a smartphone to interact with objects in a display wall very precisely.

The interaction through data gloves is also a common practice. Vogel and Balakrishnan [15] present a data glove with passive reflector points in the fingertips and a Vicon Motion Tracking to identify these points with which you can manipulate virtual objects in front of the screen. Mohering and Froehlich [16] use optical mechanisms for capturing the position of the fingers and hands, in a system that, according to the authors, provides high accuracy, and has also haptic feedback.

B. Interacting with the Kinect

Although we can easily find in the Web many applications using the Microsoft Kinect to interact, there are few scientific research papers on the subject. An interesting research use 3 Kinects to generate 3-D models of the users [8]. Despite the good results, the authors emphasize that the low resolution of Kinect cameras prevent the generation of more sophisticated models, although, given the low cost for setting up the system (approximately US$ 600.00), the results are quite satisfactory. Another interesting work using Kinect to recognize humanoid [17] is able to detect the presence of humans with an accuracy of 98.4%.

Turning to an interactive application of the Kinect, the works of Bidgelou et al. [18] and Gallo et al. [19] can be mentioned. They use the device to allow interaction with medical data. However, both jobs require an initial calibration by the user. Another approach is the one of Wilson, which uses the Kinect to simulate a touch screen [20].

All works above uses the Kinect's depth camera to recognize patterns, showing that this is a promising approach. Although some works [18][19] are interesting and present good interactive techniques, training and calibration steps are inconvenient, especially when it comes to public displays, where the user will not be willing to interact if the system is not very simple. Based on that and in what was presented in the first half of this section, a new model should be proposed.

III. Deviceless Gestural Interaction

This paper presents an approach for gestural interaction without devices, i.e., without any device attached to the user body or in his hands. This approach is desirable for interaction with public displays, where the user interact standing in front of a large screen, does not want to share any devices, and also does not want any complicated mechanism to perform the interaction. We believe that in an ideal situation, the user faces the display and automatically discover what he need to do in order to interact with it. Our approach seeks such panorama.

In the proposed system, the user interacts with his hands (either one or both) positioned in front of the body to perform translations, zoom, selection and manipulation of elements on the screen, all of this in a 2-D environment.

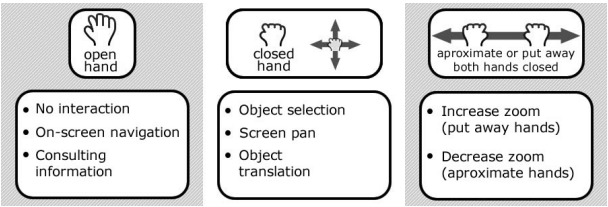

Figure 2. Gestures supported by the proposed model: at left, with both hands opened, at the center with one hand closed, and at right with both hands closed.

In order to differ between everyday gestures and gestures to be recognized by the system, the user must close his hands to interact. The system does not need any calibration and automatically identifies when a user is in front of the screen to interact. It always focuses on the user closest to the display in an area in which he can easily see its contents, i.e., without being immediately in front of the screen.

With both hands opened, the user can browse the information on the screen without changing them, seeing information that is hidden (e.g. in tooltips). Closing one hand, the user can select and manipulate the information on the screen. Closing both, it is possible to zooming. Figure 2 illustrates this graphically.

These gestures were chosen in order to trace a precise parallel with daily tasks. For example, when you query a particular entry in a list, it is common to swipe the items on the list until stop in to the desired information. When choosing a product in the supermarket, the user extends his arm and closes his hand on the object choosen, making a selection between products. When a user moves an element on a table – e.g. the mouse itself – he closes his hand over it and moves his wrist to the position where he wants to drop it.

The gestures for pan and zoom were based on techniques already established for these activities on touch devices. To zoom into a picture, we normally use two fingers on the screen, whose positions deviate to zoom in and get closer to zoom out. When a picture has undergone a zooming and does not fit completely on the screen, the user uses a finger whose position changes according to where he wants to put it. Therefore, the proposed gestures seem quite intuitive and adequate to mimic the gestures that the user use to perform, reducing the training time required for the proposed technique.

In order to evaluate the proposed model, three applications were developed, so that it was possible to verify each of the functionalities provided. Each one will be described in greater detail below.

A. Academic genealogical tree

The first application developed was a graph visualizer that exhibits the academic genealogical tree of the Computer Science Graduate Program of Federal University of Rio Grande do Sul (PPGC) in the form of a graph. In this graph each node represents a student or a professor and the edges between them represents advisory relations. The system includes a tooltip that informs the person's name of the node when the user have his hand over it.

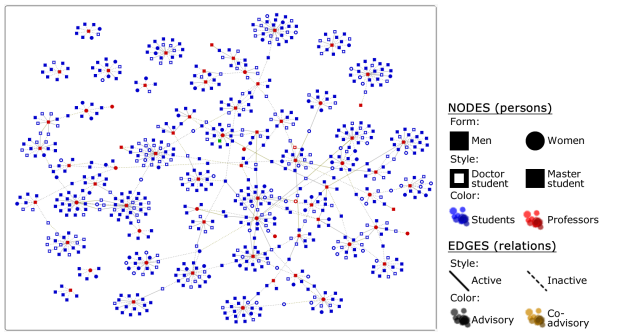

Figure 3. The academic genealogical tree of the PPGC in form of a graph: nodes are students and professors and the edges are advisory relations between them.

Nodes and Edges have different colors and forms, which hold important information: squared nodes are men while rounded are women; red nodes are professors while blue ones are students – except when representing the system user, in which case, the node is colored in green –; master students nodes are solid while doctor ones are with a white square (or circle) inside; grey edges represent advisor relationships while brown ones represent co-advisory; and, at least, solid edges represent present advisories while dashed ones represent past advisories. Figure 3 displays the complete genealogical tree of the PPGC (799 nodes and 836 edges), along with a legend of the semantics of the graph.

All gestures previously described can be used in this application: open hands to consult nodes information; selection and manipulation of the nodes; and pan & zoom on the screen. In order to the user to localize himself on the system, hand icons indicate the position of the user real hands on the screen, as well as if the hands are opened or closed.

