KNOWLEDGE AT YOUR FINGERTIPS
Multi-touch Interaction for GIS and Architectural Design Review Applications

Yvonne Jung, Jens Keil, Harald Wuest, Timo Engelke, Patrick Riess and Johannes Behr
Fraunhofer IGD/ TU Darmstadt, Fraunhoferstrasse 5, 64283 Darmstadt, Germany

Keywords: Multi-touch, Multi-user, Tabletop interaction, Design review, Tracking, Visualization, X3D.

Abstract: This paper introduces novel techniques of interacting and controlling 3D content using multi-touch interaction principles for navigation and virtual camera control. Based on applications from GIS and for the architectural design review process, implementation and usage of these interaction techniques are illustrated. A comprehensive hardware and software setup is used, which not only includes tracking, but also an X3D based layer to simplify application development. Therefore it allows designers and other non-programmers to develop multi-touch applications very efficiently, while allowing to focus on user interaction and content.

1 INTRODUCTION

Multi-touch technology is one of the most interesting fields of research in today's Human-Computer-Interface area and interactive direct-touch tabletop displays have been the focus of numerous research projects. It is not only about interacting with more than one or two fingers simultaneously, but also working together collaboratively in a multi-user scenario. Multi-touch techniques seem to be very promising since they allow people to seamlessly interact with what they see by simply touching it. Thus, new ways of generating immersion are possible, like for instance through tabletop interaction.

For most people, working on a table feels well known and very intuitive, since they are used to work on and surround tables every day. That is especially of concern when working with a lot of maps or blueprints. These are normally plotted on huge papersheets, spread on a table's surface and discussed by several people. However, many people are not capable to read and understand abstract maps or blueprints decoded with a huge amount of information, hence having problems to understand them.

By using both, blueprints or maps on the one hand and their real-time rendered 3D representations on the other hand, the benefits can be combined. Regarding an architectural design review process, one can have the overview given by the blueprint and also the 3D representation of the building itself. Hence, an architect now might achieve a higher and more sophisticated comprehension of what he is looking at. The proposed concepts of how to interact and navigate in such a multi-touch environment are exemplarily discussed by introducing a GIS application as well as an architectural design review application.

Our multi-touch table (Figure 1) is based on the well-known frustrated total internal reflection (FTIR) setup (Han, 2005). It is an optical method, that tracks the user's fingers with computer vision techniques. Therefore, the acrylic sheet of the touch table is illuminated by infrared light. Because of the refractive index of acrylic relative to air, the light is subject to total internal reflection. Whenever an object comes close enough, the total reflection is frustrated, light dissipates, and illuminates the object. Thus, a fingertip touching the surface is illuminated as well.

A camera, observing the table's surface from below, now can capture the resulting lightblobs of the fingers. In our setup we use a standard IDS uEye camera with a resolution of 752x480 pixels. To improve immersion, we chose a monolithic and reduced design of the table, with a huge display of about 150x90cm, which offers enough space for collaboration. The image is created by a standard projector embedded inside the table, with a resolution of 1400x1050 pixels.

Jung Y., Keil J., Wuest H., Engelke T., Riess P. and Behr J.
KNOWLEDGE AT YOUR FINGERTIPS - Multi-touch Interaction for GIS and Architectural Design Review Applications.
DOI: 10.5220/0001814803870392
In *Proceedings of the Fourth International Conference on Computer Graphics Theory and Applications (VISIGRAPP 2009)*, page
ISBN: 978-989-8111-67-8

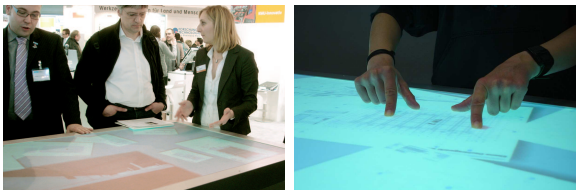

Figure 1: Architectural design review on our multi-touch table, presented at this year's CeBIT trade fair in Hanover.

