universitätbonn

Rheinische
Friedrich-Wilhelms-
Universität Bonn

Fraunhofer
IAIS

Master Thesis on

Using the Microsoft Kinect sensor for improving multi-touch interaction in the context of geological analysis

Yuelong Yu

First referee:	Prof. Dr. Reinhard Klein
Second referee:	Prof. Dr. Andreas Weber
Submission Date:	October, 2011

Declaration of Originality

I declare that I have finished this thesis independently. All the work is original. All the citations from other authors have been marked in the thesis.

Yuelong Yu
University of Bonn

Acknowledgements

I would like to thank all those who supported me during this thesis work. First, I would like to thank Professor Dr. Reinhard Klein from the University of Bonn who approved this thesis topic to be my first referee. My thanks go to Professor Dr. Andreas Weber from the University of Bonn to be my second referee as well.

Special thanks to all my colleagues in the VRGeo project of the Fraunhofer Institute for Intelligent Analysis and Information Systems. Especially, I would like express my sincere gratitude to Dr. Markus Schlattmann who gave me a lot of suggestions and guided me to finish the whole thesis work. My thanks also go to Dr. Manfred Bogen and David d'Angelo. They helped me a lot during this thesis work as well.

Finally, I wish to thank my parents and my girlfriend Xiye Zhou for supporting me all the time.

Abstract

Multi-touch is a popular technology which has been widely used. There is a lot of hardware supporting multi-touch applications in various domains.

The contribution of this thesis is presenting a context-aware tabletop system based on the Microsoft Kinect. In this system, the users can work on the multi-touch table simultaneously and collaboratively in different contexts. For this purpose, the Kinect is used as an additional sensor to track the users around the multi-touch table. Based on the information of the Kinect, a new method for detecting the user context of touch points on a tabletop is presented. This method comprises automatic calibration, combined segmentation by using the depth and infrared information of the Kinect, user tracking and association of touch points with individual users. With this method, each detected touch point is associated with an individual user.

By taking advantage of the user information for detected touch points, new functionalities including individual tool selection, locking mechanism and user-dependent annotations are implemented for the new context aware system. With these new functionalities, each tool is associated with each user, and the annotations from different users are distinguishable. During the interaction, the locking mechanism is able to decrease the interference between users. Therefore, users can work collaboratively in different contexts.

Contents

List of Figures

1 Introduction

Multi-touch enables people to use only their fingers to control an application by touching the screen. It is not an entirely new technology. A lot of multi-touch systems have been implemented since the late 1960s. Nowadays, as the price of hardware equipment decreases, the multi-touch technology is more widely used than ever before, because compared with the mouse and the keyboard, it provides a novel interaction experience between the user and the application. There is a lot of hardware supporting multi-touch applications in different domains. For example, mobile devices (like iPad) or touch tables (like Microsoft surface) are used for countless applications.

Due to the limitation of the size and the sensors of mobile devices, it is difficult for multiple users to work on such a device simultaneously and collaboratively. Large tabletops can overcome these disadvantages, several users can work on a tabletop system collaboratively, and they can also talk face to face with each other while they exchange ideas with each other. Besides, the large tabletop is also better at presenting, comprehending and interpreting rich and complex data than mobile devices. Therefore, during the last years, lots of large interactive tabletop devices have been developed in some domains where collaborative work between users is highly required. For example, in scientific research areas, where complex data is assessed, several experts often need to work together in order to get a proper solution.

The technology that is used for large interactive tabletop devices is typically projection based (e.g. [6]). This limits the flexibility of the devices, as the projection cone of a projector requires extra space and a certain arrangement with the screen surface. Besides, designing a table with adjustable angle of inclination, adjustable height and pivot function is nearly impossible. Therefore, when multiple users are working on such a table, it is possible to increase the risk of Repetitive strain Injury (RSI) [39]. Further issues involve the heat and the noise that projectors generate, particularly when using high-performance or multiple projectors, as required for increased image quality and resolution [17].

Another problem in multi-touch systems is missing context information. If a multi-touch system detects two touch points on the screen, the system itself normally cannot distinguish whether the touch points are from one hand, two hands, or even two different users. Therefore, multiple users and multiple hands can only work in the same context, which often leads to interference [29, 12, 21]. In order to solve this problem and thus enable more natural interaction, some previous research systems have already been implemented by including additional environmental sensors (e.g. a ceiling mounted camera) [8,7,32]. However, these systems all suffered from severe limitations, either restricting the surrounding of the tabletop or even the movements/locations of the users themselves.

In order to enable more natural interaction, a high fidelity multi-touch table with a depth camera as an additional sensor to achieve reliable and robust distinction and tracking of different users touching the screen surface is equipped in this thesis project. The Microsoft Kinect sensor has been chosen as it provides both depth and color information. Moreover, its depth image has currently the highest image resolution and low noise.

The main goal of this thesis is the development of a context-aware tabletop system which is capable of user tracking and distinction, and which enables multi-users to work collaboratively and simultaneously on the multi-touch table in a different context.

Based on the information of the Kinect, a new method for detecting the user context of touch-points on a tabletop is developed. This method comprises automatic calibration, combined segmentation by using the depth and infrared information of the Kinect, user tracking and association of touch points to individual users.

With the user information for detected touch points provided by the user-tracking application, many new possibilities for improving the expressiveness of multi-touch gestures and realizing software-supported multi-user multi-touch input coordination can be integrated into an existing application. Particularly, in this thesis, the individual tool selection and user-dependent annotations are integrated into an existing application that was developed at the Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS).

2 Previous Work

2.1 Direct Collaboration at the Tabletop

As mentioned in the previous chapter, tabletops have advantages in collaborative user interaction. The users can communicate face to face with each other while they exchange their ideas and the objects on the shared multi-touch table. Therefore, this setup has long been proposed also for computer mediated collaboration (e.g. [6, 31, 30, 25, 27, 24, 22, 36]). But developing appropriate user interfaces is a challenging task. For example, for common user actions like maximizing a GUI-Element, direct access to the full screen is beneficial for a single user. However, for multiple users, this design leads to interference between users. The power and expressiveness of well established interaction patterns for graphical user interfaces easily evoke conflicts in collaborative settings [29, 12, 21].

A comparison between indirect input from multiple mice and multi-touch input was done by Hornecker et al. [11]. During the experiment, they observed that the multi-touch condition has good awareness, but it also leads to more mutual interference. A comparison shows that direct multi-touch allowed more fluent collaboration. Although interferences occurred more frequently in multi-touch conditions, they could rapidly be negotiated and solved by the users.

Scott et al. [26] and Tse et al. [34] observed that territoriality is a main factor for the coordination when several users share one interaction space. However, for a multiple user activity, suitable negotiation strategies for avoiding interference are still required. Based on territoriality, a set of document sharing techniques are proposed by Ringel et al. [22]. Morris et al. [19] developed some higher level coordination strategies for avoiding interference between users. Besides, some cooperative gestures for multi-user tabletops are also developed by Morris with another team [18]. However, those coordination techniques cannot totally be implemented on multi-touch sensors. For user identification, further context information is required.

Benko [38] developed a system to explore the feasibility of expanding the interaction possibilities on interactive surfaces. In this system, they place electromyography (muscle activity) sensors on the forearm to infer finger identity, estimate finger pressure, and allow off-surface gestures (see Figure 1).

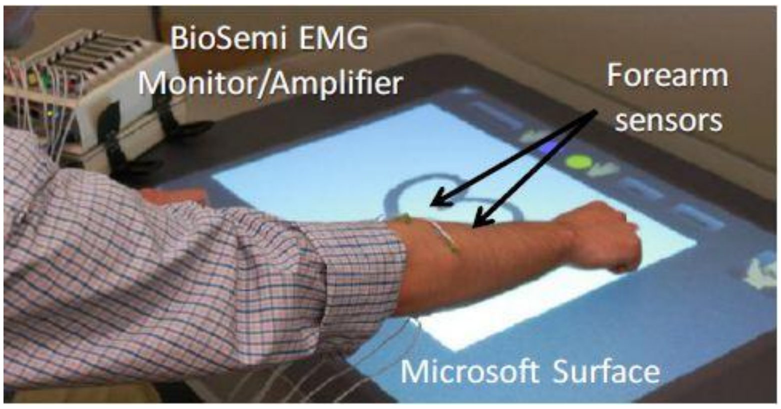

Figure 1: The muscle sensing system [38].

Marquard et al. [16] recently demonstrated the benefits of a robust association between touch-points and the hands of the users. They use a glove which is equipped with unambiguous optical markers. With this glove, the relation between fingers can be detected. By taking advantage of the detected finger relation information, hand gesture recognition, multi-user coordination policies, and a diversity of drawing tools that could be associated with individual fingers were implemented by them (see Figure 2).

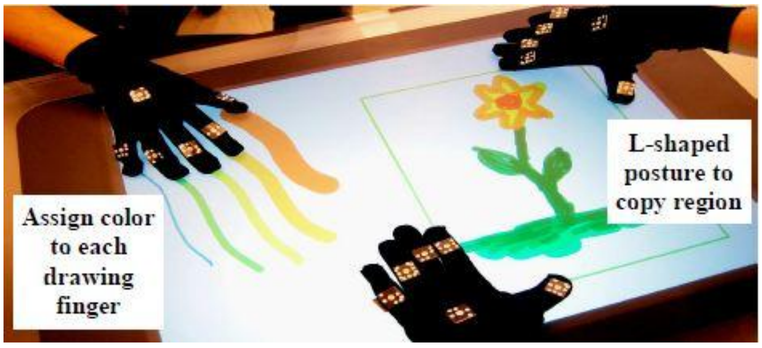

Figure 2: Interactive drawing application on the digital tabletop, using the identity of touches for interaction [16].

In the next section, the advantages and drawbacks of current existing multi-touch systems offering context awareness are discussed.

2.2 Context-Aware Multi-Touch Systems

DiamondTouch is a commercially available multi-touch sensor device [6]. In this system, a ceiling-mounted video projector displays onto a table. The table is equipped with signal emitters and there is a unique receiver on every user's seat. When a user touches the table, a capacitive coupled circuit is completed.

The user is detected by the respective receiver [6] (see Figure 3). Thus, the system can associate the touch point with the individual user. Many researchers implemented multi-user coordination policies using this system (e.g. [6, 19, 18, 22]). Unfortunately the system limits the choice of display components to front projection. Furthermore, the system only supports up to four users and requires reasonable electrical isolation between the users. Two users (or their chairs) are not allowed to touch each other or to be in very close physical proximity [6].

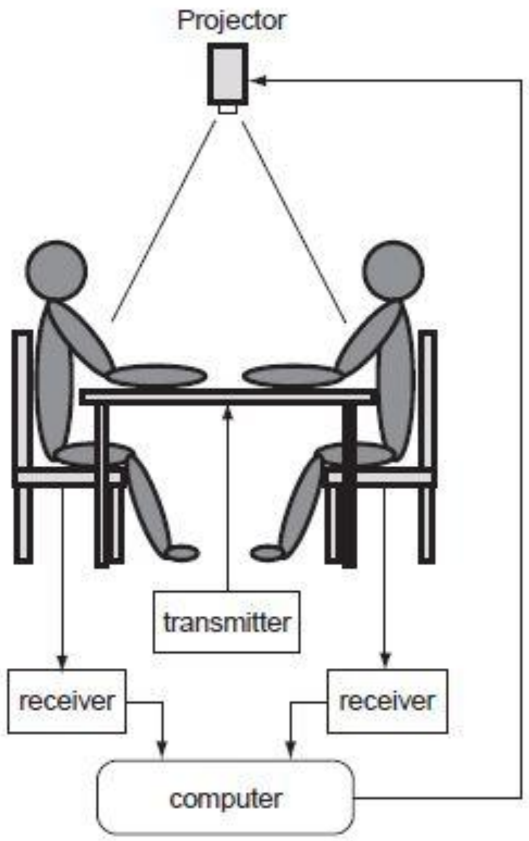

Figure 3: Structure of the DiamondTouch System [6].

Frustrated total internal reflection (FTIR) technique [9] utilizes the phenomenon of the total internal reflection; normally the lights emitted by the LED light are totally reflected between two layers. When a finger touches the screen, the total internal reflection is frustrated and the lights are escaped from the layer. By using the video camera, the lights scattered by the finger can be detected. Thus, the touch point can be detected. See Figure 4 for an illustration.

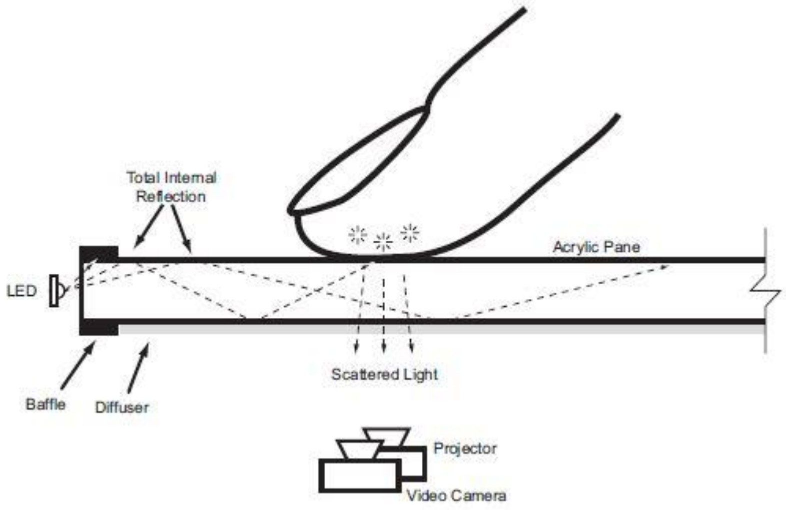

Figure 4: The FTIR technique [9].

