

Bevegelsessensor til bruk i interaktive kartinstallasjoner

John Wika Haakseth

John Wika Haakseth: Motion sensing technology for use in interactive map installations

KART OG PLAN, Vol. 73, pp. 71–78, POB 5003, NO-1432 Ås, ISSN 0047-3278

Motion sensing technology for human-computer interaction has become more readily available over the last couple of years. These devices use close-range depth-sensing principles that are comparable to those of a LiDAR scanner, recording a live depth scene that a developer can access and manipulate, e.g., by letting the user control data using natural gestures.

This paper presents project work done in the fall of 2012 in Geomatics, NTNU. The project sought to explore different cartographic interactivity methods through proof-of-concept development. The project resulted in prototypes of an interactive map installation, FloorMapper, based on open source software tools.

Two concepts of interaction were explored. One implemented virtual buttons, which allowed the user to step into interactive «boxes» placed in the XY-plane over the map to control it. The second one is based on tracking the user's joints. This enables gesture recognition and allows the user to perform more complex tasks using principles of natural interaction.

Qualitative proof-of-concept evaluations of the prototype show that the application works, but more work is needed with respect to presentation of the installation.

Key words: Motion sensors, interactive maps, HCI, open source

John Wika Haakseth, Software engineer, Norkart: P.O.B 145, NO-1300 Sandvika
E-mail: john.wika.haakseth@norkart.no

Introduction

Recent years have seen a wider availability of 3D-capable motion sensors. This opens up more possibilities for interaction than the keyboard-and-mouse-based interface that has prevailed during the last 30 years. The recent popularity of multi-touch controlled interfaces on hand-held tablets has shown that people quickly adapt to new ways of interacting with information on a screen.

Microsoft Kinect is a consumer device that uses close-range depth-sensing principles comparable to those of a LiDAR scanner. It records live at a frame rate of 30Hz, meaning that a developer can access and manipulate a live depth scene.

The motivation behind this project work was exploring different cartographic interactivity methods through proof-of-concept development. Using motion-sensing technology as an input interface may offer more flexible ways to control information on a screen.

The information can be displayed on a computer screen, projected to a large wall, or it can create an entirely immersive VR environment. This project presents the idea of projecting a map onto the floor and have users interact with it as they stand on it.

Related work

Collaboration on large surfaces that go well beyond the computer screen has been a research topic long before such technology was widely available. Florence et al. [9] explore the possibilities of a GIS WallBoard, which leaps from the desktop metaphor of office computers to a whiteboard metaphor. They discuss three different ways of interacting with the WallBoard, depending on the distance of the user to the board:

Arm's length: The user has physical contact with the board (e.g. on a touchscreen) and manipulates the data through touch,

using a pointer tool or his hands. Actions are defined using natural mapping from physical experiences, e.g. rotating the map as one would a physical map, by touching the board with two hands and rotating them.

Within spitting distance: The user relies on empty-handed gestures like pointing at a tool to select it. These gestures can also be mapped from their touch-based counterparts, but these gestures happen at a distance from the board.

Within sight: When the distance from the user to the board doesn't allow touch or gestures (either because of limitations in technology or accuracy of gestures at a distance), the user can use a computer device with the standard desktop metaphor, or a palm sized GIS or «sketch pad» to perform queries.

Being able to control a computer without using touch has been found to be very useful in the health industry. Surgeons who need to look through scan images while operating, would need to exit the sterile operating room to use the computer, then re-sterilize before returning to the surgery room, using valuable time. Ruppert et al. [21] implemented a system based on open source tools that allows surgeons to navigate and flip through 2D and 3D medical scan images of the patient while in the operating room.

The kinect has also been found useful in rehabilitation after injuries and training of senior citizens and people with motor disabilities. Maurer et al. [18] presents a prototype of a game that can be used in the rehabilitation process after total knee replacement surgery. They find this approach to physical therapy motivating for the patient. Also, the game can store statistics about the training that are useful for the therapist. Huang [13] and Chang et al. [8] assess the possibility to use the kinect in rehabilitation of young adults with motor disabilities. They show that the game-based approach offered by the kinect motivates the subjects, improving their motor skills while reducing the workload of the therapists.

Wilson [22] explored using a depth camera as a touch sensor. By using a remote

touch detector, in this case a kinect, instead of a touch-sensitive surface (e.g. a touchscreen), he finds that the surface doesn't need to be flat. The touch is detected by applying a threshold over the surface. If points are found within this, a touch is registered. Knowing more about the scene over the surface provides more information about the touch, allowing one to discern whether multiple touches come from the same user.