This application was developed as a real scenario of the implementation of the proposed interaction technique. The goal was to install a public display in a main building of the University where any student of professor could consult his connections in the academic genealogical tree of his Graduate Program.

B. Selection and manipulation of simple objects

This application was developed for conducting experiments with users. It presents a series of small tasks that must be accomplished by the users, comprising especially selection and manipulation of objects. Initially, six squares are displayed on the screen, one being green, indicating that it should be selected by the user. Upon selection, another of the six square becomes green and, thus, will be the next that the user will have to select. This procedure is repeated

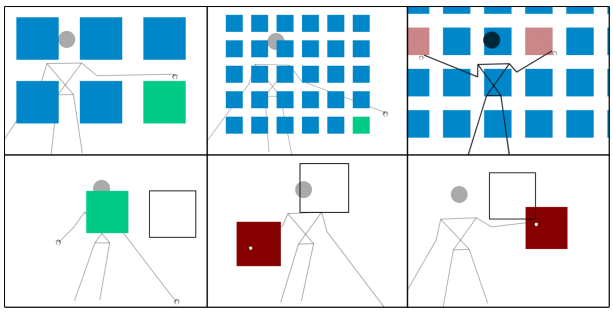

Figure 4. Screenshots of the application to select and position simple objects.

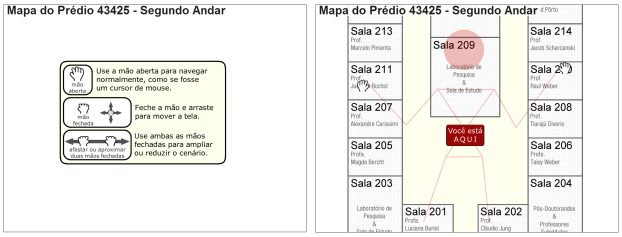

Figure 5. Two screenshots of the application that displays the map of the building.

five times, and then the size of the square decreases by half, which enables them to appear in greater numbers. After five new selections, the squares decrease in size again and, after five more selections, a third time.

At the end of these last five selections, a green square of the original size (large) appears on the screen and as well as a hollow square in black, slightly larger than the green one. From then on, the user must not only select the green square, but also position it so that it is within the hollow black square. The size of these squares also decreases after five positionings, but only twice, not three as in the selection task. After the last positioning, a message is displayed stating that the task finished.

The user can pan and zoom on the screen at any time by just closing his hands on any screen space that does not contain the green square. The user skeleton is displayed in semi-transparency, and also icons indicating the position and status of the user's hands. In Figure 4 is possible to see that the skeleton of the user is drawn behind the squares, while the icons of hands are always on top. Figure 4 shows execution steps of this application: at the top, the tasks of selection – from left to right: the home screen of the application, the decreased size of the square after the first five selections and zooming in screen; below, the positioning tasks – from left to right: the initial screen of this task, the user manipulating a square with his left hand, and finally, doing the same with his right hand.

This application supports gestures for selection, manipulation, panning and zooming on the screen. The gesture for navigation is also present, but the only information that it displays is a visual feedback about the square which is below each hand. The application was developed so that the level of difficulty of the tasks increases as users gets practice with the system, which was confirmed after the application of tests with users.

C. *Localization map of a building*

This application was built to analyze how people react to a public interactive display. It shows the map of the building of the Institute of Informatics at UFRGS where professor's offices are located as well as some research laboratories. The map shows all the rooms and corridors of the building, with indications of room numbers and names of professors that own the room, if that is the case. It has also an indicative of the location where the public display is on the building, serving to the user as a locator.

The map is only loaded when the presence of a user in front of the display is detected. While a user is not detected, a window with instructions about the system is displayed (Figure 5-left). When a user is detected, the system continues displaying the instructions for 9 seconds, but also starts to display the map of the building and the semi-transparent skeleton of the user behind that window. After 9 seconds, the window disappears and the user is free to interact with the application (Figure 5-right).

This application also shows the user skeleton so that he can sense his presence in the system, and also the same hand icons already used in the previous application. The application recognizes gestures to pan and zoom on the map.

IV. Design and Implementation

We wanted to build an interaction model that could be easily implanted anywhere. In order to achieve this, the system would need to have low financial cost and be as portable as possible. Moreover, as already observed, it was desirable that the interaction did not require any manipulation or coupling of devices, giving users freedom to move as he wanted and without the need to share objects. With that in mind, Microsoft Kinect has proven the best alternative because it has a very low cost and with the help of an SDK, provides useful data in a quite simple way. As seen in related work, the device has enough potential to provide deviceless interaction, provided that its limitations are circumvented.

Having the hardware necessary for the interpretation of gestures, a definition of the user interface was needed, which would have to be developed according to the criteria of portability desired. With the evolution of Internet applications features, especially after the arrival of the HTML5 standard, a Web approach proved to be interesting. It is well established – especially with the increase of applications "on the cloud" – that Web systems are easily portable, therefore, the choice of this approach seemed appropriate. Thus, the

proposed model was developed in a Web environment and using the Kinect as interactive device.

A. Capturing Kinect data

The Kinect interpretation is through the Microsoft's *Kinect For Windows* SDK, which has a number of advantages over its competitors, especially not requiring a user calibration pose to recognize its skeleton, as the OpenNI SDK. For this work, the depth information and user's skeleton were used.

The first step is to detect if there is a user to interact. That is, when the SDK informs that the skeleton data are ready. Then, with the joints information from the skeleton it is discovered the position of the hands of the user, using the 2D points corresponding to the joints *HandRight* and *HandLeft*. Finally, it identifies the status of the user's hands.

As currently no SDK can recognize natively if the user's hand is open or closed, a post-processing of the Kinect information need to be done to get this information. For the system to work with any user without a step of calibration, image processing algorithms are employed on the obtained depth image. To identify the status of each hand the system takes three steps, better described below.

1) Isolating the hand: The first step is to find and isolate the hand of the user. To achieve this, we use two joints of the skeleton obtained by the Kinect: wrist and hand. The coordinates of the skeleton joints are mapped to coordinates on the depth image, obtaining the points $W(x, y)$ of the wrist, and $H(x, y)$, of the center of the hand. Calculating the distance d between these points it is possible to set up a square of side $2d$ centered at point H, which encompasses the entire region occupied by the hand on the depth image, as shown in Figure 6-A.