Based on the FTIR techniques, another type of system is described in [7]. For user detection in this system, an extra webcam is used being located above the FTIR tabletop display. Using this webcam, the hands are tracked on or above the table using skin color segmentation. By associating touch points with individual users, the system can support multi-users (see Figure 5). However, for skin color segmentation, the background has to be controlled, which means that the colors from movement parts (foreground) must not be on the background (e.g. the chair, floor and the display itself) [7]. Therefore, the authors suggest cancelling the light from the screen with polarization filters to avoid interference with the color of displayed items. As an alternative, they propose tracking the dark silhouettes of the hands above the illuminated screen.

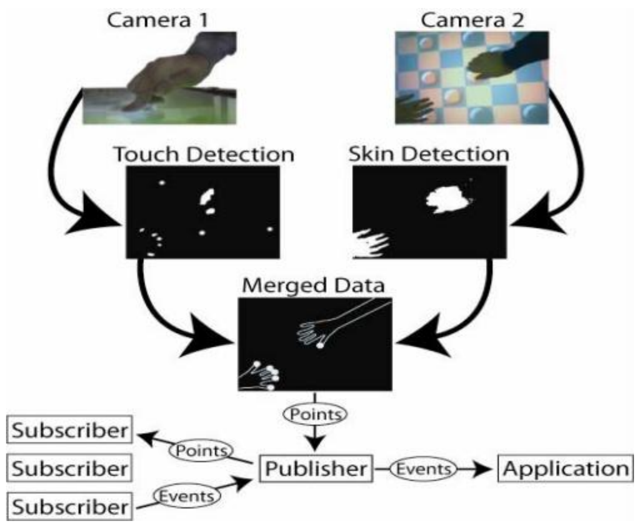

Figure 5: The workflow of the whole system [7]. The camera 1 is located inside the multi-touch table and the camera 2 is the additional webcam.

Tracking hands above the screen has also been proposed to expand direct interaction with interactive displays in depth. That means using the distance change between the multi-touch screen and the hand above or on the screen to achieve a continuous 3D interaction space [14, 1, 33, 10]. In principle, extending these approaches to associate touch points with the hands of multiple users is possible. However, the tracking range of the optical sensor systems mentioned above is limited to the area directly above the screen. Furthermore, the described systems (except [33]) do not even include high fidelity touch sensing.

Dang et al. [5] suggested a method based on the orientation of the tracked ellipse of the finger (see Figure 6) to identify the relation between the hands and the touch points. This method leads to false recognition when the thumb is touching the screen together with other fingers, since the thumb is more flexible than other fingers.

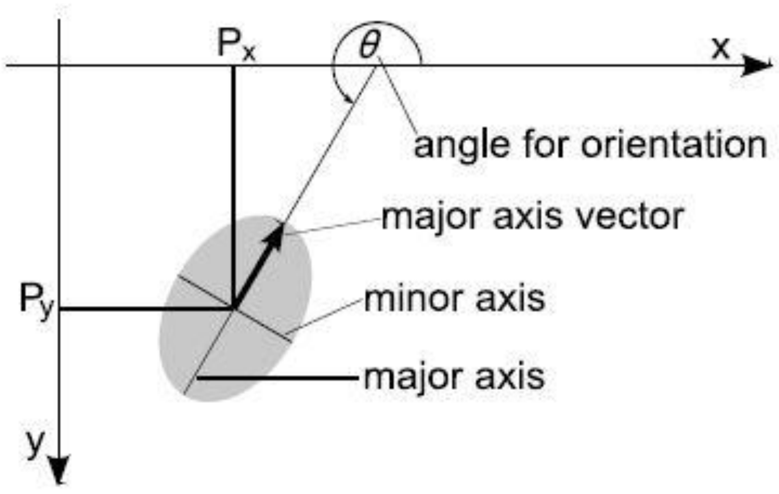

Figure 6: Finger blob represented as an ellipse with position and angle [5].

Another method presented by Andy Wilson [35] suggests using a depth sensing camera mounted on the ceiling as a sensor both for touch detection and context tracking (see Figure 7). Thus, the advantage of this method is that touch data can be directly associated with tracked user bodies. However, due to the relatively low resolution of the depth cameras, the touch points cannot be detected accurately.

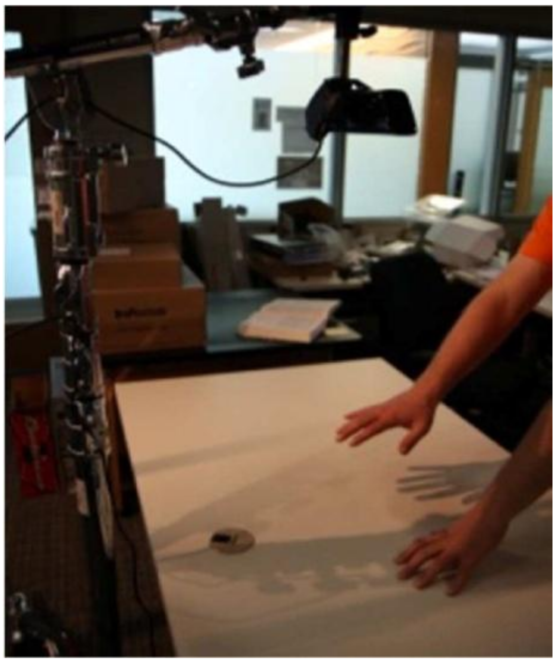

Figure 7: Experimental setup (camera height 0.75m above tabletop) [35].

Roth et al. [23] proposed a method that used an extra device for user tracking. In this method, a small infrared (IR)-emitting device is used for cryptographically user identification (see Figure 8). With an IR-ring on the finger of a user, the user can be detected when she/he is touching the screen. But, for this method, every user has to wear a ring. Furthermore, every user who wants to do operation on the screen has to touch the screen for authentication first.

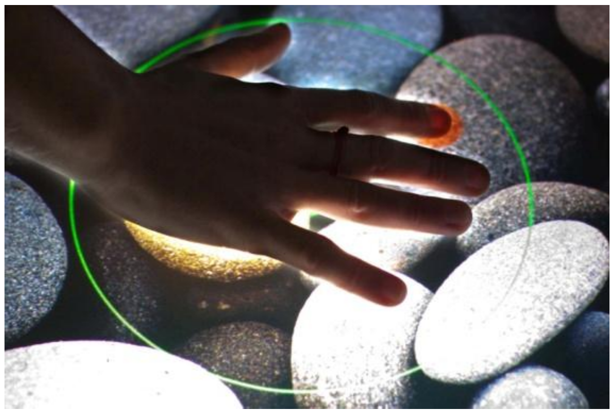

Figure 8: Shows the authenticated touch area underneath a user's hand. The IR Ring which identifies the user and authenticates the location is worn underneath the hand [23].

Another system has been designed by Walther-Franks et al [32] (see Figure 9). In this system, for user detection, infrared sensors which scan a limited area around the table are installed at the same height in the table. When a user enters into the area scanned by the infrared sensors, the user body position can be detected. However, the association between the detected touch points and a user's hand is not robust. Touch points detected in close proximity to the user's body position, tracked at the edge of the tabletop device, may also belong to somebody else reaching into his/her proximity.

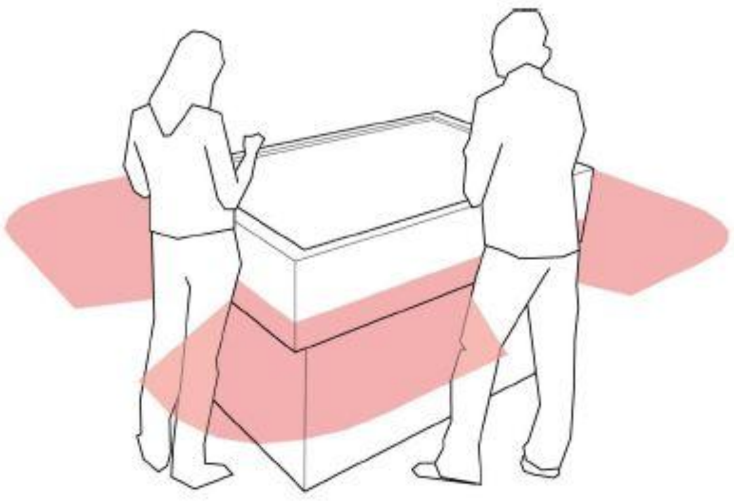
Figure 9: Proximity Position of Users [32].

3 Basic Work

In this chapter, the concepts used for user tracking in this thesis are described. After this, the hardware equipment including the multi-touch table and Microsoft Kinect is introduced.

3.1 Camera Calibration

In order to associate detected touch points with individual users, the camera parameters have to be computed. The camera parameters include the intrinsic and extrinsic parameters of the camera. The intrinsic parameters determine the optic, geometric and digital characteristics of the camera. These parameters can be described by the perspective projection, the transformation between image plane coordinates and pixel coordinates, and the geometric distortion introduced by the bending of the lens. For each camera, they are identified independently.

The extrinsic parameters describe the transformation between the camera reference frame and the world reference frame. They describe the position and orientation of the camera in the world coordinate system. In the user tracking application (See Chapter 4), the world coordinate system is decided as shown in Figure 10 (Note that axis z is not drawn here, axis z is unit normal on the display pointing up). Thus, these parameters cannot be done separately by each camera.

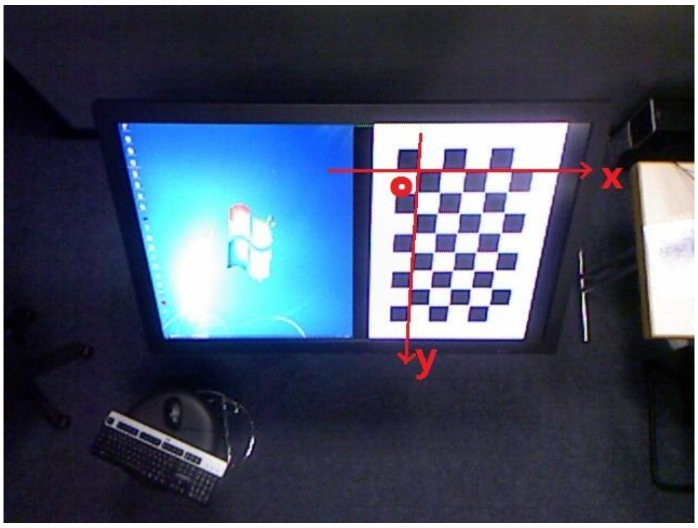

Figure 10: The world coordinate system in the user tracking application.

Camera calibration usually works as follows: First, an object with known geometry and color distribution (e.g. a chessboard) is shown to the cameras. This object provides visual feature points that can unambiguously be identified in the camera images. Then, by mapping the known feature positions with the computed feature positions in the image, the intrinsic camera parameters can be computed. Furthermore, by identifying corresponding feature positions for multiple cameras, the extrinsic camera parameters are also computed.

By using the intrinsic and the extrinsic parameters of the camera, detected touch point coordinates can be projected into the image. Note that this is a prerequisite for analyzing the relationship between image positions of touch points and regions of individual users in the image.

3.2 Image Segmentation

In order to detect users, first, an image segmentation method needs to be used in the camera images. The users need to be segmented out from the given images. Background subtraction is such an image segmentation method that works as follows: First, typically a background image is obtained by averaging several images before the foreground (e.g. walking users) is visible to the camera. Then, detecting the foreground is performed by thresholding on the difference between the current observed image and the background image. Thresholding means that a pixel in the current observed image is only marked as the foreground if the difference is bigger than a reasonable threshold. The quality of the segmentation depends on the threshold. If it is too low, lots of noise can be included. If it is too high, parts of the foreground can be missed.

3.3 Connected Components

After the image segmentation, the foreground image is more meaningful, since the background (e.g. display, carpet) has been removed. In this image, in order to distinguish among the users, the image regions belonging to the different user bodies in the image need to be distinguished. A concept of connected image components is used and each component is interpreted as a different user.

In computer vision, the definition of a connected component in the foreground image is a set of image pixels with maximal area where every two adjacent pixels belong to the foreground.

By using this concept, the binary foreground image can be interpreted as a set of different connected components. With applying a reasonable component size threshold on each connected component, the noise can be filtered out. Each of the remaining connected components in the image can be assumed to be a single user.

3.4 TUIO Protocol

In order to share touch information between applications in the context-aware multi-touch system, a communication protocol which can be described as a system of digital messages and rules which are used for exchanging those messages in or between systems needs to be well defined. The TUIO protocol [13] is a protocol which allows the transmission of an abstract description of interactive surfaces, including touch events and tangible object states. The protocol encodes control data from a tracker application (e.g. based on computer vision). The application sends the message to any client application that is capable of decoding the protocol (see Figure 11).

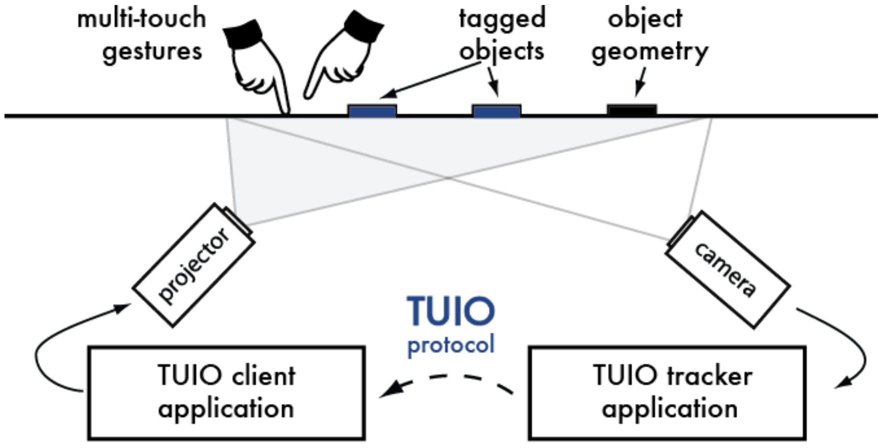

Figure 11: The workflow of the TUIO protocol [13].