2 RELATED WORK

Multi-touch technology can be seen as the basis for a lot of new techniques in the way, we are working with computers. However, it is obvious that also the corresponding applications may become increasingly complex and different subjects have to be taken into account, like finger tracking and gesture recognition, software setup, and also more sophisticated graphical interfaces and interaction principles as well. With his FTIR setup (Han, 2005), Han made multi-touch research interesting and public again, since this optical finger detection method is not only low-cost, but also easy to implement, and a robust setup for tabletop interaction and Wall-like installations (Han, 2006).

Also Microsoft's Surface (Microsoft, 2008) and Apple's iPhone (Apple, 2008) are using this kind of technology for bridging the gap between advanced graphical user interfaces and today's still limited interaction principles. Hence, in order to overcome the old WIMP paradigm that is still common for desktop applications, in (Agarawala and Balakrishnan, 2006) interesting new physical based desktop metaphors like stacks of documents are discussed.

In (Jordà et al., 2007) the collaborative generation process of live music performance on tabletop interfaces is explores. Their 'reacTable' is a special tabletop interaction device, which mainly uses tangibles for interaction that are equipped with special markers. With the 'DiamondTouch' in (Dietz and Leigh, 2001) a multi-touch table based on a capacitive system for finger detection was presented, which is mainly developed for multi-user scenarios and collaborative working. The table itself features touch and gesture based interaction techniques. Using so-called antennas, it is also able to distinguish between different users.

In (Rekimoto, 2002) a sensor architecture for hand and finger detection was introduced. It is sensitive to human hands, finger gestures, and shapes. The system thus can recognize a user's entire arm. Rekimoto is also one of the first ones, who showed basic techniques to simultaneously translate, rotate and scale 2D objects, and who presented gestures for object manipulation. A similar approach for continuously transforming objects in order to control Google Earth is also described in (Kim et al., 2007), whereas in (Wu et al., 2006) a set of design principles that support multi-hand gestural interaction is introduced.

In (Moscovich and Hughes, 2006) simple transformations and form-based manipulations of digital objects are described. However, the focus is on special mouse-cursor-like techniques. A good overview on tabletop interaction techniques in general can be found in (Shen, 2007). Recently in (Jung et al., 2008) multi-point interaction was introduced to the X3D standard (Web3DConsortium, 2007), which also provides some extensions for GIS (Web3DCon., 2008), by extending its pointing sensor component.

As can be seen, multi-touch applications show a wide variety of different interaction methods and gestures, which map fingertips and movement to different system functionalities. But in most cases they only fit to the special purpose of a task or tool. Designing the application and its interaction principles becomes often specific and tool-dependent, since there are no basic rules or guidelines, yet. However, devices like Apple's iPhone introduce quasi standards, i.e. low-level gestures, for (multi-) user interaction. Because these techniques scale well, multi-touch in addition inherently implies multi-user functionality: on bigger systems it is now possible for many users to interact with an application simultaneously. So, after a short review of vision techniques, in this paper it will be shown, how these methods also can be used for interacting in and with 3D environments.

3 BLOB TRACKING

The task of the tracking process is to detect the illuminated finger tips in the grayscale picture of the video camera, and to generate events if a finger is touching, moving or disappearing from the surface. The finger tracking procedure can be divided into two problems: blob detection and blob tracking. Blob detection handles the recognition of bright spots in the image, which results in a set of 2D image position of fingertips. The detection of blobs in images has been widely used for detecting active markers. Recently, such methods have been applied to detect fingertips on multi-touch surfaces (Touchlib, 2008).

The first step of the blob detection consists of an adaptive background subtraction step, which produces a foreground image consisting only of bright regions not present in the background model. From the many background subtraction techniques which exist (Piccardi, 2004), a simple running average turns out to be most suitable for our demands. In every frame