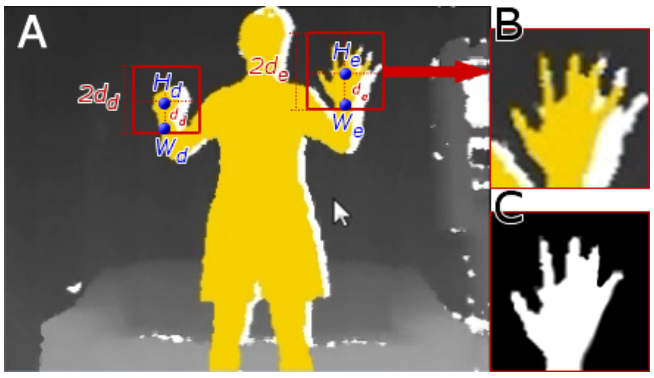

Figure 6. Isolating user hands on the Kinect depth image. Using the Kinect joints hand (H) and wrist (W), separated by a distance d, we obtain a square with side $2d$ that encompasses the hand.

Then, the depth information P, in millimeters, of the point H is obtained from this square image (Figure 6-B), and used to be compared with all other image pixels. Pixels are colored in white when its depth information is inside the range $[P(H) - 30mm, P(H) + 70mm]$. Otherwise they are colored in black. At the end of this process, it is obtained an image of the user's hand in white over a black background, as can be seen in Figure 6-C.

2) Detecting the contour of the hand: The next step is to detect the contour of the hand previously isolated. This procedure is done in two specific steps: first, a high-pass filter that left only the outline of the hand is applied to the image; then the image goes through a new processing step to detect contour pixels that are adjacent to each other. The second step is necessary to eventually identify – using the K-Curvature algorithm [21] – if the hand is opened or closed.

The first step, i.e. the high-pass filter, is done by an image convolution with a $3x3$ mask, centered in the pixel being read. If the center pixel is white and any other pixels inside the mask are black, this pixel belongs to the contour and is kept in white. Otherwise, it is colored in black, as exemplifies the Figure 7. At the end of this process, only the contour of the hand remains in white.

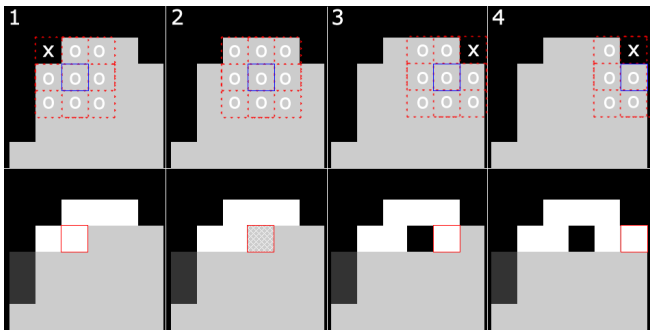

Figure 7. Contour extraction. Above, the image to be processed. The blue square indicates the center of the $3x3$ mask. Below, the corresponding results, with processed pixels in black or white.

To detect the adjacency between contour points, first it is defined an array A to store them. Then, starting from any white pixel, it is made a comparison between the neighboring pixels within a $3x3$ mask. If any of the neighboring pixels is white and it is not yet in A, the pixel is added to the array and the mask is centered in that pixel, repeating the process. If none of the neighbors of the pixel is white and is not yet in A, the mask is expanded to $5x5$ for new comparisons. If even so a neighbor that fits the test is not found, the mask is enlarged once more to $7x7$. If the test fails again, we conclude that the entire contour has been read and the algorithm stops. At the end of this step, the array A contains, in order, the contour points of the image.

Testing with 3 different sizes of masks is necessary because the quality of the Kinect depth image is very limited and it may be that, with a small mask (e.g. $3x3$), the whole contour is not detected. Figure 8 exemplifies situations where larger comparisons masks are needed.

At the end of this processing, the array A contains (in order) the contour points of the hand and can be used in the

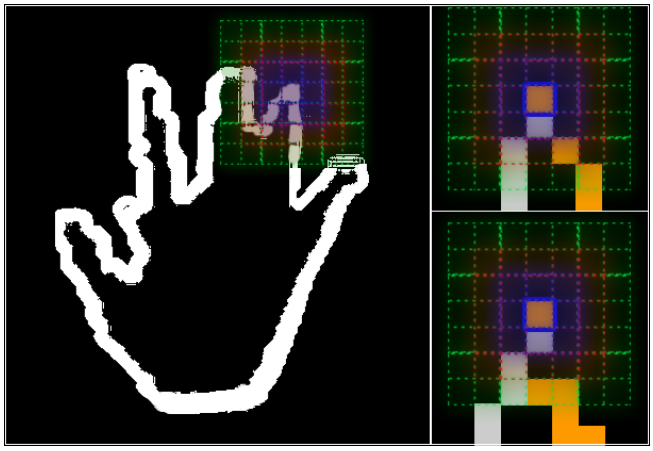

Figure 8. Searching the adjacent contour points of the hands: at left, a typical problematic point; at right, grey pixels represent pixels already in the array A, and orange points those that are not. Above, a situation in which a mask $5x5$ would be necessary to identify all contour. Below, a situation where only a mask $7x7$ would suffice.

execution of the final step of the algorithm.

3) Discovering the state of the hand: Finally, the last step to discover if the hand is opened or closed is to find the fingertips, which represent discontinuities in the contour of the image. To achieve this, we used the K-Curvature algorithm [21], which runs on a contour. To implement the algorithm it is necessary to define two constants: $pointsInterval$, which defines the range in which the contour points will be read; and $limitAngleToBeFingertip$, which defines the angle between the vectors that represent a discontinuity in the contour. In the proposed model, these values were set to 6 and 50, respectively, after a series of tests with different values.

The algorithm takes as input the array A containing the points of the contour. The algorithm starts processing at the position $pointsInterval$ of the array and continues at intervals until reaching the end of the array minus $pointsInterval$ positions. For each position i being analyzed, the positions $i - pointsInterval$ and $i + pointsInterval$ are read. Then the angle between the vectors $(i - pointsInterval, i)$ and $(i, i + pointsInterval)$ is calculated. If this angle is less than $limitAngleToBeFingertip$, the position i corresponds to a point of discontinuity. See below the pseudo-code that implements the algorithm described.