Two main types of messages are defined in the TUIO protocol. One type is SET messages, which are used to provide information about an object's state (e.g. position) and other recognized states. The other type is ALIVE messages, which are used to indicate the current set of objects present on the surface using a list of unique Session IDs. In addition to SET and ALIVE messages, FSEQ messages are defined to uniquely indicate update steps with a unique frame sequence ID.

In the TUIO protocol, a set of profiles is defined to allow the transmission of cursor, object and blob descriptors within the context of two dimensional surfaces, in special cases, in the 3D space above the table surface as well. Besides, the TUIO protocol also allows the definition of free form custom profiles, which allow a user-defined set of parameters in similar format with predefined profiles. The profiles details can be seen below. The semantic types of set messages can be seen in Table 1. Please refer to [13].

- **2D Interactive Surface**
  /tuio/2Dobj set s i x y a X Y A m r
  /tuio/2Dcur set s x y X Y m
  /tuio/2Dblb set s x y a w h f X Y A m r

- **2.5D Interactive Surface**
  /tuio/25Dobj set s i x y z a X Y Z A m r
  /tuio/25Dcur set s x y z X Y Z m
  /tuio/25Dblb set s x y z a w h f X Y Z A m r

- **3D Interactive Surface**
  /tuio/3Dobj set s i x y z a b c X Y Z A B C m r
  /tuio/3Dcur set s x y z X Y Z m
  /tuio/3Dblb set s x y z a b c w h d v X Y Z A B C m r

- **custom profile**
  /tuio/_[formatString]

In the custom profile, a user-defined format can be used as the attributes of the set message. The defined format can be flexible. For example, a message like "/tuio/_point set x y" can be defined by a user for transmitting a 2D point coordinate via the TUIO protocol.

Table 1: semantic types of set messages

s	Session ID (temporary object ID)
i	Class ID (e.g. marker ID)
x,y,z	Position
a,b,c	Angle
w,h,d	Dimension
f,v	Area, Volume
X,Y,Z	Velocity vector (motion speed & direction)
A,B,C	Rotation velocity vector (rotation speed & direction)
m	Motion acceleration
r	Rotation acceleration
p	Free parameter

3.5 Hardware Equipment

In the context-aware multi-touch system, a multi-touch table for supporting multi-touch interaction and a Microsoft Kinect for supporting the user tracking are included. In this section, details on both hardware equipments are introduced.

3.5.1 Multi-touch Table

The multi-touch table that is used for this work was developed under the umbrella of the VRGeo Consortium (see Section 5.1). According to the feedback from the VRGeo Consortium members being representatives of the international oil and gas industry, the VRGeo Research & Development (R&D) team had several requirements related to the development of a new multi-touch table:

Form factor and resolution: The form factor of the display is large enough so that multi-users can work on the display simultaneously. Besides, in order to distinguish very small structures, the display needs to have ultra high resolution and image quality.

Robust real-time multi-touch: The display is able to support multi-touch technology. The response time of the multi-touch should be acceptable for the users.

Ergonomics: When users are using the multi-touch table, it is possible to reduce the risk of Repetitive strain Injury (RSI) [39]. Therefore, a drafting-table-like stand was designed that the users are able to change the height or inclination of the display.

Movable assembly: A construction that allows easy re-location and transport.

In order to fulfill the first and the two last requirements, a LCD display (LC-5621) (see Figure 12) produced by Barco (www.barco.com) was selected. The display is 56'' with ultra high resolution (3840x2160 pixels). The display has been specially designed for use in dedicated professional applications such as medical applications and it delivers crisp, clear and color-accurate images on a large display size.

Figure 12: Barco LC-5621.

In order to support robust and accurate multi-touch detection on this display, the dreaMTouch overlay manufactured by Citron (see Figure 13 Left) [15] is used. In this overlay, high numbers of IR emitters and IR receives are installed in all four sides of the display bezel. This way, touch points can be detected by computing the occlusion induced by the fingers or point devices infinitesimal above the screen, Figure 13 Right illustrates this work.

The overlay is able to accurately track up to 32 touch points simultaneously. The data throughput of 50 coordinates per second is fast enough for real-time collaboration. Besides, the overlay is strongly attached to the display and does not need any re-calibration after transport.

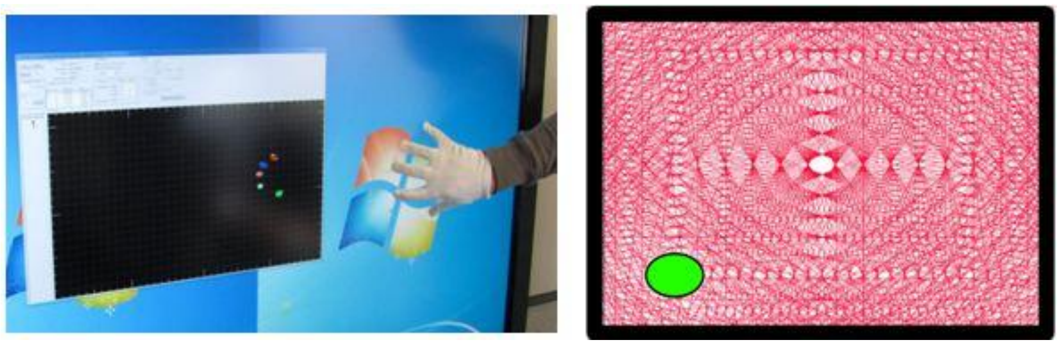

Figure 13: Left: DreamTouch Application. Detected touch points are visualized in the black frame of the application with different colors. Right: the red lines are the infrared lights emitted by the infrared emitters. The green point is a touch point.

In order to be an adaptable assembly, the VRGeo Research & Development team additionally developed a partly motorized stand (see Figure 14) which has the following characteristics:

Motorized adjustable height (see Figure 14 Top): The table can be moved into heights between 75 cm and 125 cm. Two height adjustment buttons are attached to the table for this purpose. With this characteristic, people can work on the table either sitting or standing.

Motorized inclination angle (see Figure 14): The table can also be leaned forwards and backwards between 0 and 70 degrees by pressing two inclination adjustment buttons. Thus, depending on tasks and the number of people working on the assembly, the multi-touch display can be used either uprightly or flat. This characteristic can also reduce the risk of Repetitive Strain Injury.

Manual pivot function (0-90 degrees) (see Figure 14 bottom): The display can be switched between landscape mode (0 degree) and portrait mode (90 degrees) by manually moving. When the display is in one mode, it can be fixed by a magnetic snap-in and a locking screw. Therefore, depending on the type of data to be worked with, users can choose a suitable mode for the display. Besides, the display is mounted centered, so manually movement is effortless.

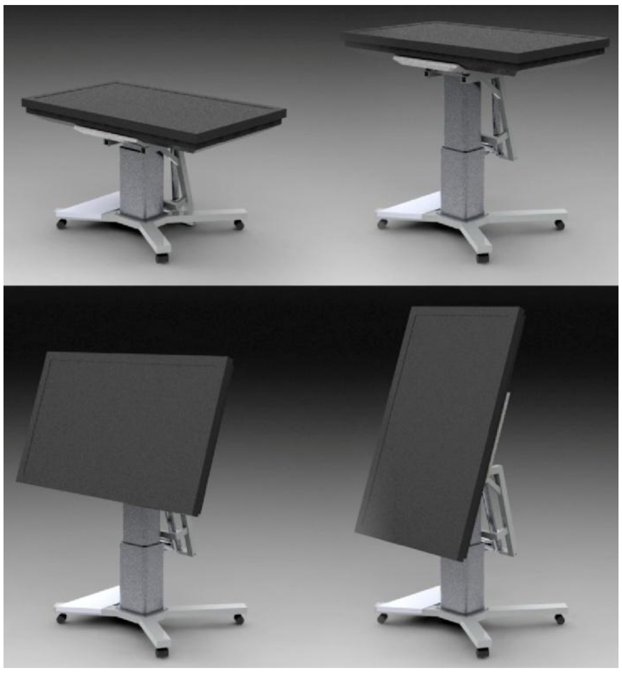
Figure 14: The stand in different positions. Top: Flat display at different heights. Bottom: Tilted display in portrait and landscape mode.

In addition, four rollers enable the whole assembly's quick in-house re-location. Besides, the assembly is also equipped with 3 axis accelerometer and micro controller at motors; thus, reporting a change of height or inclination angle of the multi-touch table is possible.

3.5.2 Microsoft Kinect

In the context-aware multi-touch system, besides the multi-touch table, the other important part is the device used for providing user tracking. For this purpose, according to the proposed approaches (See Chapter 4), several requirements have to be fulfilled for this device:

Segmentation: The device can provide depth information which can be used to solve the segmentation suitably. The segmentation method with depth information is able to overcome the problems of normal webcams.

Calibration: For automatic calibration, the device must be able to detect the pattern displayed on the screen.

Continuous tracking: The device must be capable of continuously providing image information for the user tracking application.

According to these requirements, the Microsoft Kinect [40] (see Figure 15) was selected as the device to provide the environment information.

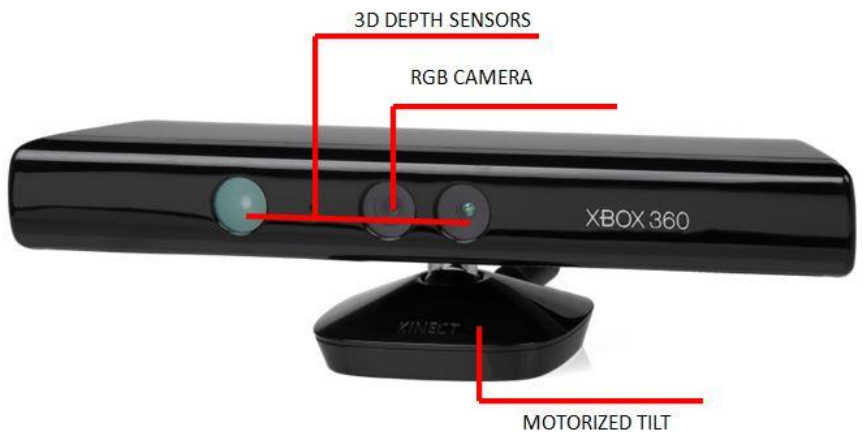

Figure 15: The Microsoft Kinect and its Structure.

The Microsoft Kinect was introduced in November, 2010. It is designed as a motion sensing input device for the Xbox 360 game console. As shown in Figure 15, the Kinect includes one laser-based infrared (IR) projector, one infrared sensor and one RGB sensor. For depth computing, both the projector and the infrared sensor are used. The depth calculations are based on triangulating features in the image. A fixed pattern with light and dark speckles is sent out by the IR projector. By using the pattern image captured by the infrared sensor and a known pattern memorized at a known depth, depth can be calculated. These sensors can provide video outputs at a frame rate of 30Hz. That means the image is updated every 1/30 seconds. This is fast enough for tracking walking users or user hand movements. The RGB sensor provides video stream with 8-bit per channel VGA resolution (640X480 pixels). The depth image is in VGA resolution (640X480 pixels) with 11-bits which can provide 2048 levels of sensitivity. However, the effective image resolution of the depth image is lower (approx. 320X240 pixels). The infrared sensor also can provide image information with a frame rate of 30 Hz resolution of 640X480 pixels.

Besides, a problem is observed that the chessboard pattern displayed on the screen for the camera calibration cannot be seen by the depth sensor or the infrared sensor of the Kinect. This is because the display does not emit infrared light, thus, here using the RGB sensor to detect the pattern for camera calibration is mandatory (see Figure 16).

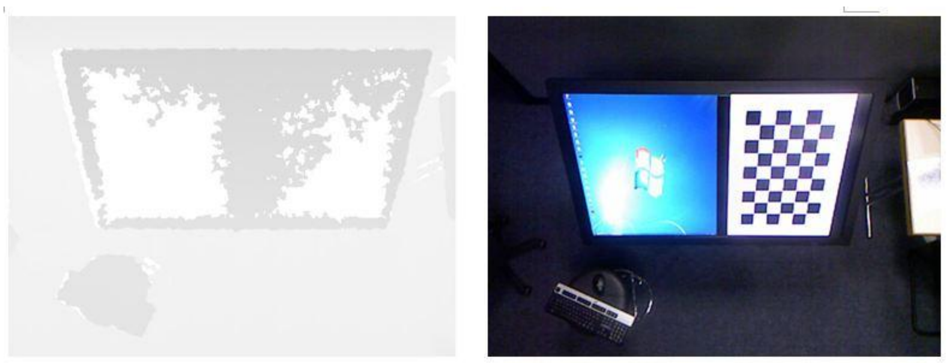

Figure 16: Right: the pattern can be viewed by the RGB sensor. Left: the pattern cannot be viewed by the depth sensor.

Furthermore, compared to other depth cameras in the market, the Microsoft is low-cost (only around 120€). Thus, according to the requirements mentioned before, the Microsoft Kinect which includes all sensors in one (depth, color, infrared) became the best choice.

4 Multi-touch Context Tracking

In this chapter, based on the existing methods and hardware described in Chapter 3, a new method for user tracking, user distinction and associating touch points with individual users is introduced.

Figure 17 illustrates the process of the user tracking application. Based on the image information (depth, infrared, color) from the Kinect, the application works as follows: First, if the camera calibration has not been done, it is automatically done by detecting the chessboard pattern displayed on the screen (see Section 4.2) based on the color information. After this, based on the depth and infrared information, a combined segmentation method is used to remove the background (e.g. floor, display) (see Section 4.3). Then, users are detected and tracked (see Section 4.4). At last, the touch points detected on the multi-touch table are associated with the previously detected users (see Section 4.5).