Materials

MS Kinect

The Microsoft Kinect (Figure 1) is a low-cost 3D-capable motion sensor, designed to allow users to use their bodies as controllers when playing specifically designed games on the Xbox 360 console.



Figure 1: The Microsoft Kinect sensor.

Other devices for natural interaction techniques include include the PlayStation Eye Camera [5] and the Nintendo Wii controller [4]. The kinect was chosen for this project because of its active and mature developer community as well as its relevance for photogrammetric purposes.

Kinect's great availability and low cost has made it popular in research environments and visual artist- and DIY communities. Microsoft provides its own kinect SDK [1], however, it is proprietary and only works under Windows OS. Today, many alternatives exist, some open-source. For this project, SimpleOpenNI [20] was used.

Technological specifications, motion sensing principles and accuracy

Table 1: Technological specifications of the Microsoft Kinect

Resolution	640x480 pixels (Both IR and RGB)
Frame rate	30Hz
Connectivity	USB
Practical range	Approximately 1.5m to 5m for skeletal tracking during informal testing.

The kinect measures depth using the triangulation principle [11, 15]. It consists of an infrared laser emitter (Figure 2), an infrared camera and an RGB camera. The emitter projects a beam that is split into a constant pattern of light onto the scene. The pattern is then captured by the IR camera and compared to that of a reference pattern stored in the sensor.

This principle is similar to those of a LiDAR scanner. It records live, at a frame rate of 30Hz, meaning that a developer can access and manipulate a live depth scene.



Figure 2: Taking a picture with a mobile phone of the front of the kinect while it is recording reveals the IR-beamer. The middle lens is the RGB-camera, and the one on the right is the IR-camera.

The kinect is intended to be used at a distance of between 1.2m•3.5m, but captures data beyond that and as close as 0.7m. Khoshelham and Oude Elberink [14] presented an accuracy analysis of the kinect and compared the output data to that of a high-end la-

ser scanner point-cloud. They concluded that the random error increases and depth resolution decreases quadratically with increasing distance from the sensor. The error reaches a maximum of 4cm at 5m, while the resolution is 7cm at this distance.

Processing

Processing [3] is an open source, cross-platform (Windows, Mac OS, Linux and more) Java-based language and IDE for developing creative applications that focus on visualizations, animations and interactions.

The strength of Processing is that it is fairly simple to get started, but it still offers full Java functionality and the possibility to add libraries to extend its functionality.

Libraries

In addition to OpenGL, two libraries were used for FloorMapper.

Unfolding Maps

Unfolding Maps [18] is an open-source library that adds mapping functionality, allowing for common map interactions such as panning, zooming and rotating. The maps themselves are tile-based and can be designed by the developer, using a service like TileMill [15].

SimpleOpenNI

SimpleOpenNI [20] is an open-source OpenNI wrapper for Processing. In addition to allowing Processing to read raw data from the kinect, it provides OpenNI features such as user recognition, skeleton tracking and gesture detection.

Map database

The MBTiles format [16] was chosen to store the map data. MBTiles is an open specification that stores tiled map data in an SQLite database that can be kept in a single file on a server or computer. Unfolding Maps supports reading the file locally on the computer, allowing fast rendering of the map in the application.

The tiles were created in TileMill [15]. TileMill is an open source tool that allows the user to import vector and raster data from sources such as shapefiles, kml, GeoJSON,

GeoTIFF and furthermore, style the data using a CSS-like syntax called Carto and export them to several formats.

Prototypes

Two prototypes of FloorMapper were implemented, using different concepts for interaction. They run on all the major platforms, Windows, Mac and Linux, but OpenNI and OpenNITE need to be installed on the system [20]. Common for both prototypes is the hardware setup (as seen in figure 5) and the basic ideas of projecting a map onto the ground where people can pass by, and having the map react to user interactions.

The application was designed to achieve good usability, as defined by Preece [19]. To comply with the principles, rules were defined as shown in table 2.

Virtual buttons

The interaction method used in this prototype is described by Borenstein [7]. The ki-

nect records depth points live. Borenstein defines a hotbox as box defined in space (in this case, relative to the kinect), that counts the number of points that are located within. The application is then able to start events when enough points are found within the box.