At the end of process, if any discontinuity point (a fingertip or a point between two fingers) has been detected, it is considered that the hand is open. Figure 9 shows an implementation of the algorithm on an image. Red dots circled in white are the points of discontinuity detected, being formed by the orange (for fingertips) or green vectors (for regions between two fingers). We can observe that not every point of discontinuity was detected, but detecting only one of them is enough for this work.

Algorithm 1 The K-curvature algorithm used in this paper.

```
numberOfDiscontinuities = 0
for i = 0 to arraySize(A)-pointsInterval do
    p1 = contourPoints[i - pointsInterval]
    p2 = contourPoints[i]
    p3 = contourPoints[i + pointsInterval]
    angle = angleBetweenPoints(p1, p2, p3)
    if (angle <= limitAngleToBeFingertip) then
        numberOfDiscontinuities += 1
        i = i + pointsInterval
    end if
end for
```

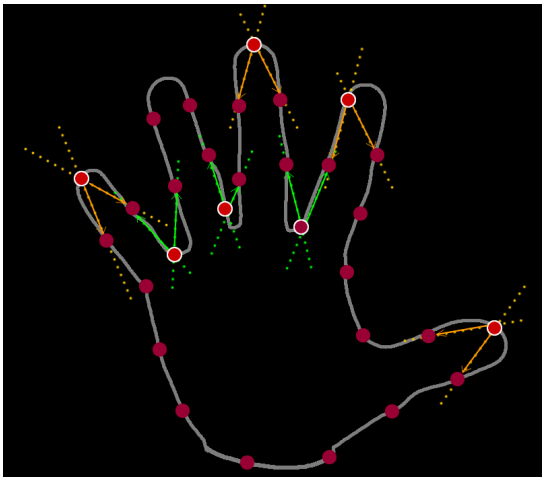

Figure 9. Running the algorithm K-Curvature [21] on the outline of a hand, with the points of discontinuity detected marked with white circles.

As fast movements of the hands can cause the system to misinterpret their states (e.g. an open hand can be recognized as closed if the user is moving it too fast, because of the "blur" in the camera), a small buffer of 10 positions was built to store the states of the hand as in a queue, which is read and transmits the state that is in greater number among its positions. This causes a slight delay in interpretation, but compensates with a greater stability.

To verify the accuracy of the whole process, an application was developed in C#, that shows the Kinect depth image being read at the center of the screen and both hands in each side, with the detected descontinuty points identified. Figure 10 presents a screenshot of the application.

B. Integrating the Kinect to the browser

For this integration be possible, it must be used a client-server architecture, where the application that reads and interprets data from the Kinect acts as a server and communicates via message exchange with the client application, developed for Web platform. The exchange of messages is

Figure 10. Application develeped to test the whole hand detection algorithm: in white, the fingertips identified.

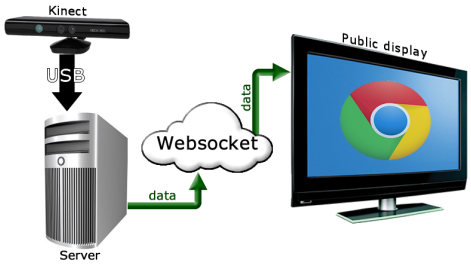

Figure 11. The architecture of the proposed system, describing the communications involved in the process of transposing the data interpreted from the Kinect to the Web application in a public display.

made using a WebSocket. The application that interprets the Kinect should open a WebSocket and wait for connections. In turn, the Web application should connect to the opened WebSocket and inform that it is ready to receive messages. After that, the server application must send messages that contain useful data that were obtained from data processing of the Kinect. It is transmitted basically an array containing the following information: location of the joints corresponding to the head, both shoulders, elbows, wrists, legs and hands, two joints corresponding to the waist and, in particular, two integers that indicate if the hands are open (0) or closed (1). Received these data, the Web application can interpret them and use them for different purposes, using only the data that it needs. Figure 11 illustrates the communications involved in the developed system.

C. *Interpreting the data on the Web*

The Web applications used in this study were constructed using languages HTML5, JavaScript and PHP. The entire visual part was implemented in HTML5, using CSS for graphic details. The interactivity of the pages takes place by means of JavaScript language, using the library jQuery, which facilitates the manipulation of screen elements.

To interpret the data received via WebSocket's message, the applications have a state machine that is updated according to the state information of each hand. The possible states are *idle, selection with right hand, drag with right hand, hold with right hand, selection with left hand, drag with left hand, hold with left hand, and zoom*.

The hand position is updated regardless of the states, and is indicated by icons that show open and closed hands. Additionally, a skeleton formed by line segments is constructed

Figure 12. Place of the application of user testing. It is possible to see the location where the user should position himself.

on based on the joints information received via message, as can be seen in Figure 4. These updates are made every incoming message, as well as updating the state machine.

Selections and the drags/translations are determined according to the position of the hand. If there is an element in the position of the hand, this element will be affected, otherwise will be the whole canvas, which is built with the use of an HTML5 element *¡canvas¿*. The action release is used to release the element or screen for positioning action. The zoom is performed based on the position of both hands, focusing on mid-point between them: when it starts, it is stored the distance between the points that represent the positions of the hands, d_0; then, in later updates, the distance between these points, d_i, is recalculated. If d_i is greater than d_0, it is considered that an expansion was performed, i.e. an increase of zoom. Otherwise, i.e. if d_i is smaller than d_0 is considered that a reduction was made, namely a decrease in zoom.

V. User Evaluation

Once the system was developed and working, the three case study applications were evaluated according to the relevant criteria to its installation in a public display, namely: ambient lighting, presence of other people, type of location, task type performed and presentation of information. Each will be explained in more detail below.

Firstly, the application of the graph visualization was informally evaluated in order to get an overview of the proposed solution. In this preliminar evaluation some minor errors where detected and solved, specially questions related to the hand status recognization precion, which proved insuficient for precise interactions, as to select and move a single node of the graph.