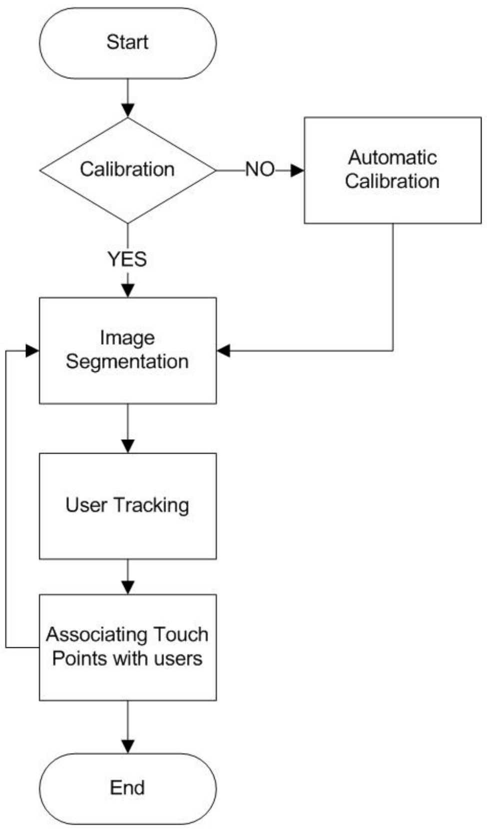

Figure 17: The process of the user tracking application.

4.1 Hardware Setup

As mentioned in Section 3.5, the system includes a high resolution multi-touch table and a Microsoft Kinect. As an additional sensor for providing additional information that enables reliable and robust user tracking, the Microsoft Kinect is mounted about 3 meters above the floor (see Figure 18). Thus, the Microsoft Kinect can capture the entire screen area and about 50 cm of its surrounding area in each direction.

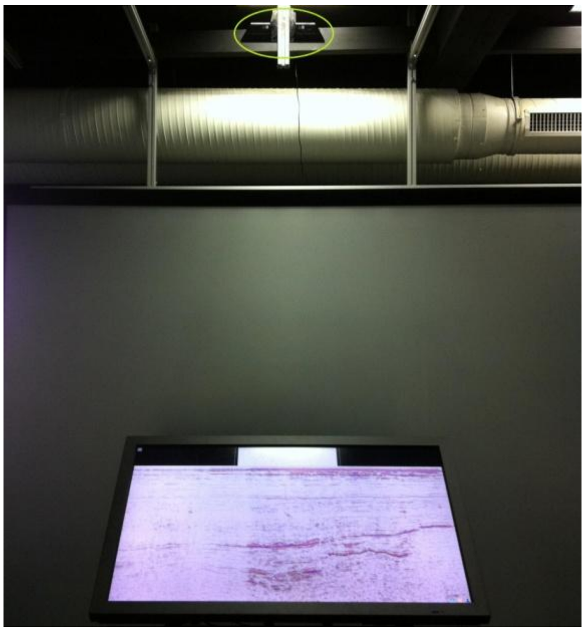

Figure 18: The arrangement of Multi-touch table (Bottom) with Microsoft Kinect (Top, in the green circle).

The whole setup can be seen in Figure 19. To drive the application on the multi-touch display, a standard desktop PC (PC A in Figure 19) is used. The Microsoft Kinect is connected to a second machine (PC B in Figure 19) via a USB cable. On this machine, the image information from the Microsoft Kinect is processed by the user tracking application. The two PCs are connected with a network switch. The touch information detected by the multi-touch table and the tracking information are shared between two PCs via network using the TUIO protocol (see Section 3.4).

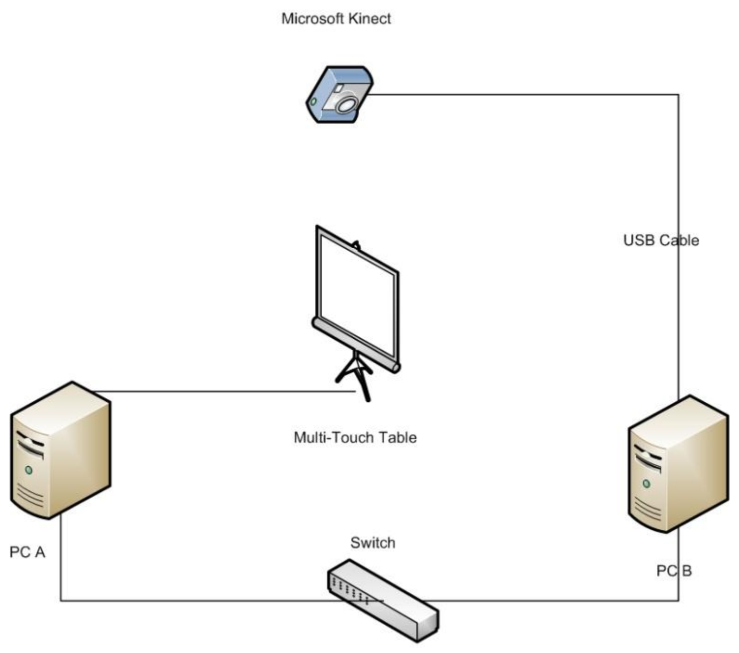

Figure 19: Hardware Setup.

4.2 Automatic Calibration

There are three different sensors (infrared, color, depth) in the Microsoft Kinect. For those sensors, first, the intrinsic parameters of them have to be computed, since those parameters do not change over time. Besides, the sensors are all embedded in the Kinect, so their extrinsic parameters with respect to each other are also needed to be computed only once. For computing those parameters, the method published by Nicolas Burrus [3] is used. In his method for computing the intrinsic parameters of the color sensor, a standard chessboard recognition method is applied. Then, he extracted four corners of the chessboard on the depth image for computing the intrinsic parameters of the depth sensor. At last, he also selected the four corners of the chessboard on the color image. By mapping corners' depth pixels with their color pixels, the relation between the color sensor and depth sensor is computed [3].

The extrinsic parameters defining the relation between the display area and each sensor of the Kinect device must be frequently recalibrated. These parameters must be recomputed every time when the display or the Kinect has been moved. Due to the adaptability of the assembly (see Section 3.5.1), this operation can happen very often. For this purpose, a fully automatic calibration method is proposed. The screen itself displays a calibration pattern (chessboard) instead of a printed chessboard pattern. This pattern can be recorded by the color sensor of the Kinect. Then, with the OpenCV method [2], for each chessboard corner, the image position in the coordinate system of the color sensor is identified. By mapping the 3D positions in world coordinate system of the corners with their corresponding image positions, the extrinsic parameters of the color sensor can be computed (see Figure 20) [2].

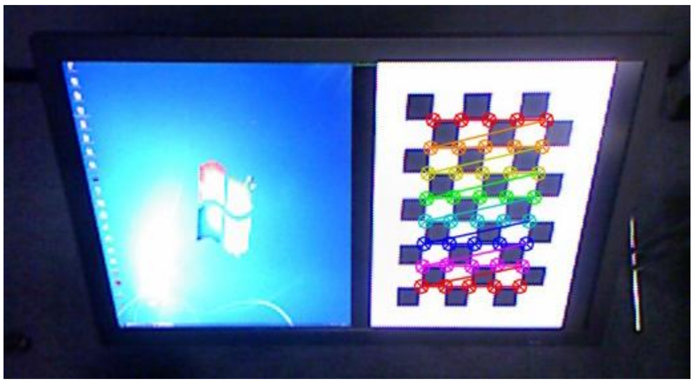
Figure 20: Calibration using OpenCV.

Note that as mentioned in Section 3.5.2, the chessboard pattern displayed on the screen cannot be seen by the depth sensor or infrared sensor of the Kinect. Thus, using the color sensor of the Kinect is mandatory for this procedure.

The relation between the depth sensor and the display (M3) can be computed by using the transformation between the color sensor and the display area and the previously calibrated internal relation between the color sensor and the depth sensor. In Figure 21, the relation between sensors and display are described. With Nicolas Burrus' method, the internal relation (M2) between the color sensor and the depth sensor can be calibrated. M1 can be calibrated by using the color sensor and the calibration pattern on the screen. Thus,

$$P_c = M1 \times P_s \quad (1)$$
$$P_d = M2 \times P_c \quad (2)$$
$$P_d = M3 \times P_S \quad (3)$$

Where Ps is a point in the world coordinate system, Pc is the corresponding point to Ps in the coordinate system of the color sensor, and Pd is the corresponding point to Ps in the coordinate system of the depth sensor.

Together with the equations (1), (2) and (3), the relation between the depth sensor and the display (M3) can be represented as:

$$M3 = M2 \times M1 \quad (4)$$

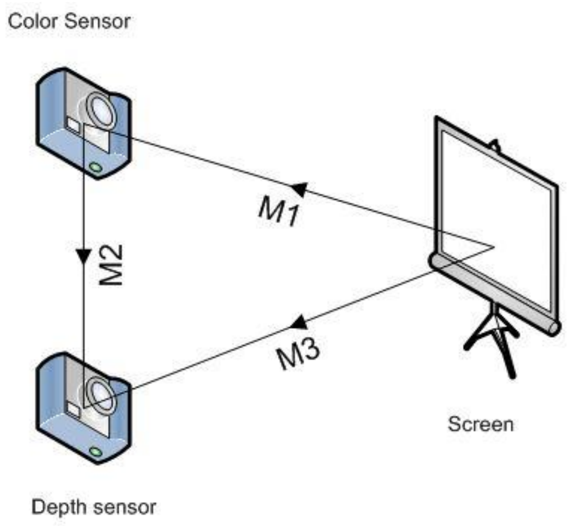

Figure 21: Camera-Extrinsic Parameters.

Besides, if desired, the calibration can be triggered automatically by using the sensors (see Section 3.5.1) attached to the multi-touch table. Those sensors are able to report a change of height or inclination angle.

By manually measuring the distance of each display corner to the displayed chessboard pattern beforehand, the 3D position of each corner can be computed from the calibrated origin and orientation (see Figure 22). This information is used in combined segmentation (see Section 4.3) for display region calculation and touch points association (see Section 4.5) for computing the touch position in world coordinate system.

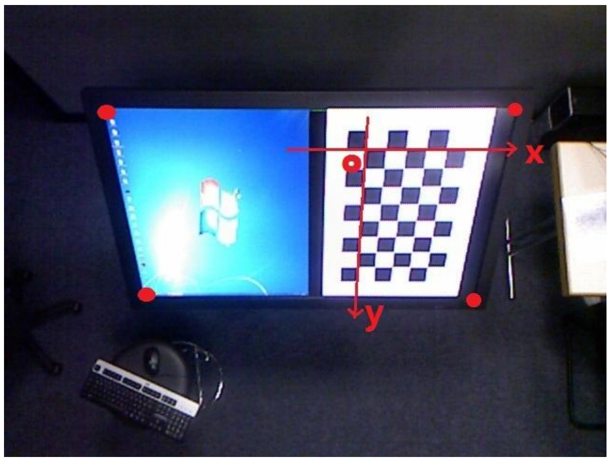

Figure 22: The world coordinate system in the user tracking application and the corners of the display. Four red points are four corners of the display.

4.3 Combined Segmentation

The Microsoft Kinect includes three sensors. By taking advantage of the information acquired from each sensor, different ways to do the image segmentation are possible. However, for each method, there are drawbacks.

First way is using the color information, but color segmentation or background subtraction severely restricts the colors appearing in the background. As this includes the display content itself, such disadvantage cannot be overcome. Furthermore, canceling the light from the displays by means of polarization filters as described in [7] is not possible in such a highly adaptive setup.

Second way is using the depth information. Comparing with using the color information, it has several advantages. Only objects occluding the view of the Kinect must be avoided. However, for robust operation, some problems still exist. The imprecision and quantization of depth values obtained by the Kinect impede precise depth segmentation when body parts (e.g. fingers or hands) are close to or on the display surface (See Figure 23).

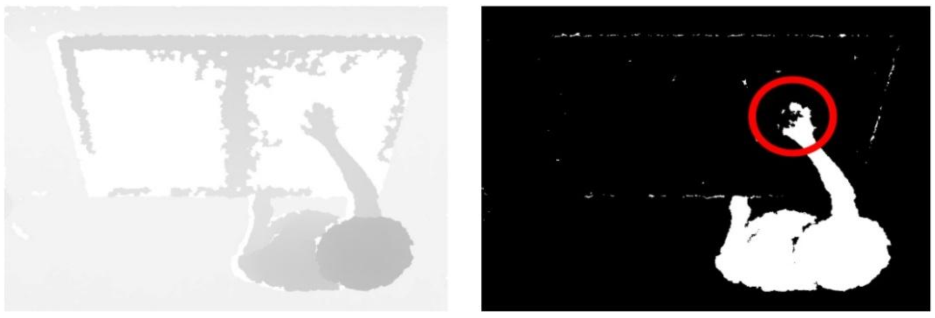

Figure 23: Depth based segmentation leads to problems near the display surface. Left: depth image. Right: depth based segmentation with missing foreground inside the red circle.

A third way of segmentation is using a background subtraction algorithm on the infrared (IR) intensity information from the Kinect. Because the display does not emit infrared light, this also works when user body parts are close to or touching the display surface. However, it also has its own drawbacks. Using background subtraction again restricts the background around the display. E.g. a carpet lying on the floor or a bag added by a user can affect the segmentation result (See Figure 24).

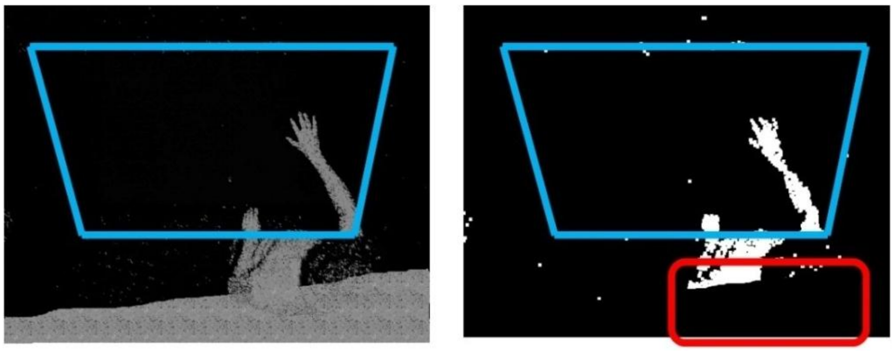

Figure 24: Infrared based segmentation can induce problems around the display. The blue rectangle indicates the display region. Left: IR image. Right: IR based segmentation with missing foreground in the red rectangle.