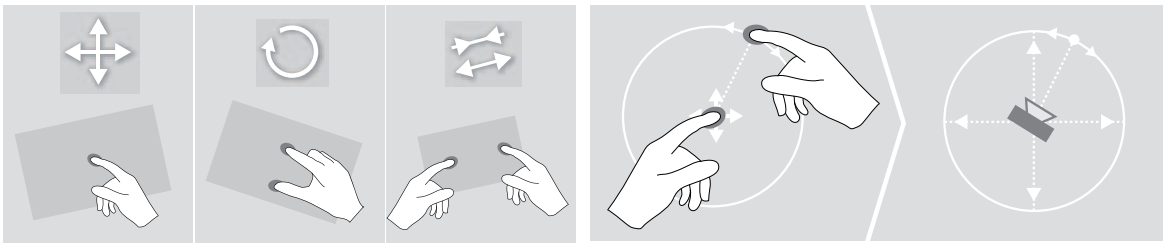

Figure 2: Illustration of used gestures. a) General interaction principles: (i) one finger selects and drags an object, (ii) two fingers rotate an object or (iii) manipulate its scale by stretching the fingers apart. b) Camera control: (i) the first finger maps to the camera's position, (ii) a second finger in addition can control the camera's orientation.

the value of every pixel of the background model is updated by $B_t = \alpha I_t + (1-\alpha)B_{t-1}$, where B_t is the intensity of a background pixel at time t, I_t is the pixel intensity of the camera image at time t and α is the learning rate. For a fast adaption during the startup a second learning rate has been introduced, which is only applied during the first seconds. At this time no detection will occur. A pixel is regarded as part of the foreground, if $|I_t - B_t| > c$, where c is a fixed threshold. The intention of detecting blobs in a foreground image is to avoid the detection of false positives, which can occur from other light sources.

To detect blob positions in the foreground image we apply a two-pass algorithm (Rahimi, 2001) to label connected components of pixels in a 4-neighborhood. For every region of connected components, the mean position, the mean intensity, the number of contributing pixels and the spacial covariance is computed. To disregard blobs, which do not originate from fingertips, some tests like analyzing the roundness or the number of pixel are performed. All blobs passing the tests are regarded as detected fingertips and are used as input for the tracking step.

For a meaningful interaction, the blobs not only have to be detected, but also tracked from frame to frame. Blob tracking assigns a unique ID to each blob and tracks it from frame to frame. Thereby the movement of a finger is detected. Since several blobs can be detected in an image at the same time, a sophisticated association of blob positions in a continuous image stream beyond a simple nearest neighbor assignment is also needed. In order to be able to find the corresponding blobs, the first approach is to look near the old position. Depending on the frame rate of the camera the finger might be found there. If the finger is moving faster than this distance per frame, the new position can be approximated using the velocity. Sometimes "tracking holes" can occur due to physical stick-slipping of the finger. In those cases it vibrates while moving over the surface and thus might not emit light toward the camera. Usually the finger "returns" in the next frames. The position therefore will be hold at the last position, but furthermore is estimated the next n frames taking the former velocity and acceleration into account. When the blob returns, the ID will be reassigned, if not, it will be discarded.

When holding the fingers down on the surface and keeping them in one position, another problem lies in the dynamic background subtraction, which adapts the finger to the background. This results in a loss of blobs when holding down for a longer period of time. To prevent this effect, every blob that does not move within a certain distance and frames gets a 'fixed' status. This status flag can be used to generate a binary mask that helps the dynamic background subtraction to ignore those areas in adaptation. Another advantage of this flag lies in the ability to prevent jittering, which significantly increases the user experience.

Because the positions of tracked blobs are detected in the image space of the camera image, it is desirable that the 2D positions of the blobs are transformed into the image space of the projector image. This transformation can be achieved by a simple homography H. If $u = (u_x, u_y, 1)$ is the homogeneous coordinate of the detected blob position, this position can be transformed into the projector image space by $v = H * u$. To determine the homography H, the 4-point method based on the Direct Linear Transform algorithm is used. As many cameras with a wide field of view come along with a strong radial distortion, it is necessary to consider these effects, too. If x is an undistorted point and $\tilde{x}$ the distorted point, we approximate the radial distortion by $\tilde{x} = x(1 + k_1 r^2 + k_2 r^4)$, where $r = \|x\|$ is the distance from the image center and k_1 and k_2 are distortion parameters.