The application that shows the map of the building was installed at the entrance of the building to which it's map refers. There it was monitored for a period of 12

hours spread over two days in order to observe people's reactions before a public interactive display. The application of selection and manipulation of simple objects was used for formal evaluation with users. To this, 38 users were asked to perform the tests proposed and the time spended to execute the tasks and the number of errors were recorded, as well as the use of pan and zoom. All testers performed the tasks at least twice, the first one being for they become accustomed to the system. The users also answered two questionnaires: the first before carrying out the tests in order to characterize them and the second with their opinions regarding the tasks, applied after testing, for subjective evaluation purposes.

Users who performed the tests are divided among 30 men and 8 women, have an average age of 23.82 years and a median of 24, all accustomed to the daily use of a computer. The experiment was conducted in a room with artificial lighting by fluorescent lamps with two variations of brightness, as well as with and without the presence of other people. Three users had to be discarded due to errors in the execution of the application, totaling 35 users with useful data. There was a mark on the floor indicating the optimum position where the user should position himself, which was ample in order to allow horizontal movement. It was produced a video[1] explaining about the tasks and gestures that was shown to all the testers after they answer the first questionnaire. With the explanation of the test done on video, all users received exactly the same instructions, not being induced by something that the conductor has passed individually. Figure 12 shows the environment of the tests.

The tests had as independent variables, i.e. which does not depend on the user, the job type (selecting/positioning) and the size of the squares. The dependent variables, which change according to the user, were the time of each task and the number of errors occurred. It was considered as *error* a selection of other than the green square on the task of selecting and the dropping of the square to be positioned in a location other than the correct destination on the positioning task.

VI. Results

Although it has its problems, the proposed model in this study has great application potential when used for simple interactive tasks. The application of selection and manipulation of simple objects was used for formal evaluation with users. In the evaluation, testers responded to several questions about the tasks and their responses indicate that the proposed system had good performance in certain situations, such as the selection of elements and panning and zooming in the screen. Figure 13 presents the general evaluation of the system by the users. If we consider that a medium evaluation as a good result, we can see in that graph that all the evaluated criteria are with more then 50% of acceptance.

[1]https://www.youtube.com/watch?v=tXKwPY9-X5c

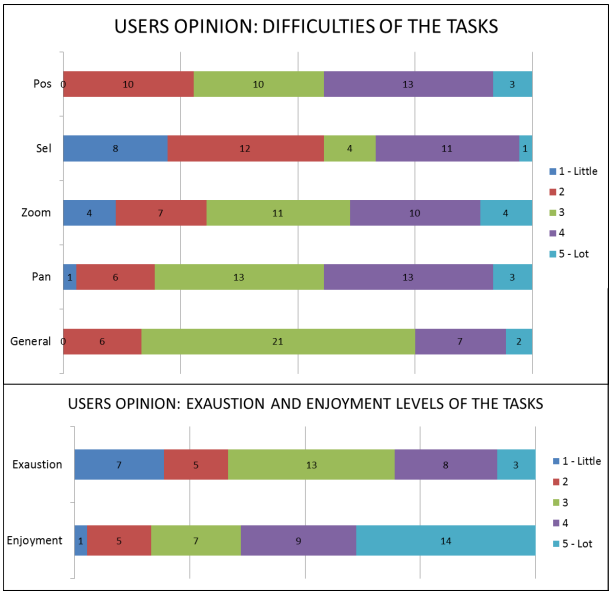

Figure 13. Graphic displaying the user evaluation of the system according to the questionnaire applied.

In a field for general comments on the second questionnaire, many users noted that the system is tiresome on certain situations, which may have been generated by the difficulty in positioning the smaller squares when the target position was too far away. Another interesting feature that was observed by users is that the practice of the interactive gestures greatly improves user performance. Quoting a comment: "with a short time of use/interaction, the domain of the application functionality is already reasonable". Thanks to the implementation of the entire task only as training at first, the user already performs the tests with greater speed and accuracy than did in training session.

Below, in specific subsections, the results of all criteria evaluated in this work for a gestural deviceless interaction for a public display are described.

A. *Illumination conditions*

In order to evaluate if the difference in illumination was determinant, 8 users were selected, without any specific criteria, for testing the application twice (in addition to the first, the training one) with two light variations measured by a luximeter: with normal lighting, of 731 *lux*, and with no lighting other than the display, of 221 *lux*. To not be a determining factor in the assessment, users performed tests alternating between the two light intensities.

As expected, the lighting was not a relevant factor in interactive tasks. According to an analysis of variance (ANOVA), selecting squares did not had its time determined by the amount of illumination, obtaining p-values 0.9256, 0.2822, 0.7816 and 0.6303 for square sizes ranging in sizes of four major the lowest, in order. The positioning also

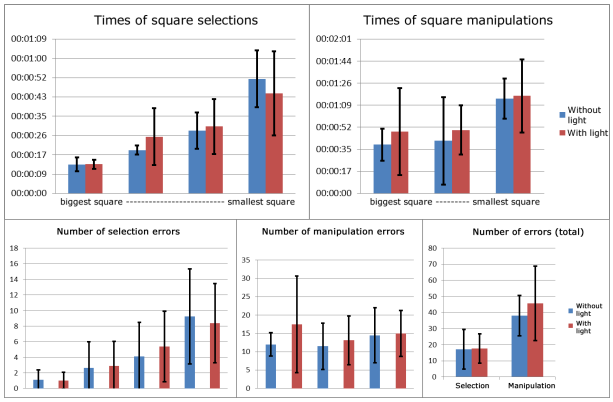

Figure 14. Graphics showing the differences in the data obtained in tests with the illuminated by lamps or not: above in relation to the average execution time of each task and below in relation to the average number of errors occurring in each task. Regarding the size of squares, the arrangement is always from the biggest to the smallest. The illuminated environment is represented by the blue bars and without illumination by red. The black line on each bar marks the standard deviation.