Therefore, finally, a combination of the infrared segmentation result and the depth segmentation result inside the image region corresponding to the display surface is proposed. During the camera calibration step, four corners of the display have been computed. Thus, the image region corresponding to the display surface can be computed easily. Using a logical or-operation, the depth-based segmentation of the entire image is combined with the infrared segmentation, but only inside the image region corresponding to the display surface.

In the implementation, before interaction can take place, 20 depth and infrared images are averaged as the depth and infrared background images, respectively. In every frame, a background subtraction algorithm is applied to the depth image and infrared image respectively. Afterwards, to obtain binary images, for the depth subtraction image, a threshold (pixel value difference) 2 is selected, while 10 is selected for the infrared subtraction image. At last, based on the method discussed in the previous paragraph, the two images are combined together as the final segmentation image. Figure 25 illustrates the way of combining the segmentation images and the resulting segmentation performance.

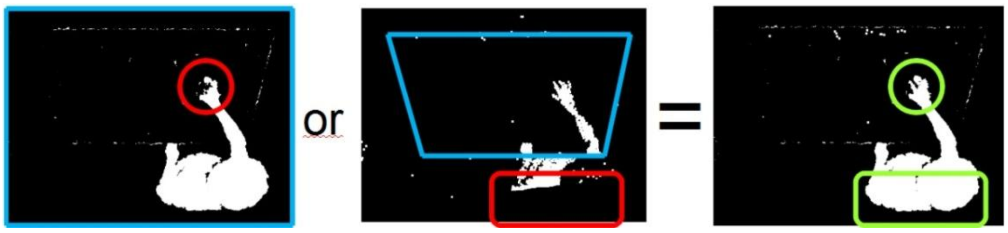

Figure 25: A combination of depth segmentation (left image) and IR intensity segmentation (middle image) leads to better segmentation results (right image) and solves the respective problems (see regions in green circles).

Besides, a problem is observed that some users may take bags or something with them when they come to use the application. These objects do not belong to the beforehand learned background images (both infrared and depth), so when doing the image segmentation, these objects are seen as foreground. This may cause problems. For example, upper parts of user bodies are separated, but if their legs are all touching the bag, then, two users are seen as one user. Normally those objects are lying on the floor. Therefore, they are usually located below the display. Exploiting this, a method for segmenting out those objects by using the depth information of the Kinect is implemented. The distance between each corner of the display and the Kinect is calculated. The corner with the largest distance is selected. Afterwards, with the given distance, by using the equation (5) published in [20], the corresponding depth value of the corner in Kinect can be calculated.

$$\mathrm{z} = \frac{7.5 * 580}{\frac{1}{8} * (1090 - \mathrm{d})}\ \mathrm{cm}(5)[20]$$

Where z is the distance factor, d is the depth value. 7.5 is the horizontal baseline between infrared sensor and infrared projector (in cm). 1/8 is the sub-pixel accuracy. 580 is the focal length of the infrared sensor (in pixels). 1090 is an offset value for Kinect device. Please refer to [20].

In the depth background image, for all pixels whose values are bigger than this given value, their values are changed to this given value (see Figure26). That means objects below the display will not be segmented as foreground.

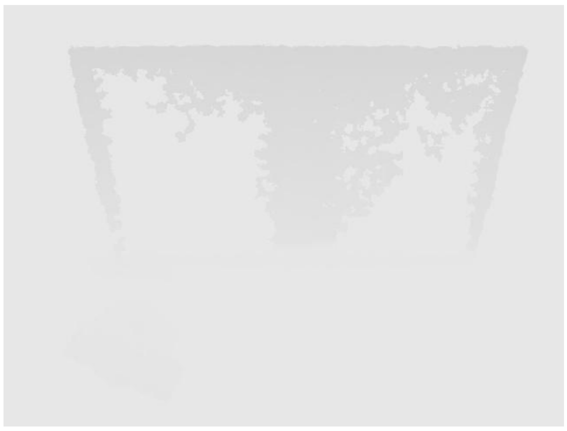

Figure 26: The new depth background image.

4.4 User Tracking

After the segmentation, only noise and regions belonging to the users remain in the image. In this image, a method used for searching for the largest connected components is used. In OpenCV, this can be done by finding contours which can be described as sequences of points defining a line/curve in an image [2] (see Figure 27). Each contour can be represented as one connected component.

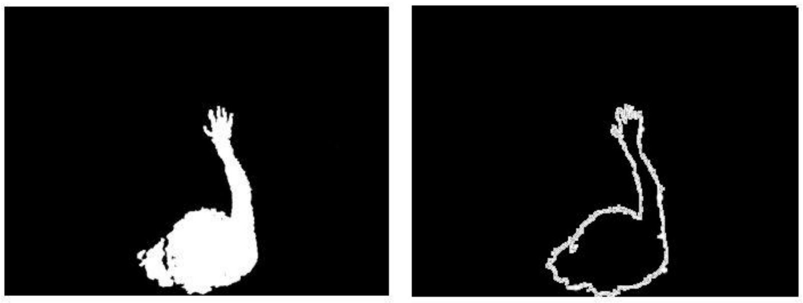
Figure 27: Left: segmentation result. Right: corresponding contour.

Afterwards, a simple threshold on the component size is used to filter out small components that typically correspond to image noise. The remaining connected components (CC) can be assumed to correspond to a single user. Each of them is assigned a distinct user ID (see Figure 28).

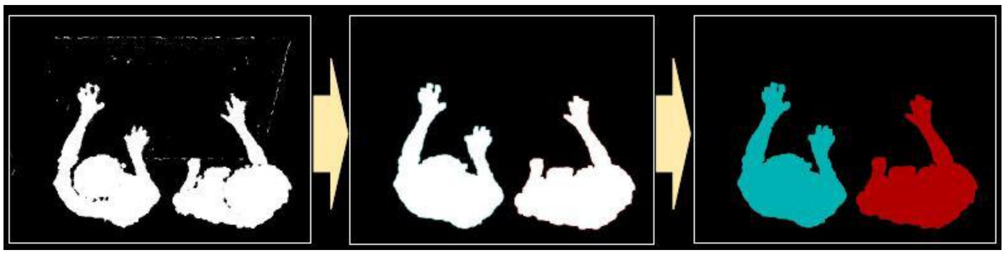
Figure 28: Left: a combined segmentation. Middle: Largest Connected components. Right: Largest connected components colored according to user ID.

For user ID consistence, in every frame, the center of each component is calculated. This is enabled by computing the bounding box for the component in OpenCV [2]. For each component, its center is compared with all centers of components in the previous frame and the user ID of the component which is closest is assigned to this component. For the component which has no nearest component, it is interpreted as a new user and a new user ID is assigned to this component.

4.5 Associating Touch Points to Users and Hands

Finally, for each detected touch point the corresponding user and hand have to be estimated. As mentioned in Section 4.1, touch information is transferred from the PC which is connected to the multi-touch table using the TUIO protocol. The message format used in the application is "/tuio/2Dblb" (see Section 3.4). In every SET message, one touch point is described (see below).

/tuio/2Dblb set s x y a w h f X Y A m r

The semantic types in this message can be seen in Table 1 in Section 3.4. Here x and y are not the screen pixel position of the touch point. Because the TUIO coordinate system is normalized for each axis, x and y are normalized positions and range from 0.0 to 1.0. The screen pixel positions can be calculated with the following equations:

$$\mathrm{X} = \mathrm{x} * \mathrm{w} \quad (6)$$
$$\mathrm{Y} = \mathrm{y} * \mathrm{h} \quad (7)$$

Where x, y are normalized coordinates, w and h are width and height of the touch frame respectively. The four parameters are all included in the message.

The x-axis, y-axis of the TUIO coordinate system is in the same plane with the x-axis, y- axis of the world coordinate system. The orientations of two coordinate systems are the same. However, the origins of both coordinate systems are not at the same position (see Figure 29). Therefore, by applying a translation on the screen pixel position of the touch point, the 2D touch point position can be translated into world coordinate system. By measuring its coordinate in z-axis (here coordinate in z-axis is 0), the 3D position of the touch point in world coordinate system is calculated.

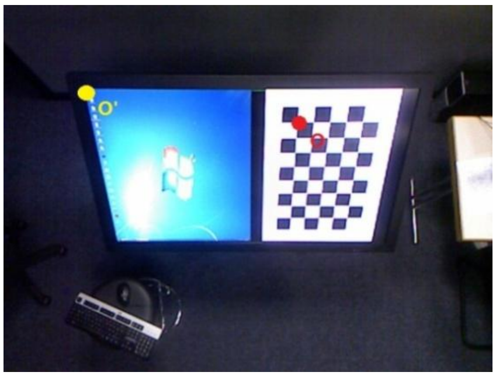

Figure 29: The origin of the TUIO coordinate system (Yellow Point) and the origin of the world coordinate system (Red Point).

With the help of the transformation matrix derived from the camera calibration, the transformed 3D touch points are projected to the image containing the user regions (see Figure 30).

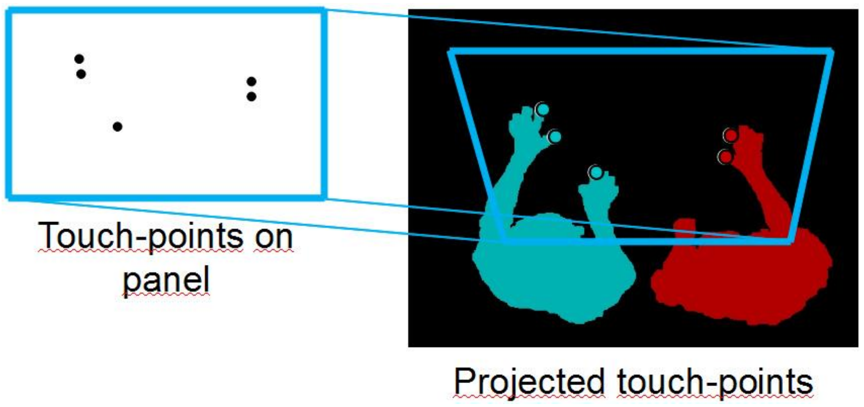

Figure 30: Projection and assignment of the touch points (black dots on the left) to the image containing the user regions (right). The blue rectangle indicates the area of the display surface. The projected touch points (colored dots with black margin on the right) are assigned to the closest user region.

As mentioned in Section 4.4, contours are sequences of points. An OpenCV point polygon test method [2] is that interpreting the contour as an approximate polygon, then testing whether a given point is inside the polygon or not. For every touch point, this method is used to decide its user ID. During the process, if a touch point is located inside a user region or on the edge of a user region, the corresponding user ID is transcribed directly. Otherwise the user ID of the closest user region is assigned to the touch point.

4.6 Implementation Details

Based on the method introduced in the previous paragraphs, the user tracking application is implemented.

4.6.1 Code Architecture

The whole application is implemented in C++. For the application, an open source library OpenKinect [37] is used. This is a library used for acquiring image information (depth, color, infrared) from the Kinect. Additionally, the OpenCV [2] library is used for supporting camera calibration and image processing.

In the application, four major classes are implemented:

Class cCalibration: This class is implemented for supporting automatic calibration (see Section 4.2). In this class, the extrinsic parameters of the depth sensor and the corners of the display are computed. A method used for projecting 3D points into image position is also implemented. This method is used in Class cUDPListener for projecting the 3D touch point into image position.

Class cBGSegm: The combined segmentation method (see Section 4.3) is implemented in this class. A method used for accumulating and averaging the background image, a depth segmentation method, an infrared segmentation method and a method used for combining the depth segmentation result and infrared segmentation result are implemented in this class.

Class cFindContours: This class is implemented for user detecting and tracking (see Section 4.4). In this class, a method for finding largest connected components based on OpenCV contour searching [2] is implemented for searching the connected components in the image. Additionally, a user ID consistence method is implemented for keeping the user ID consistence.

Class cUDPListener: This class is used to get the touch point information from the dreaMTouch application. In the class, a point test method (see Section 4.5) used for associating the touch points with individual users is implemented. It also has a method used for sending the touch point and its user ID to the VRGeo Seismic Touch application (see Section 5.4.1).

4.6.2 User Interface

When the user tracking application starts, the color image and depth image are visualized with the resolution of 640x480 pixels respectively (see Figure 31). The application starts to automatically detect the chessboard pattern for camera calibration. As soon as the calibration succeeded, key “f” can be pressed to change the visualized images to be the infrared image and depth image (see Figure 32) instead. Furthermore, then the application starts performing the image segmentation and association of touch points with users. Meanwhile, the colored user regions image and the touch points are also visualized automatically (see Figure 33).

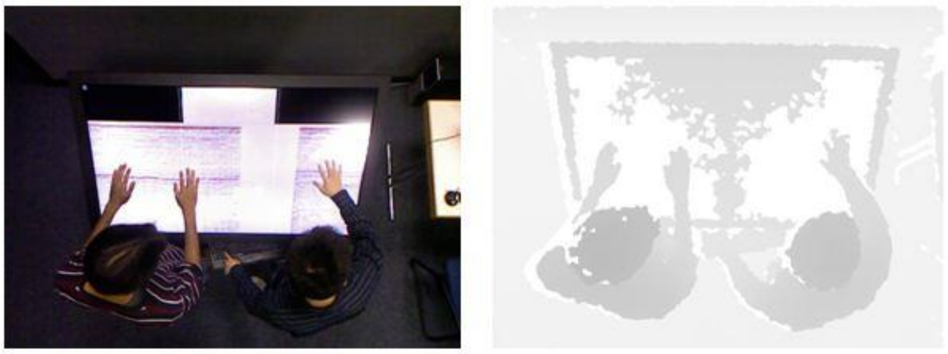
Figure 31: Left: the color image. Right: the depth image.