4 DESIGN PRINCIPLES

Navigating through a 3D view of a building or landscape means controlling the virtual camera's position and orientation. But often people still have problems to keep control while navigating with common VR interaction devices, and thus, they usually have diffi-

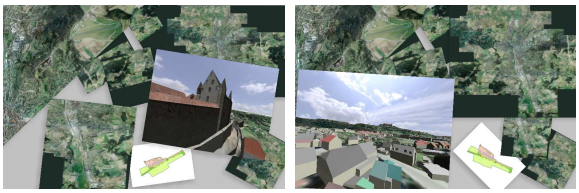

Figure 3: Interactive visualization of GIS data (different types of 2D maps as well as the corresponding buildings and the terrain in the 3D view) on the multi-touch table; recently shown at the Expo Real 2008 trade fair in Munich.

culties to fulfill typical reviewing tasks, like moving the camera around a 3D object while keeping it in focus, or moving it along a building's facade. With e.g. a space mouse a user has all degrees of freedom he might need to take a look around. But controlling the camera is still an issue and one has to be experienced in using this device. So, instead of simply navigating and concentrating on the review process, users are distracted by handling the interaction device.

4.1 Simple Gestures

In typical design review applications the user is interacting with only one 3D scene that contains the objects to be reviewed. But as mentioned 3D interaction is still neither very intuitive nor does this type of application allow to directly compare an abstract 2D information with its concrete realization in a 3D digital mock-up. Thus, to overcome the aforementioned problems, in our example applications we therefore decided to combine 2D maps and blueprints with their corresponding 3D rendering, utilizing multi-touch interaction for navigation. Figures 3 and 4 show two example applications from GIS and architecture.

Instead of only having one 3D rendering we are actually using two. The first one is used for rendering the virtual drawing table that contains the maps or blueprints with which the users can interact very similar to the way they are used to it from real life experience as shown in Figure 1. The second rendering displays the 3D scene to be reviewed on another object that also lies on the virtual drawing table, just like a window into a second virtual world. A more detailed explanation can be found in section 4.3. To increase the ease of use, we chose well known low-level gestures as mentioned in section 2. We tried to keep the amount of simultaneously needed fingers as small as possible, since one can not control and use several fingers very well at a time, due to physical and cognitive issues (Schieber and Santello, 1996).

The focus of most tasks relies on the forefingers. Using one finger touching an object a user can easily select and move it around. While stretching apart two fingers, the map can also be scaled/ zoomed and rotated by the angle described by the two fingers (see Figure 2a). We do not distinguish whether one uses fingers of the same or of different hands, as we do not distinguish between different users. There exist a lot of ideas of how to manage workspaces and tasks in multi-user scenarios, as e.g. described in (Shen, 2003; Shen et al., 2004), but for our proposed scenarios special kinds of application-based multi-user functionalities are neither needed nor useful. Hence, all users can work in parallel and generally can use the whole area of the table the same way, and also perform actions together like zooming or dragging. Users can choose collaboratively, what might fit to their needs or to the purpose they want to perform.

4.2 Virtual Camera Control

The virtual camera can be controlled by simply positioning the fingers onto the table's surface. One first finger touching a map or blueprint controls the position of the camera. Using then a second finger defines the direction to look at, and controls thus the orientation of the camera. Again, one can use fingers of one hand or of both hands. Mapping camera parameters onto the user's fingers supports him performing navigation tasks, because now there is a pretty direct representation for the camera through the fingers, thus helping to orientate oneself and to control navigation within the 3D world. As can be seen in Figure 2b, by moving one finger around the other, the camera of the reviewing scene moves around the corresponding point, while keeping the touched object in focus.

Furthermore, by interpolating smoothly between succeeding camera poses, this offers a fluent and continuous camera movement comparable to cinematic camera movements. Especially in combination with our second method for calculating the viewing direction, by simply orienting the virtual camera along the direction of finger movement, this is, unlike using artificial interaction devices, a very intuitive and direct means for navigating in a virtual environment. By also updating the position and especially the height of the virtual camera per texel according to the given heightfield of the maps, as shown in Figure 3, realistic terrain following can be implemented as well and leads to a much better experience than the tinkered sandbox landscapes one might remember from school. Another important aspect is, that by scaling a map, implicitly the velocity of camera movement is also scaled. Thus, when zooming into a map the corresponding camera movement slows down and automatically adapts to the scene, because within a certain time Δt the path length d of the finger movement in the map's normalized image space is smaller when

Figure 4: The design review application. a) Illustration of fingers and camera orientation (depicted by green blobs). b) Each blueprint on the table corresponds to one level of the building, whose rendering is, similar to a photo, part of the table scene. c) Closeup: the application provides basic visual feedback (fingers/ blobs are denoted by colored circles and numbers).

the image is scaled larger and vice versa.