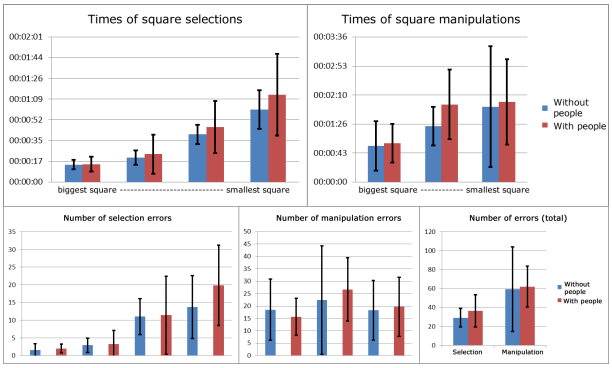

Figure 15. Graphics showing the differences in the data obtained in tests with and without the presence of other persons: above in relation to the average execution time of each task by users and below in relation to the average number of errors occurred in each task. Regarding the size of squares, the arrangement is always from the biggest to the smallest. The interaction without interference is represented by the bars in blue and with the presence of people the ones in red. The black line on each bar marks the standard deviation.

had the same response, obtaining p-values 0.4394, 0.3514 and 0.8991, ranging in sizes of squares large, medium and small, respectively. The number of errors in each task also was not significant, with p-values 0.9274 and 0.4267 for selection and positioning errors respectively. Thus, according to ANOVA, the illumination should not be a relevant factor when applying the proposed system in a public display, as can be seen in the graph of Figure 14.

B. Presence of other persons in the same place

To evaluate whether the presence of other people in the environment is an important factor to interact in a public display, 10 people, other than those who evaluated the differences in lighting, were randomly selected to complete the tasks twice: once with the presence of people passing behind and ahead of them, bowing at his side and trying to simultaneously interact with the display, and a second without any interference. Again, avoiding the interference in the tests, the order of tasks was alternated between users. For the test with interference of other people, only one another person was sufficient to introduce an annoyance to the tester, since he stayed all the time aside of the tester.

Contrary to what was anticipated, according to an analysis of variance of the execution times of the tasks and the number of errors, it was not possible to conclude that the presence of other people in the same environment that the interacting user impact on the completeness of the tasks. The analysis of the selection of squares had p-values 0.8566, 0.6293, 0.5106 and 0.4404 with square sizes varying in descending order. In turn, the positioning of the squares analysis had p-values 0.7939, 0.3363 and 0.8034 from square in three sizes in descending order too. The analysis of variance of the number of errors produced p-values 0.2637 and 0.8600 in relation to the selection and manipulation respectively.

Not confirming the hypothesis is a very positive point for the proposed system, indicating that it is able to identify the person who is interacting and remain consistent throughout the interactive task, recovering quickly from errors, even though the average time of completion of tasks is greater when there were other people. Figure 15 presents graphics showing how close are the times and number of errors in the various tasks with and without the presence of other people in the environment. The analysis of the limits imposed by the lines of standard deviation indicates that both environment settings are similar.

C. Sort of place

The place where the interaction in a public display take place can influence the execution of tasks due to lightness influence or due the presence of another people in the place. These specific criteria were evaluated and the results presented above. However, some psychological factors can also influence the outcome and these factors can not be measured quantitatively. Aiming to evaluate the system in a real environment of use, observations were made when the system was installed at the entrance of a building with some traffic of people (around 100 persons per day) and it was possible to perceive the existence of certain behavior patterns.

On the first day in particular, it was possible to see that people demonstrates interest passing by a public interactive display, looking quizzically at the screen when their skeleton appears on it. However, despite the interested, people seem to be afraid to try out the system, deciding to follow his path after a some hesitation.

This pattern differs when people are in a group, in which case people seems to feel encouraged to interact with the

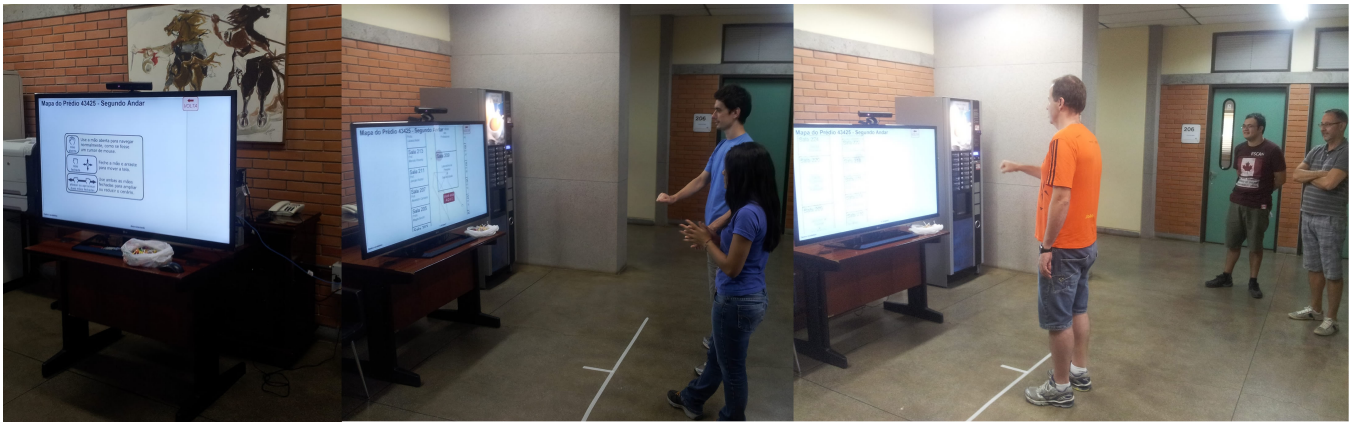

Figure 16. At left the system as installed in the entrance hall of the building whose map is displayed in the application. At right two pictures showing groups interacting with the display.

system as if to appears bold. One of the elements of a group always takes the lead and begins to interact with the screen, then the other members approximate and try to do the same, as if to disturb his companion, but usually without success, since the system focuses only on one user. After a few moments of exploration, the group moves on, commenting about their experience. Figure 16 shows the display in its initial state where it was placed and two situations in which groups interacted with the system.

The fear in interacting with the system also disappears when the user thinks he is alone in the environment. This became clear when observing a user who would often come to take papers from the printer next to the screen and always watched the display, but as there were always other people going by, he did not dare to interact. However, in one of his visits to the printer, he looked around and saw no one and then tried to use the system for a few minutes, but soon returned to his business (especially because as he carried papers in one hand, the system did not recognize if it was open or closed and did not behave properly). The observer was always in a location far ahead of the display, about 6 feet, but the system holds the users' attention with such intensity that the observer was not noticed.