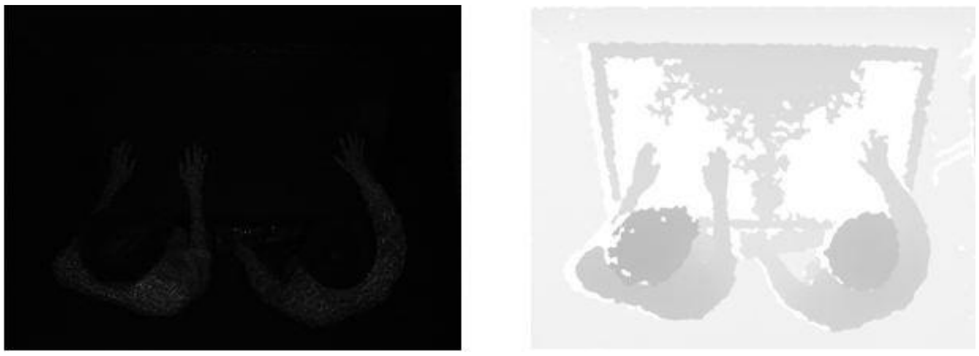
Figure 32: Left: the infrared image. Right: the depth image.

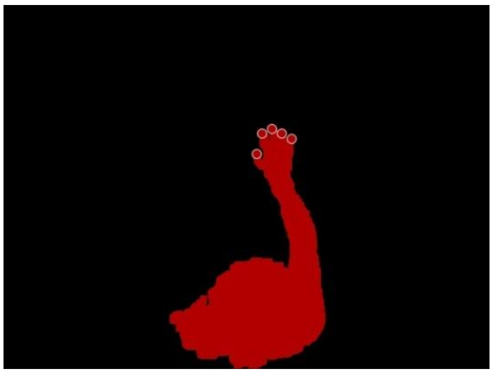
Figure 33: Touch points and colored users.

5 System Integration

5.1 The VRGeo Project

Seismic data are used as the primary source of information by oil and gas companies to locate oil and gas deposits. For getting seismic data, normally an energy source is used. It is able to send sound waves into the subsurface strata. When those waves are occluded with underground formations, they are reflected back to the surface. On the surface, a microphone type of device which can digitize and record the reflected waves named geophone is used to detect the reflected waves. Figure 34 illustrates this work. Software is then used to process the raw data to develop an image of underground formations. The resulting seismic data is then analyzed by a team of specialists using suitable interactive visualization techniques.

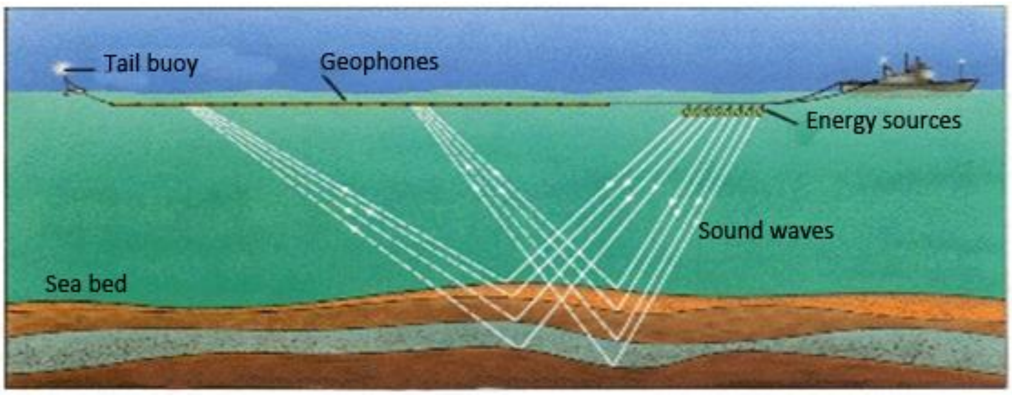

Figure 34: The work for acquiring the Seismic Data.

The VRGeo [4] project is a research project to develop application prototypes to explore seismic data. The applications are developed continually at Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS) for the VRGeo consortium. The consortium was established in 1998 by Adolfo Henriques from Statoil (Norway). Members of the consortium are from the oil and gas industry and their providers of software and related hardware. The mission of this consortium is to develop new methods and technologies, which can be eventually utilized for their field of work, i.e. oil and gas exploration. The main focus of VRGeo is on visual analytical systems for the oil and gas industry. The Research & Development topics include real time advanced visualization (e.g. volume rendering) research, natural interfaces (e.g. multi-touch) development, and team working environment (e.g. multi-user display system) development. Some demonstrators can be seen in Figure 35.

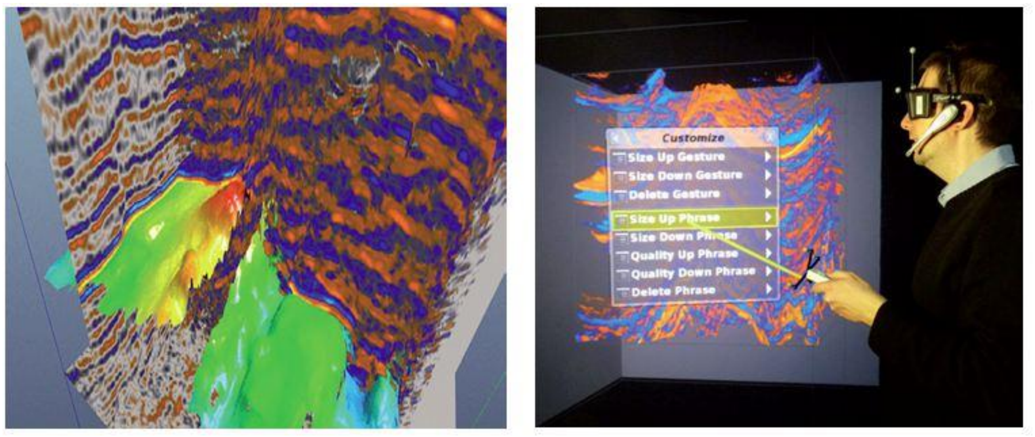

Figure 35: Left: Seismic data are visualized in real-time and 3D. Right: Analysts interact and collaborate to explore huge oil and gas data sets [4].

At the two annual meetings of the consortium, new applications and new features in different topics are presented to the representatives of the VRGeo Consortium members. During the sessions, the participants (experts in the fields of Virtual Reality, geology and geophysics) can test different applications themselves and discuss their thoughts and opinions. Afterwards the representatives give their feedback on the applications to the VRGeo Research & Development team. This is the basis of the research agenda for the next meeting. This also determines which technology will be kept as part of the application and which will be abandoned.

5.2 The VRGeo Seistouch Application

Based on the high resolution multi-touch table, a new VRGeo seismic multi-touch application was introduced at the VRGeo Consortium December 2010 meeting. It is an application prototype based on the multi-touch table for the interpretation of seismic data as depicted in Figure 36.

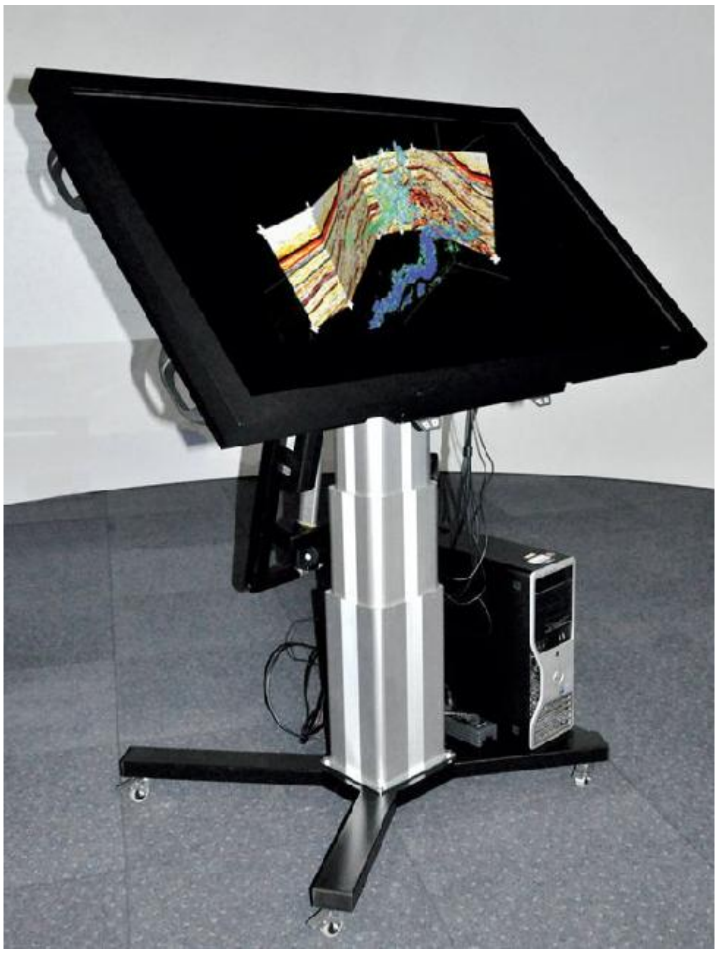
Figure 36: the multi-touch based prototype for the interpretation of seismic data [4].

A volumetric dataset is a set of volume which is a 3D array of voxels. The VRGeo seismic multi-touch application displays cut planes derived from a volumetric dataset on the screen for the collaborative interpretation of seismic features (e.g. fault). A horizontal cut plane is the horizontal 2D projection of the 3D volume data. Poly-plane is a polygon mesh extracted from a volume. The application primarily supports navigation along a horizontal cut plane used as a basemap (see Figure 37 Left), and selection and annotation of orthographic cut poly-planes, the so-called seismic lines (see Figure 37 Right).

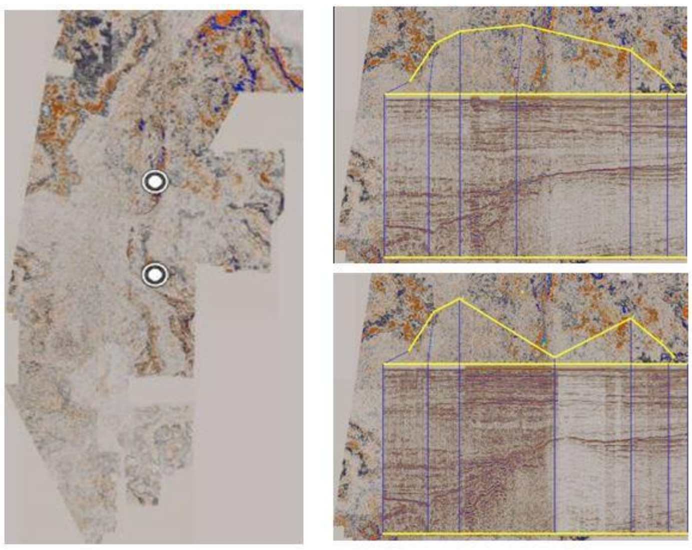

Figure 37: Left: the basemap. Right: seismic lines.

As introduced in Section 3.5.1, detecting touch information (e.g. position, finger) on the multi-touch table is enabled by the dreaMTouch technology. For supporting multi-touch interaction, the seismic multi-touch application receives messages which include the touch information from the dreaMTouch application.

The seismic multi-touch application includes both multi-touch interaction and command interface. In order to take advantage of the multi-touch technology, in the application, lots of multi-touch gestures are implemented. For example, for one finger interaction, dragging and drawing operations are implemented. For two fingers interaction, a zooming operation is implemented. With the multi-touch technology, users can use their fingers for dragging, zooming, seismic poly-line extraction and modification on the basemap. As well dragging, zooming, using typical annotation tools (e.g. Point annotations, Line annotations) (see Figure 38) on the seismic lines are possible, respectively.

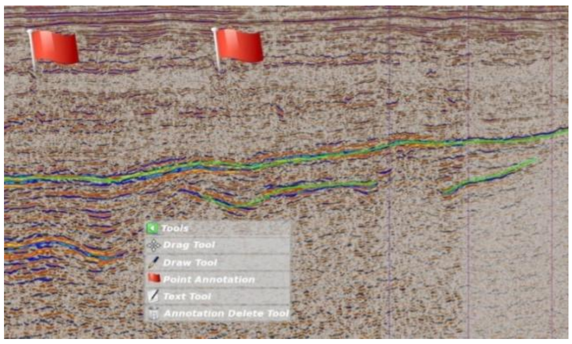

Figure 38: Multi-touch interaction on seismic lines. The green line is one type of line annotations. The red flag is one type of point annotations.

In the application, the command interface (the menu) is activated by two-finger double tap on the multi-touch screen. The menu on the basemap includes options on tools switching, seismic poly-line modification and extraction (see Figure 39). On the seismic lines, there are options on annotation tools switching and the operation on seismic lines (e.g. Fullscreen Seismic lines) (see Figure 40).