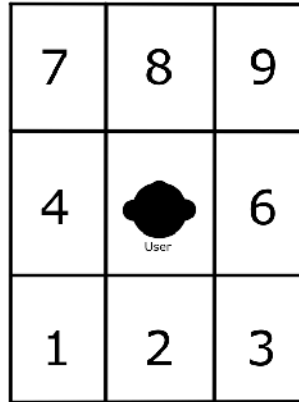


Figure 3: The numpad metaphor.

Table 2: Usability rules for FloorMapper

Principle	Description
Effectiveness	Application should have a finite set of clearly stated actions available.
Efficiency	Once learned, actions within the application should be simple to perform.
Safety	Application defines rules to avoid the user getting lost (panning outside the map area/zooming too close or far out).
Utility	Application has a limited set of options.
Learnability	Application maps gestures from other known interfaces to control the map when possible.
Memorability	Once learned, the gestures should feel natural for the user.

In the prototype, boxes were placed in the XY-plane above the map. Figure 3 shows how the eight boxes were placed on the map. As people passed over the map, the map would react and grab their attention.

The user is then instructed to stand in the middle of the map (at «number 5») where no action is happening and take a step toward any of the other «buttons» to interact with the map. This application used the metaphor of the arrow keys on the numpad to allow the user to pan the map up (8), down (2), left (4) and right (6). At the boxes corresponding to 7

and 9 on the numpad, the map would rotate left and right, while 1 would reset it.

This non-complex interaction technique is robust and relatively simple to implement. Applying the numpad-metaphor should also make the application intuitive, even for novice users. This does, however, limit the number of commands available to the user without making things overly complicated. The current application has eight buttons available. Given a larger surface area in the application, the number of buttons could be higher.

The simple implementation and commands do lend themselves well to certain kinds of use in which simple interaction is needed. For a map with a finite number of commands, e.g. placed in a public space, this could be ideal.

Complex gestures with advanced visualization

This version takes advantage of SimpleOpenNI's skeleton tracking capabilities for more natural interactions.

After a user is calibrated, the kinect is able to track all the main joints, which allows for

more natural interactions. The calibration requires the user to stand in an awkward position in front of the kinect with hands in the air, but this can be bypassed by bundling a pre-made calibration file with the application.

When a user enters the space and is automatically calibrated, to notify her that «something is happening», an area underneath her foot shows the corresponding area as an aerial photo, using a looking glass metaphor as shown in the screenshot in figure 4. This demonstrates one strength of using natural interaction to explore multiple data sources.

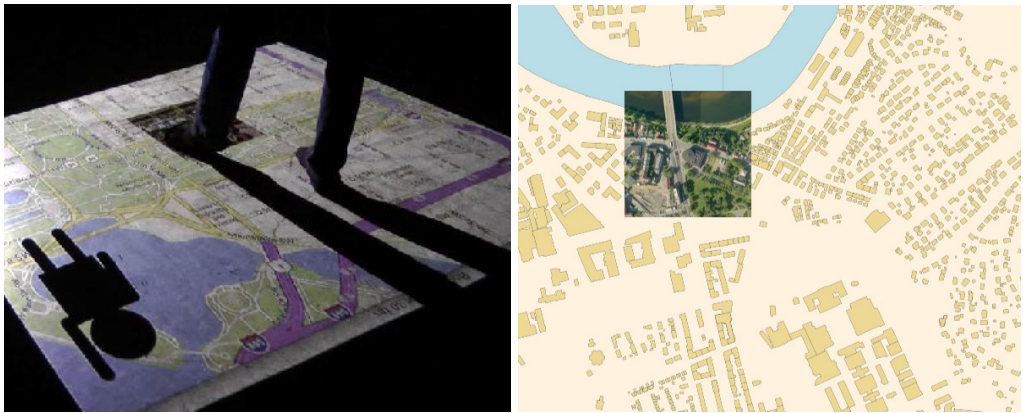


Figure 4: Area under foot displays aerial photography.

To notify the user that she can raise her arm to access a control panel, a figure, raising its arm, is displayed. The control panel displays four different tools the user can then point at to select:

- Pan: When the pan tool is active, the user either swipes with her arms or take a step towards a direction to allow the map to move.
- Zoom: This tool allows the user to zoom in or out in the map by moving her arms towards or away from each other, similar to the «pinch-to-zoom» motions, known to users of touch surfaces.
- Rotate: The rotate tool allows the user to rotate the map by holding her arms up and moving them opposite directions over the map, as if rotating a physical map.