The system behaved identically both in the closed room and in the lobby of the building, although one can not say the same of those who interacted with it, especially those who did it alone. There seems to be something that restricts people when they must perform unusual gestures in public. In general, the system seemed to be well accepted by users who dared to use it and succeeded. One user commented that "to interact correctly is just a matter of getting used to the gestures".

D. Sort of task

As explained in Section III, the developed applications explore activities of navigation, selection and manipulation of objects and pan and zoom on the screen. The application

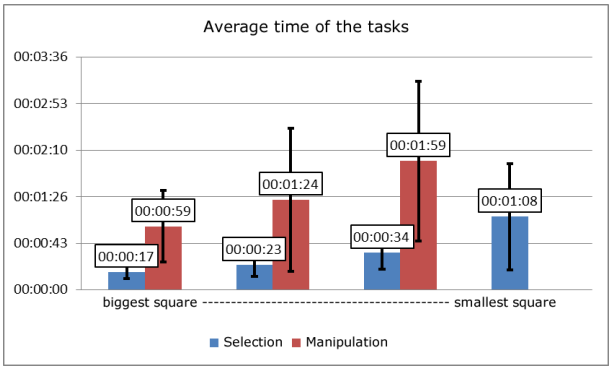

Figure 17. Graphic comparing the average time of completion of each task of selection (blue) and manipulation (red). The black lines indicate the standard deviation range.

of selection and manipulation of squares was especially developed to evaluate the type of tasks, varying in difficulty and type and allowing the user to always make translation and zoom on the screen, trying to present the tasks with a level of increasing difficulty. For this evaluation, we compared the results of every 35 users with maximum illumination and without the presence of people.

As assumed, after an analysis of variance, it become pretty clear that the selection task is easier to perform than the manipulation. The three sizes of squares that are common to both tasks resulted in the analysis p-values $2.4523*10^{-10}$, $1.0989*10^{-6}$ and $10*6.4350^{-9}$ with sizes in descending order. Thus, with a very high probability, the selection task should be easier than the positioning, as shown by the graphic of Figure 17.

On further analysis, it is concluded that even the selection of the second largest square should be easier than positioning the larger square (the easiest to position), resulting a p-value $4.9689*10^{-8}$ in an analysis of variance. The same is true when comparing the selection of the second smaller square

with the position of the largest square, in which case results a p-value 0.00019, a number quite higher than the others, but still statistically significant.

However, no statistically significant correlation was found between the task of selecting the smallest square and of manipulating the larger square, in which case ANOVA results p-value 0.3794. This scenario does not occur when analyzing the number of errors when the p-value obtained is 0.0013, indicating that the task of selecting the lowest square produces less errors that positioning the larger square.

Regarding gestures to translate the screen and change the zoom level, all users used them at least once. The translation was performed frequently by mistake, which disturbed the interactive task especially in positioning. Both the translation and the zoom were performed on both types of tasks. Many users used the translation to bring the square to be selected to a position above their shoulders, an area where the interpretation of gestures was more accurate. Furthermore, many testers used the zoom when the squares were presented in its smaller size, both in selection or positioning tasks.

E. Information presentation

In the application of selection and manipulation, the squares have its size decreased during the execution of tasks, and are presented in greater number in the screen, both in the selection and manipulation tasks. Using this, we evaluate whether the size and number of objects interfere in time to accomplish the tasks, comparing the results of all 35 users.

Concerning the size and quantity of the square, the hypothesis that larger squares are easier to select and position was confirmed. Considering the hypothetical square sizes 8, 4, 2 and 1 for the selection, the squares of size 8 should be easier to select than those of size 4, according to an analysis of variance with p-value 0.0025. Following the rule, a square of size 4 should be easier to select than one with size 2, with a p-value 0.0004, and of size 2 relative to size 1 with a p-value 0.0002.

The results repeats in the positioning task. Considering the hypothetical sizes of squares 8, 4 and 2, the squares of size 8 should be easier to position that those of size 4, according to an analysis of variance with p-value 0.0529, and those of size 2 with p-value $3.8358*10^{-5}$. In turn, squares of size 4 should be easier to position than squares of size 2, according to an analysis of variance with p-value 0.0369.

VII. Lessons Learned

In the development of this project, the low precision in the recognition of some elaborated gestures imposed by the low resolution of the Kinect's depth camera – only 640x480 pixels – was a recurrent problem. Due to this, some user movements were not identified, which caused some frustration to the user. Fortunately, this problem may be solved with the employment of the new Kinect 2, that shall arrive with new Microsoft video-game, X-Box One.

There is not confirmed data about the new Kinect, but some information lead us to believe in a huge upgrade of the device. With a Full HD camera, for example, the resolution problem of the fingers detection might be solved. And there is still the possibility of the Kinect 2 be able to natively recognize the individual fingers along with the recognition of the rotations between the skeleton joints, which might be useful for a tridimensional interaction approach.

Besides the technological needs, we also noticed, during the observation of the use of the system in a real place, that the application available on the public display must present relevant information to the users. As in the conducted experiment the application had just information that the users already knew – since everyone was well known to the building referenced by the localization map – they did not used the system for more than 5-10 minutes. This way, maybe they did not have the necessary time to become accustomed to the proposed gestures and thus be able to perform a most robust interaction.

VIII. Conclusion and Future Work

Was presented a study on a gestural interactive system for public displays that does not require the user to hold any device. In the proposed system, the user stands in front of a public display and use his hands to interact with the information presented on the screen, using for that gestures similar to the ones used in everyday situations, as close the hand to select an object and put away two points of contact to enlarge the screen.

Based on the evaluations, it is possible to say that the system, despite having deficiencies in certain aspects, behaved well enough to interact with large objects in selection and manipulation at short distance tasks. It is also clear that systems employing features of pan and zoom can take advantage of the interactive method proposed, since users in a real pubic place were able to interact with the display in an application of this kind.