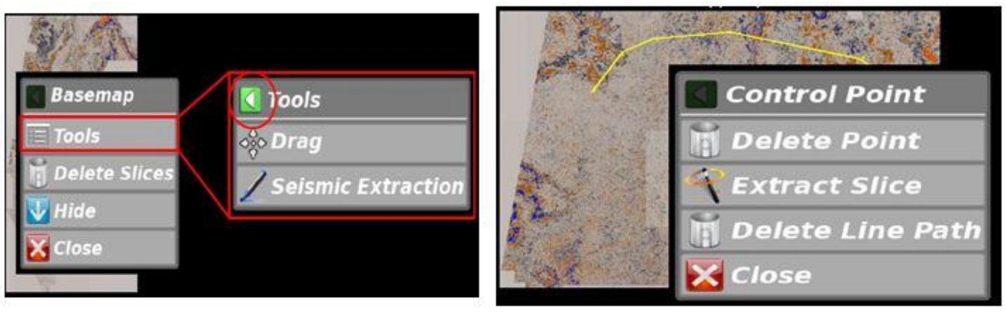

Figure 39: Menus on the basemap. Left: Tools switching. Right: seismic poly-line modification and extraction.

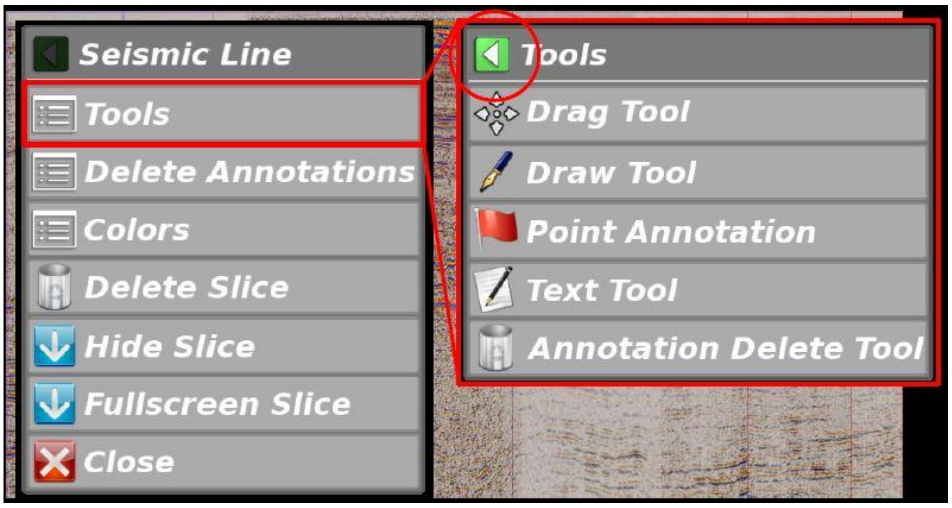

Figure 40: Menu on the seismic line.

5.3 Possible Enhancements

With the user information provided by the user tracking application, lots of new possibilities can be implemented and integrated into the context-aware tabletop system. In this section, several possibilities are described.

5.3.1 Involuntary Touch

In the multi-touch system, only the user hands can be seen as valid input parts. Other input parts from a user must be ignored. Involuntary touch is touch input not from the user finger. This problem can be caused by either the cuff of the cloth or the arm of the user. This is a common problem during the multi-touch interaction. Representatives of the VRGeo Consortium usually wear suits when they come to attend the meeting and test the multi-touch application. When they are interacting, their cuffs of suits can easily touch the screen together with their fingers. Similarly sometimes users lean on the display, in this situation, e.g. the user’s elbow is touching the screen. This must be ignored as well. This involuntary touch can cause a problem as follows: when a user wants to draw a line, but his/her cuff is also touching the screen, then the drawing operation becomes the scaling operation.

5.3.2 Orientation of GUI-Elements

In the tabletop system, when users are doing interaction, they usually surround the display from all accessible sides. In this situation, an orientation problem of the GUI-Elements may occur. That means the GUI-Element is not oriented to the user who wants to use it. For this problem, Shen et al. [28] described a metaphoric “magnet” feature allowing the reorientation of all GUI-elements to improve legibility (see Figure 41).

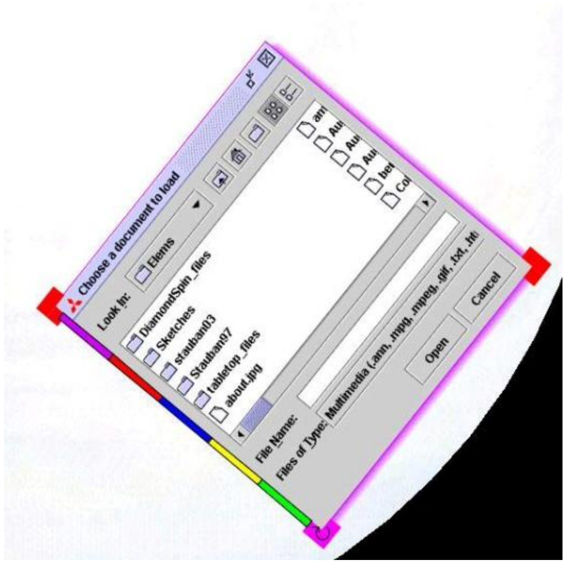

Figure 41: A menu with arbitrary orientation [28].

Besides, in the user tracking application, the user movements can also be tracked. Thus, if a user moves around the display, it is possible to move the GUI-Elements together with the user.

5.3.3 Individual Tool Selection

When users are doing line or point annotating on the application, they expect to keep a selected tool until they select a new one. However, in the current system, tools are associated with the user who selected it. That means when a user is using an annotation tool, but another user selects a dragging tool on the menu, then the tool of the previous user is changed to be a dragging tool. This leads to interference when multi-users are working on the application together. A functionality used to associating particular tools to the fingers of individual hands can be implemented for this purpose. Therefore, each tool is associated with individual user.

5.3.4 Locking Mechanism

In the multi-touch system, users often want to stop others from moving an element when they are doing an operation. To this end they touch the moving element in an attempt to stop it. Without any context-awareness such conflicting input from multiple users generally results in scaling the respective graphics element. For solving this problem, a locking mechanism can be implemented. This mechanism can be designed like that it only locks the type of possible manipulations to those already operated by the user who first acquired the element. Thus, if one is moving an item, others may still interfere to hold it, but this interference will not cause any other type of transformation, e.g. scaling.

In combination with the individual tool selection this approach can also be improved to allow multiple users to apply different operations simultaneously on the same element. Similar to real world experiences, one may for example continue to draw lines on an object while it is moved around by somebody else.

5.3.5 User-Dependent Annotation

In the current system, the annotations (e.g. a line) from different users cannot be distinguished. This may lead to interference between users. One user may delete another user's annotations during interaction unwillingly. To solve this problem, with the user information for the detected touch point, functionality for user-dependent annotation can be implemented. That means each annotation is associated with an individual user. The annotations from different users can be distinguished.

5.3.6 Occlusion Awareness

For multi-user interaction systems, when there are several users working on the display simultaneously, it easily occurs that some of the display region will be occluded by the users. For this reason, an occlusion detection method can also be implemented. For example, when one user's GUI-Element is occluded by another user's hand, the application can automatically visualize the GUI-Element somewhere else on the display.

5.4 The Context-aware VRGeo Seistouch Application

According to the feedback on the VRGeo seismic multi-touch application at the VRGeo consortium December 2010 meeting, from the mentioned functionalities in Section 5.3, the individual tool selection, locking mechanism and user-dependent annotation are implemented for the VRGeo seismic multi-touch application. With implementing these functionalities, the experts from the VRGeo consortium can work on the tabletop system collaboratively, but in different contexts. Note that implementing all functionalities was not in the scope of this master thesis.

5.4.1 Application Communication

The system consists of three applications, the user tracking application used for context tracking, the seismic multi-touch application used for interpreting the seismic data and the dreaMTouch application used for detecting touch points. These applications send respective information via the TUIO protocol as illustrated in Figure 42.

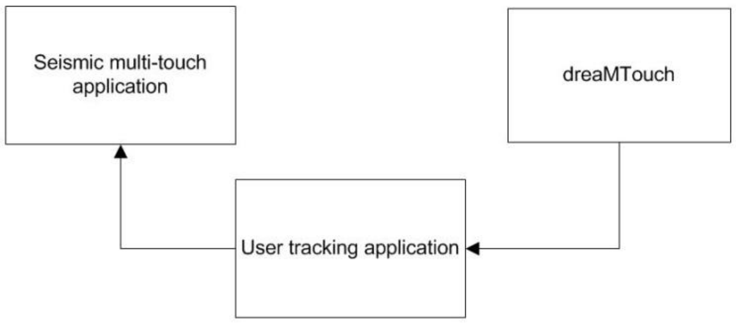

Figure 42: Communication between the applications.

In every frame, the dreaMTouch software acquires and processes touch point information from the hardware. Afterwards, the touch point information (e.g. finger ID, finger position) is sent to clients who are connected to the dreaM-Touch application. In the system, the user tracking application is connected to the dreaMTouch for acquiring touch point information. Meanwhile, once it gets the information, it sends the information to the seismic multi-touch application. The seismic multi-touch application uses the touch point information to support multi-touch interaction. In the user tracking application, the touch points are associated with individual users. Therefore, every touch point has assigned a user ID. Then the user tracking application send the modified touch point information which has already contained the user IDs to the seismic multi-touch application.

The dreaMTouch application uses the defined TUIO message format ("/tuio/2Dblb", see Section 3.4). However, from the user tracking application to the seismic multi-touch application, a new user ID field needs to be transmitted, so an existing TUIO message format cannot be used. As introduced in Section 3.4, the TUIO protocol also supports custom message format. Therefore, a new type of message format named "/tuio/_kinect" is defined for this purpose. The message format can be seen below.

- **TUIO Kinect Message**
  /tuio/_kinect ALIVE User1, User2…
  /tuio/_kinect SET User1 Finger11, Finger12…
  /tuio/_kinect SET User2 Finger21, Finger22…
  FSEQ

In the new defined message format (TUIO Kinect Message), the ALIVE message includes the user IDs which are in the scene (the area which can be captured by the Microsoft Kinect). For every user ID in the ALIVE Message, one additional SET message is sent. In every specific SET message, it starts with the respective user ID. After the user ID, it is followed by the fingers which belong to this user ID. At last, the message is ended by a FSEQ message like a normal TUIO message.

5.4.2 Integrated System

After the communication rules between the different applications are defined, the seismic multi-touch application can be modified to exploit the relationship between fingers. For example, two fingers with different user IDs are translated to two separate dragging gestures from two users, but not a scaling gesture.

The next work is implementing the interaction logic in the seismic multi-touch application so that multi-users can work on this application collaboratively but in different contexts. As mentioned in the previous paragraph, the individual tool selection, locking mechanism and user-dependent annotations are implemented for the seismic multi-touch application. Compared with the previous version of the application, the new seismic multi-touch application has the following new features.

In the new system, a particular tool is associated to an individual user. This is the most important augmentation for multi-user multi-touch interaction. This is implemented by associating a menu to a specific user. Every time, when a user opens a menu, the application associates the menu to the specific user by taking advantage of the user ID information from the finger. That means every menu also has a specific user ID. If one user touches another user’s menu, it is invalid. For example, when one user opens one menu, another user with a different user ID selects one tool in the menu, the tools of both users are not changed. Only when a finger has the same user ID with the menu, then the tool selection operation can be seen as valid. This way, the tool can associate with an individual user. Thus, different users are able to have different tools simultaneously. For example, one user can have a point annotation tool and another can have a dragging tool simultaneously. Furthermore, due to the association between the menu and the user, when a user leaves the tracking area totally, the user’s menu can also be automatically closed.

With user-dependent tools, several users can do operations together on the same element. However, this may lead to interference between users in certain situations, especially when one user uses the dragging tool. For example, when one user is doing point annotation on the element, but another user tries to drag it. Those conflicting inputs from multiple users must be forbidden, users wants to stop others from moving the element when they are doing operations, especially operations like line annotating. Therefore, a suitable locking mechanism on the element is implemented in the application. This mechanism only allows that only one user can work on the application at one time. In the application, the following policy is defined. For all users who are manipulating the element, if one of them has a dragging tool, the locking mechanism is used. For this situation, a first come first served rule is used. This means that only the first user who already acquired the element can do operations. Therefore, an unexpected interference from a dragging operation can be avoided. As one exception, if all users currently use annotation tools (e.g. line annotation, point annotation), the locking mechanism is disabled. All users are allowed to do these kinds of operations (e.g. drawing line, doing point annotation) on the element simultaneously. This way, multi-users can work on this application simultaneously for doing annotation. Meanwhile, the users do not need to worry that their work is interrupted by dragging operations from others.

In order to further give feedback on the context awareness, user-dependent annotation is implemented as well. That means the annotations from different users are distinguishable. In the application, the annotations from different users are marked as different colors so that the users can recognize their respective annotations. This way, multi-users can work on this application collaboratively but in different contexts.

6 Result

For the new context-aware system, several new functionalities (Individual tool selection, Locking mechanism, User-dependent annotation) are implemented. For the individual tool selection, each tool is associated with an individual user (see Figure 43).

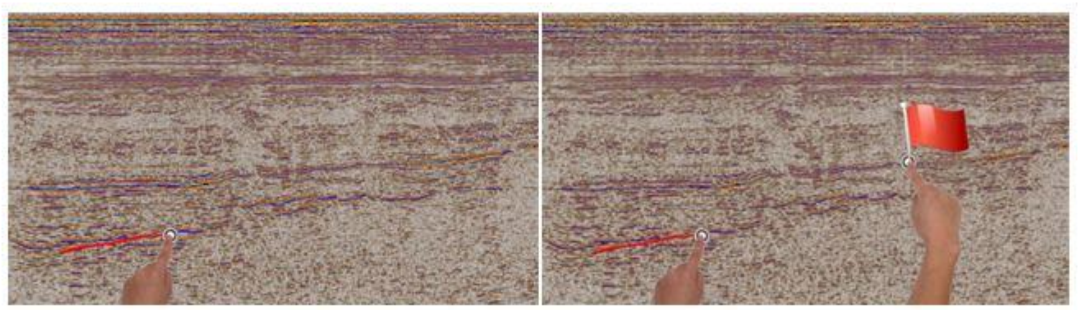
Figure 43: Individual tool selection. Left: One user is doing line annotation. Right: two users are doing annotation simultaneously, but with different annotation tools.