4.3 Examples and Discussion

The mentioned architectural design review application shown in Figure 4 was developed for Messe Frankfurt and shows its new exhibition hall. The application consists of some transformable architectural 2D blueprints of several floors of the building and a high quality rendered 3D view of the hall, which changes according to the area that was touched in one of the blueprints (with a third and fourth finger as visualized in Figures 2b and 4a). Several people can move and zoom these plans simultaneously, but also navigate through the model of the building.

As previously described, using blueprints for navigation is one of the key features of this multi-touch application, and additionally an imitation and digital enhancement of an architect's native work environment. Working on a touch table is just like they know it from daily work, except that the application shows a valid real time visualization of the whole building. Since building plans are usually decoded with special symbols and a high amount of information, which more or less only professionals can read and imagine, the user now can see and "grasp" the high resolution rendering of the blueprint he is looking at.

With one or two fingers the user now simply moves the plan around. While stretching apart two fingers, the blueprints can be scaled and also rotated. The tile showing the 3D visualization lies on the virtual table next to the blueprints, and is also modeled as a textured plane, and thus can be transformed the same way as the blueprints. But whereas their textures are static, the texture displaying the architectural visualization has to be dynamically rendered every frame in a first rendering pass according to the movement of the virtual camera. Architects can thereby move, rotate and scale the 3D visualization like any other plan. For the ease of use, all selected objects come to the top in the order they have been touched.

As shown in section 4.2, the view is controlled by using two additional fingers. To distinguish between various floors or areas and to select them for navigation, first two fingers of the left hand for instance can grab a plan and select it, just like one would hold a sheet of paper while writing on it. Then, a finger of the other hand points onto the construction plan and thus controls the camera's position. Moving around this finger and using a second finger of the right hand defines the viewing direction (see Figures 2b and 4). The GIS application follows the same interaction metaphor with the exception that each map corresponds to a different area of the landscape instead of different floors, and thereby the viewing height is not set per floor plan but per touched texel.

People who used these applications accepted very well the way of selecting objects, and it felt quite familiar to them. Also, none of them had further problems in remembering or using the described gestures. But there are still some issues: although the paper-sheet metaphor is doing fine, the gestures' usage has to be explained. One also has to "fix" a plan during the whole navigation-process and has to work with three or four fingers. While taking just a look around works well, doing more sophisticated movements is a bit complicated: when the third finger points onto a certain object on the plan and the fourth finger shall move around it, there is a physical limitation of hand- and finger-movement, making it hard to move around an object while keeping it in focus. Thus, by providing a default direction based on the direction of movement, typical tasks like walking along a road, which might be of interest in the GIS application shown in Figure 3, can be accomplished more easily.

Nevertheless, the result is a much better perception and understanding of otherwise arbitrary looking data – not only for the design and planning stage but also for later presentation. Both applications are created and setup using X3D and the InstantPlayer (Avalon, 2008) for rendering and interaction. Not only all the plans but also the 3D view are modeled as textured "Plane" geometries in X3D. This way, translation, rotation and scaling of plans and maps work

like in common multi-touch applications as already described in section 4.1. By using special extensions of the standard X3D pointing sensor component, like the new ”HypersurfaceSensor“ node that is described in more detail in (Jung et al., 2008), it is thereby easy to design and implement multi-touch applications.

5 CONCLUSIONS

In this paper we have presented two applications as an example of how tabletop interfaces and multi-touch can be used in 3D environments. We have exemplarily explained some concepts of how to interact and navigate in multi-touch environments, which might also be useful for other areas like shipbuilding or vehicle construction. We have shown that multi-touch interaction can be both, interacting with more than one or two fingers simultaneously, and also working collaborative in a multi-user scenario. Moreover, we have also pinpointed some problems concerning blob detection and tracking and how to overcome them. Multi-touch techniques are very promising since they allow people to seamlessly interact with what they see by simply touching it. It feels natural, and can lead to more sophisticated interaction principles. Therefore, new ways of generating immersion are possible.