- Place marker: When this tool is selected, a place marker is put under the user's shoe, and the coordinates of this location are stored in a text file. The idea is that users can pin their hometown or a place dear to them on the map. This data could be used later for visualizations on a web page, e.g., «Our guests at the stand came from these towns».

When the user has finished using a tool, she can always raise her arm to access the control panel again, or leave the space to end the session. The map then resets to the default location, zoom level and rotation.

This interaction method allows for more advanced control functions as well as more finely tuned control for certain commands.

The metaphors for interacting, «drag» to pan, «pinch» to zoom, with the map are already known to users familiar with touch-enabled devices. This method does, however, have some limitations as found in tests. The user calibration can be non-trivial, and implementing for multiple users can be complicated. However, if more advanced commands than those a simple button push can provide are needed, this method is useful.

It should be noted that none of these interaction methods are objectively «better» than others. They each lend well to different use-case scenarios and could be used in combination to complement each other.

Evaluation

The virtual button implementation was tested in a non-structured qualitative evaluation during Researcher's Night at NTNU in 2012. This is an event for local high school students at the campus, where the academic departments present themselves to more than 1000 prospective students from the region.

The installation was placed in an area where people would walk by and try it out. We found that the application itself works. It is robust and handled multiple users at the same time. Most users who tried it out mastered the commands quickly, with a little demonstration of what was possible.

We noted, however, that some prospective users were skeptical of entering the map area. Studies [12] have shown that the kinect has a high degree of acceptance, and it is also a popular consumer device, so people aren't «afraid» of using it. The setup of this installation may seem elaborate, and this can contribute to first-time users being wary of «just trying it out.» It could also seem complicated for users not used to working with maps, so a video demonstration could also be useful.

A common response of users interacting with the application was to try and figure out what caused the map to move. Users who hadn't noticed the kinect or were unfamiliar with it, tended to put weight on the ground where the «buttons» were located, thinking there was a sensor underneath. When pre-



Figure 5: Hardware setup of FloorMapper.

sented with the kinect afterwards, they would get an «aha» reaction. Some would then ask more about the concepts, upon which they could get more information about the sensor and be shown some source code.

Future work

The focus on the development of the prototypes in this project has been on interaction design. Even within this limited field, there are many potential areas for further work.

The prototypes only work when the map is projected to the floor, and some of the actions require that the Kinect is placed in a precise spot relative to the map. For the application to be usable in more general cases, for instance to allow the map to be larger or smaller than default, the application should be able to calibrate the map in relation to the kinect on startup.

If the map is to be projected onto a table or vertically onto a wall for collaboration or any other general cases, the interaction rules need to be generalized. Also, the application would need to find ways to handle a situation in which two or more users are meant to be in control simultaneously.

Visualization of the map has not been a focus in this project. The surface onto which the map is projected greatly affects the legibility of the map. In various tests, the map was projected onto white, gray and green floors, which clearly had different effects on the map's look. The strength (luminosity) of the projector also has a great impact on the map's appearance.

As noted in the evaluation, some users were wary of trying the prototype. This is not a problem of touch-less interfaces in general, but rather of the presentation of the installation. A future version should be made more inviting.

Touchless interfaces are still in their infancy. 2013 will see the release of a new computer-specific peripheral for motion sensing, the Leap Motion [2], that is already set to be bundled with computers from HP [6]. This, along with further developments in the kinect community, will also provide new possibilities for GIS and collaboration purposes.

The Microsoft Kinect can also be applied to point data collection, similar to a laser scanner. Khoshelham and Oude Elberink [14] conclude that the data should be acquired at a distance of 1 • 3m to achieve satisfactory results. This makes the kinect very useful for mapping indoor areas. In addition to being very low-cost, compared to commercial laser scanners, it is very portable and operates fast. It is also able to record 30 frames per second. If the resulting point clouds can be automatically merged, an indoor environment can be mapped by simply having the kinect sensor record every feature.

Conclusion

The technology represented by Microsoft Kinect provides new, interesting opportunities for human-computer interaction. During the course of this project, prototypes were developed to present some of the possibilities for controller-free interaction.

In user tests, we found that many users, though curious, were skeptical of walking into the map area and interacting with the installation. Future versions of the applicati-

on should be made more inviting, and include a demonstration of the features of the application in use.

The test did demonstrate that when the technology, already familiar to many of the users, was presented to them in a new and approachable manner, many showed an interest to learn more about it.

The source code of the prototypes is published to Github [11].