Although it does not provide a definitive solution to the interaction in public displays, the work serves to indicate the main difficulties when seeking a solution to this interactive problem. The proposal introduced by the system, using the closing hand to differentiate the selection from navigation seems pretty intuitive for users, as well as the use of both hands to modify the intensity of zoom on the screen.

An interesting future work to be done is to integrate the system introduced in this paper with another interactive method, like those provided by mobile devices. In this scenario, the user would see and interact with information in a public display, but the application would allow him to use his own mobile phone to insert information or perform delicate interactions or also receive a copy of the data. In that way the user does not interact directly with any device that is not his own, avoiding any problems of sharing or learning curves.

Finally, it should be noted again that the solution proposed in this work to provide a free gestural interaction for public displays is interesting because it makes independent the applications of gesture interpretation and of information presentation. Thus, any developer could take advantage of the information obtained by the application that interprets Kinect and build his own Web application that does what he wants, provided it connects to a WebSocket and use the state machine to decipher the information received, thus obtaining the user's intent. And all this with a low financial cost, which comprises the values needed for the purchase of a Microsoft Kinect, a large display and a computer.

ACKNOWLEDGMENT

We would like to thank all the users who kindly tested the system and ceded their image rights. This work was partially supported by Microsoft Brazil Interop Lab at UFRGS, CNPq-Brazil through projects 311547/2011-7 and 485820/2012-9, and CPD-UFRGS that supports Thiago Motta.

REFERENCES

[1] U. Hinrichs, S. Carpendale, N. Valkanova, K. Kuikkaniemi, G. Jacucci, and A. V. Moere, "Interactive public displays," *IEEE Computer Graphics and Applications*, vol. 33, no. 2, pp. 25–27, 2013.

[2] ——, "Interactive public displays," *IEEE Computer Graphics and Applications*, vol. 33, no. 2, pp. 25–27, 2013.

[3] S. Boring and D. Baur, "Making public displays interactive everywhere," *IEEE Computer Graphics and Applications*, vol. 33, no. 2, pp. 28–36, 2013.

[4] P. T. Fischer, C. Zllner, T. Hoffmann, S. Piatza, and E. Hornecker, "Beyond information and utility: Transforming public spaces with media facades," *IEEE Computer Graphics and Applications*, vol. 33, no. 2, pp. 38–46, 2013.

[5] P. Peltonen, E. Kurvinen, A. Salovaara, G. Jacucci, T. Ilmonen, J. Evans, A. Oulasvirta, and P. Saarikko, "It's mine, don't touch!: interactions at a large multi-touch display in a city centre," in *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 1285–1294.

[6] G. Jacucci, A. Morrison, G. T. Richard, J. Kleimola, P. Peltonen, L. Parisi, and T. Laitinen, "Worlds of information: designing for engagement at a public multi-touch display," in *Proceedings of the 28th international conference on Human factors in computing systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 2267–2276.

[7] D. A. Bowman, R. P. McMahan, and E. D. Ragan, "Questioning naturalism in 3d user interfaces," *Commun. ACM*, vol. 55, no. 9, pp. 78–88, Sep. 2012. [Online]. Available: http://doi.acm.org/10.1145/2330667.2330687

[8] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan, "Scanning 3d full human bodies using kinects," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, pp. 643–650, 2012.

[9] D. Stødle, T.-M. S. Hagen, J. M. Bjørndalen, , and O. J. Anshus, "Gesture-based, touch-free multi-user gaming on wall-sized, high-resolution tiled displays," *Journal of Virtual Reality and Broadcasting*, vol. 5, no. 10, Nov. 2008, urn:nbn:de:0009-6-15001,, ISSN 1860-2037.

[10] D. C. McCallum and P. Irani, "Arc-pad: absolute+relative cursor positioning for large displays with a mobile touchscreen," in *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, ser. UIST '09. New York, NY, USA: ACM, 2009, pp. 153–156.

[11] S. Boring, D. Baur, A. Butz, S. Gustafson, and P. Baudisch, "Touch projector: mobile interaction through video," in *Proceedings of the 28th international conference on Human factors in computing systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 2287–2296.

[12] N. Pears, D. G. Jackson, and P. Olivier, "Smart phone interaction with registered displays," *IEEE Pervasive Computing*, vol. 8, pp. 14–21, April 2009.

[13] X. Cao and R. Balakrishnan, "Visionwand: interaction techniques for large displays using a passive wand tracked in 3d," in *Proceedings of the 16th annual ACM symposium on User interface software and technology*, ser. UIST '03. New York, NY, USA: ACM, 2003, pp. 173–182.

[14] H. Debarba, L. Nedel, and A. Maciel, "Lop-cursor: Fast and precise interaction with tiled displays using one hand and levels of precision," in *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, 2012, pp. 125–132.

[15] D. Vogel and R. Balakrishnan, "Distant freehand pointing and clicking on very large, high resolution displays," in *Proceedings of the 18th annual ACM symposium on User interface software and technology*, ser. UIST '05. New York, NY, USA: ACM, 2005, pp. 33–42.

[16] M. Moehring and B. Froehlich, "Effective manipulation of virtual objects within arm's reach," in *Virtual Reality Conference (VR), 2011 IEEE*, march 2011, pp. 131 –138.

[17] L. Xia, C.-C. Chen, and J. Aggarwal, "Human detection using depth information by kinect," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, june 2011, pp. 15 –22.

[18] A. Bigdelou, T. Benz, L. Schwarz, and N. Navab, "Simultaneous categorical and spatio-temporal 3d gestures using kinect," in *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, march 2012, pp. 53 –60.

[19] L. Gallo, A. Placitelli, and M. Ciampi, "Controller-free exploration of medical image data: Experiencing the kinect," in *Computer-Based Medical Systems (CBMS), 2011 24th International Symposium on*, june 2011, pp. 1 –6.

[20] A. D. Wilson, "Using a depth camera as a touch sensor," in *ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '10. New York, NY, USA: ACM, 2010, pp. 69–72.

[21] N. Shaker and M. Abou Zliekha, "Real-time finger tracking for interaction," in *Image and Signal Processing and Analysis, 2007. ISPA 2007. 5th International Symposium on*, sept. 2007, pp. 141 –145.