Future work will focus on a more intuitive interaction for the application’s camera movement functionality. Furthermore, we would also like to integrate generic gesture recognition for e.g. selecting and fixing a blueprint. As discussed, the user then does not have to keep an almost static position of his fingers during the navigation process, but can have some kind of initiation and relaxation phase as described in (Wu et al., 2006) while working with the application.

REFERENCES

Agarawala, A. and Balakrishnan, R. (2006). Keepin’ it real: pushing the desktop metaphor with physics, piles and the pen. In *CHI ’06: Proc. of SIGCHI conf. on Human Factors in comp. sys.*, pages 1283–1292. ACM.

Apple (2008). iphone. http://www.apple.com/iphone/.

Avalon (2008). Avalon. http://www.instantreality.org/.

Dietz, P. and Leigh, D. (2001). Diamondtouch: a multi-user touch technology. In *UIST ’01: ACM symposium on User interface software and technology*, pages 219–226, New York, USA. ACM.

Han, J. Y. (2005). Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST ’05: ACM symposium on User interface software and technology*, pages 115–118, NY, USA. ACM.

Han, J. Y. (2006). Multi-touch interaction wall. In *SIGGRAPH ’06: ACM SIGGRAPH 2006 Emerging technologies*, page 25, NY. ACM.

Jordà, S., Geiger, G., Alonso, M., and Kaltenbrunner, M. (2007). The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In *TEI ’07: Int. conf. on Tangible and embedded int.*, pages 139–146, NY. ACM.

Jung, Y., Keil, J., Behr, J., Webel, S., Zöllner, M., Engelke, T., Wuest, H., and Becker, M. (2008). Adapting X3D for multi-touch environments. In Spencer, S., editor, *Proceedings Web3D 2008: 13th International Conference on 3D Web Technology*, pages 27–30, New York, USA. ACM Press.

Kim, J., Park, J., Kim, H., and Lee, C. (2007). Hci(human computer interaction) using multi-touch tabletop display. *Communications, Computers and Signal Processing, 2007. PacRim 2007.*, pages 391–394.

Microsoft (2008). Microsoft surface. http://www.microsoft.com/surface/.

Moscovich, T. and Hughes, J. F. (2006). Multi-finger cursor techniques. In *GI ’06: Proc. of the 2006 conference on Graphics interface*, pages 1–7, Toronto, Canada. Canadian Information Processing Society.

Piccardi, M. (2004). Background subtraction techniques: a review. *Systems, Man and Cybernetics, 2004 IEEE International Conference*, 4:3099–3104.

Rahimi, A. (2001). Fast connected components on images.

Rekimoto, J. (2002). Smartskin: An infrastructure for freehand manipulation on interactive surfaces. In *SIGCHI conf. on Human factors in computing systems*, pages 113–120, Minneapolis, USA.

Schieber, M. and Santello, M. (1996). Hand function: Peripheral and central constraints on performance. *Journal of Applied Physiology*, (6):2293–2300.

Shen, C. (2003). Ubitable: Impromptu face-to-face collaboration on horizontal interactive surfaces.

Shen, C. (2007). Interactive tabletops. In *ACM SIGGRAPH 2007 courses*, pages 40–53, NY, USA. ACM.

Shen, C., Vernier, F. D., Forlines, C., and Ringel, M. (2004). Diamondspin: an extensible toolkit for around-the-table interaction. In *CHI ’04: Proc. SIGCHI conf. on Human Factors in comp. sys.*, pages 167–174. ACM.

Touchlib (2008). A multi-touch development kit.

Web3DCon. (2008). X3D-Earth. http://x3d-earth.nps.edu/.

Web3DConsortium (2007). *Extensible 3D (X3D)*. http://www.web3d.org/x3d/specifications/ISO-IEC-FDIS-19775-1.2/index.html.

Wu, M., Shen, C., Ryall, K., Forlines, C., and Balakrishnan, R. (2006). Gesture registration, relaxation, and reuse for multi-point direct-touch surfaces. In *TABLETOP ’06*, pages 185–192, Washington, DC, USA. IEEE.