References

- [1] Download the kinect sdk & developer toolkit – kinect for windows. URL: <http://www.microsoft.com/en-us/kinectforwindows/develop/developer-downloads.aspx>.
- [2] Leap motion. web, 2012. URL: <https://leapmotion.com>.
- [3] Processing.org, September 2012. URL: <http://www.processing.org> [cited Sept 2012].
- [4] Wii official site at nintendo. web, 2012. URL: <http://www.nintendo.com/wii/>.
- [5] Playstation eye camera [online]. 2013. URL: <http://us.playstation.com/ps3/accessories/playstation-eye-camera-ps3.html> [cited April, 2013].
- [6] H. B. Hp to bundle leap motion. URL: <http://www.engadget.com/2013/04/16/hp-to-bundle-leap-motion-on-select-products-incorporate-the-tec/>.
- [7] G. Borenstein. *Making Things See: 3D vision with Kinect, Processing, Arduino, and Maker-Bot*. Make: Books. O'Reilly Media, Incorporated, 1 edition, 2012. URL: <http://makingthings-see.com>.
- [8] Y.-J. Chang, S.-F. Chen, and J.-D. Huang. A kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. *Research in Developmental Disabilities*, 32(6):2566 – 2570, 2011. URL: <http://www.sciencedirect.com/science/article/pii/S0891422211002587>, doi:10.1016/j.ridd.2011.07.002.
- [9] J. Florence, K. Hornby, and M. J. Egenhofer. The gis wallboard: interactions with spatial information on large-scale displays. Seventh International Symposium on Spatial Data Handling, pages 449–463, London, August 1996. Department of Spatial Information and Analysis, University of Maine, Taylor & Francis. URL: <http://www.spatial.maine.edu/~max/SDH-wall.pdf>.

- [10] B. Freedman, A. Shpunt, M. Machline, and Y. Arieli. Depth mapping using projected patterns, 2012.
- [11] J. W. Haakseth. Github – haakseth. Web, 12 2012. URL: <https://github.com/haakseth/FloorMapper>.
- [12] A. Hassani. Touch versus in-air hand gestures: Evaluating the acceptance by seniors of human-robot interaction using microsoft kinect., December 2011. URL: <http://essay.utwente.nl/61245/>.
- [13] J.-D. Huang. Kinerehab: a kinect-based system for physical rehabilitation: a pilot study for young adults with motor disabilities. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, ASSETS '11, pages 319–320, New York, NY, USA, 2011. ACM. URL: <http://doi.acm.org/10.1145/2049536.2049627>, doi:10.1145/2049536.2049627.
- [14] K. Khoshelham and S. Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12:1437–1454, 2012. doi:10.3390/s120201437.
- [15] MapBox. Tilemill – fast and beautiful maps. Web. URL: <http://mapbox.com/tilemill/>.
- [16] MapBox. Mbtiles specification, 2011. URL: <https://github.com/mapbox/mbtiles-spec> [cited April, 2013].
- [17] B. Maurer, F. Bergner, P. Kober, and R. Baumgartner. Improving rehabilitation process after total knee replacement surgery through visual feedback and enhanced communication in a serious game. In *Proceedings of the 30th ACM international conference on Design of communication*, SIGDOC '12, pages 355–356, New York, NY, USA, 2012. ACM. URL: <http://doi.acm.org/10.1145/2379057.2379124>, doi:10.1145/2379057.2379124.
- [18] T. Nagel. Unfolding maps: Unfolding is a library to create interactive maps and geovisualizations in processing and java. URL: <http://unfoldingmaps.org>.
- [19] J. Preece. *Interaction design : beyond human-computer interaction*. J. Wiley & Sons, New York, NY, 2002.
- [20] M. Rheiner. Simple-openni - google code. web. URL: <https://code.google.com/p/simple-openni/>.
- [21] G. C. S. Ruppert, L. O. Reis, P. H. J. Amorim, T. F. Moraes, and J. V. L. Silva. Touchless gesture user interface for interactive image visualization in urological surgery. *World Journal of Urology*, 30:687–691, 2012. URL: <http://dx.doi.org/10.1007/s00345-012-0879-0>, doi:10.1007/s00345-012-0879-0.
- [22] A. D. Wilson. Using a depth camera as a touch sensor. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, pages 69–72, New York, NY, USA, 2010. ACM. URL: <http://doi.acm.org/10.1145/1936652.1936665>, doi:10.1145/1936652.1936665.