The locking mechanism only allows several users to work on the multi-touch screen simultaneously when they are all using annotation tools. Otherwise, only one user can work on the screen at one time.

Figure 44 shows that two touch points from two users are detected. In a previous version of the multi-touch application, this was explained as a scaling operation. In the current application, this is explained as two dragging attempts from two users. However, due to the locking mechanism, only one of them can do the dragging operation at one time.

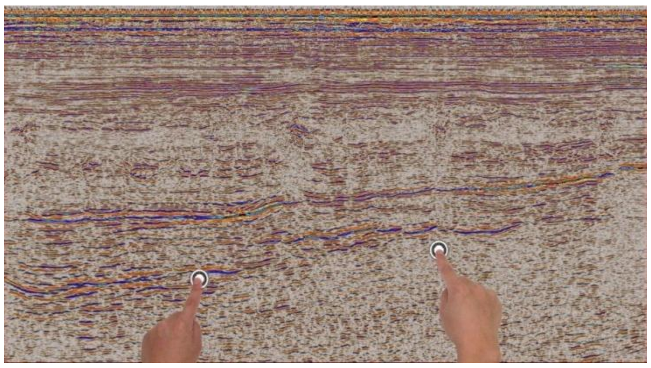
Figure 44: Two touch points from two users.

Figure 45 shows that one user is doing a line annotation, while the other user performs a scaling operation. In this situation, the scaling operation does not work, because the interaction object was locked by the first user.

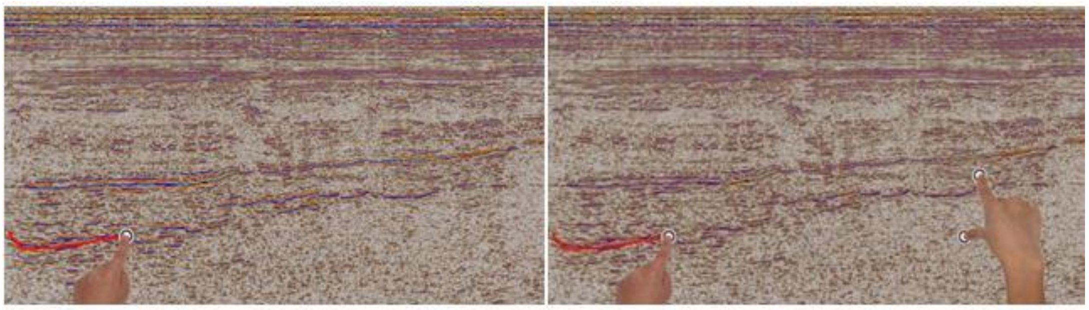

Figure 45: The locking mechanism. Left: one user is drawing a line. Right: another user tries to scale the slice.

Based on these two functionalities, user-dependent annotation is also implemented. The annotations from different users are distinguishable (see Figure 46). Thus, multi-users can work on this system collaboratively but without missing the context information.

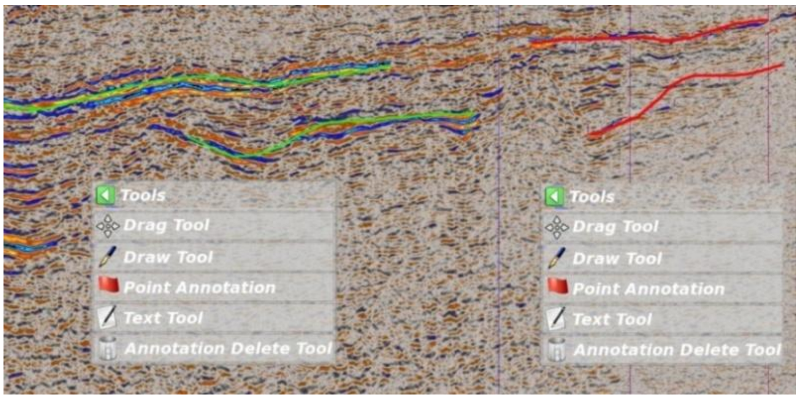

Figure 46: Line annotations from different users. The annotations are distinguished by different colors.

In the new context-aware system, multi-users are able to work on the application simultaneously; different users can be detected by the user tracking application (see Figure 47).

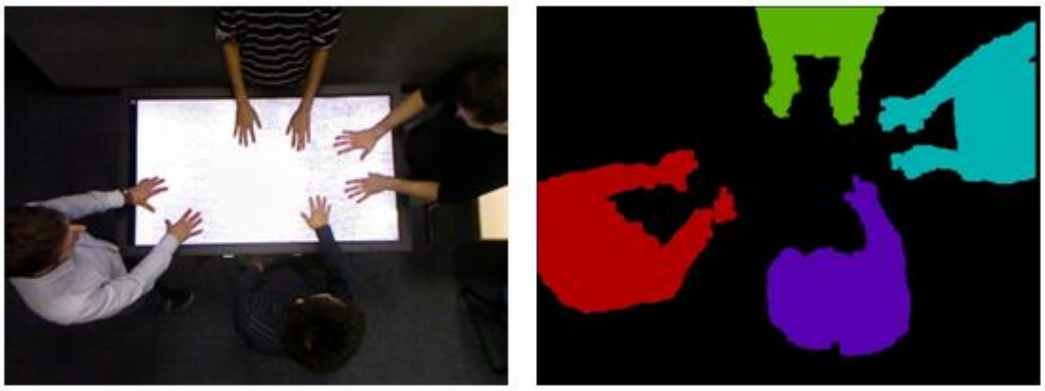

Figure 47: Left: Four users work on the multi-touch table. Right: the users are colored according to user IDs.

Comparing the new context-aware system with the Diamond Touch [6] system, it has two advantages: First, the hardware setup is simpler than Diamond Touch system. In the Diamond Touch system, each chair where a user sits in is equipped with a capacitance for the circuit. Thus, this leads to problems. Users must sit on the chair, because their IDs are bound to the chair. In the new context-aware system, the user IDs is identified by the user tracking application. Second, the users can move flexible. In Diamond Touch, if the users want to move around the table, they need to take similar devices which include capacitances for the circuit which enables the multi-touch detection and touch point association. But in the new context-aware system, users can work more flexible. There is no additional sensor attached to the users themselves. The users around the multi-touch table can either sit or stand when they are working on the multi-touch table.

Comparing with another system which used a webcam as an additional sensor for user tracking (see Section 2.2) described in [7], the new context-aware system performs better again. Because of using the combined segmentation technique which is supported by the depth sensor and infrared sensor of the Kinect, the user tracking application in this new context-aware system is able to overcome the problems of using the color camera for image segmentation. Therefore, it is not necessary to restrict the colors appearing in the background. Furthermore, also certain dynamic changes in the background are possible without any background image update, for example, changing the brightness of the display or the color appeared in the display, because the depth is taken into account. Besides, the combined segmentation method also works well either in a strong artificial light condition (see Figure 48) or in a total dark condition (see Figure 49). Therefore, the user tracking is more robust and flexible in the new context-aware system than in the system in [7].

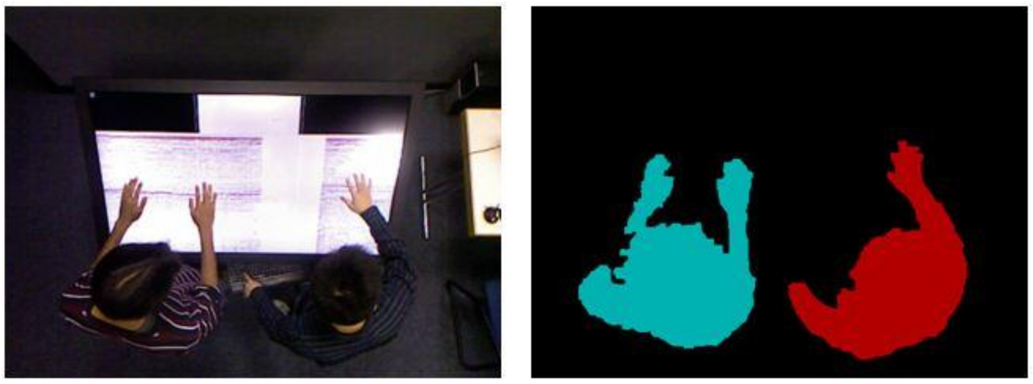
Figure 48: Segmentation result in a strong artificial light condition.

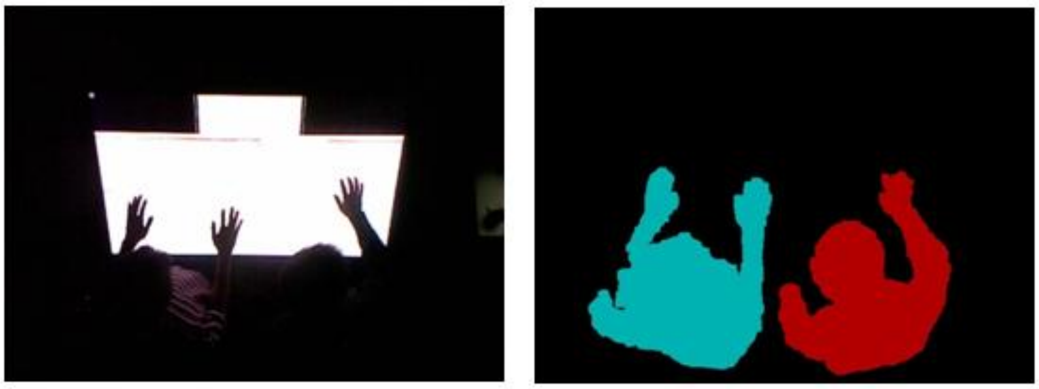
Figure 49: Segmentation result in a total dark condition.

As introduced in Section 5.1, all applications in the VRGeo project are developed for the VRGeo Consortium members. Therefore, the context-aware tabletop system was also evaluated by the experts of the consortium. At the VRGeo Consortium June 2011 meeting, the system was presented to the representatives of the members from the VRGeo consortium. Everybody was given the chance to test the new system. Most of the VRGeo members have funded expertise in the fields of Virtual Reality, geology and geophysics. Although test experience varied from person to person, the overall feedback was very positive. In particular, they positively commented that the problem on when two users are touching the same screen object simultaneously, the relation of their fingers cannot be identified is solved. In the new system, with the user tracking information, the relation of the fingers can be identified.

However, from the feedback received from the VRGeo Consortium members and my own observations at the VRGeo meeting, there are still some limitations in the system.

First problem is still the involuntary touch. Doing operations on the application, the users suffered from this problem. Especially, when they are doing point or line annotation, at this time, their hands are too close to the display. Therefore, the cuffs of their clothes can easily touch the screen.

Another important problem is that the method cannot distinguish users touching each other. In the current system, this means that two different users are seen as one when they are touching each other. This problem happens frequently and unwillingly, especially when there are many users working around the display. Although the problem of user touching caused by bags or something is solved (see in Section 4.3), the upper body parts of the users touching problem still exists (see Figure 50).

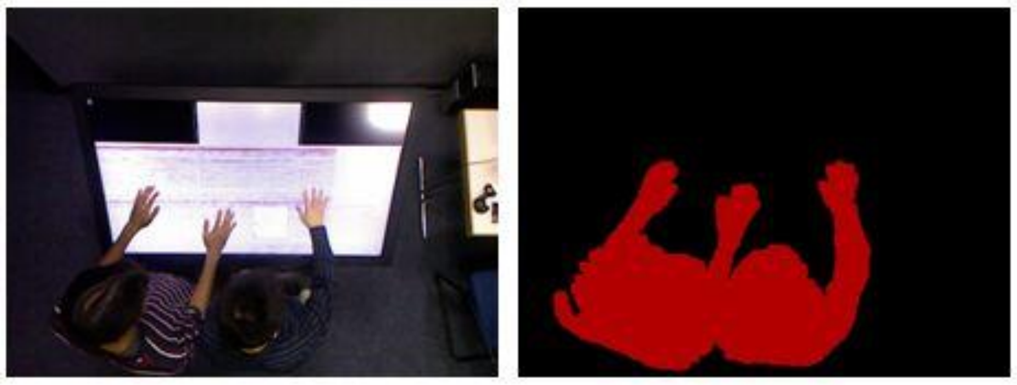

Figure 50: Two touching users have the same user ID.

7 Conclusion and Future work

In my thesis, a context-aware tabletop system based on a Microsoft Kinect was presented. This method includes automatic calibration, combined segmentation, robust user tracking and associating touch points with individual users. With this new method, the users in the environment around the multi-touch table are tracked. Thus, the system can detect different users. With user information, new functionalities are integrated into the VRGeo seismic multi-touch application. The new system enables multi-users work on it collaboratively but without missing the context information.

Despite the encouraging feedback from the VRGeo Consortium member representatives, there are still some problems needed to be solved and many ideas like implementing continuous interaction space mentioned in [33] can be further improvements to the system. In the current system, many users suffer from the involuntary touch problem and the user touching problem. Thus, for the next generation system, these should be the first two issues to be solved.

Another future development could be to implement more interaction possibilities for the system. According to what mentioned in Section 5.3, in the current system, only the individual tool selection, locking mechanism and user-dependent annotation are integrated into the system. However, there are more possible enhancements that can be done for the system (e.g. the orientation of the GUI-Elements). Thus, in future, those functionalities will also be integrated into the current system.

Besides, with taking advantage of the segmented depth information, additional interaction can be implemented. With the depth information, positions of user's hands can be detected, either on the multi-touch screen or above the multi-touch screen. This is a similar concept with the concept of a continuous interaction space as suggested in [14, 1, 33, 10].

8 References

1 Hrvoje Benko and Andrew D. Wilson. Depthtouch: Using depth-sensing camera to enable freehand interactions on and above the interactive surface. Microsoft Research Technical Report, March 2009.

2 G. Bradski. The OpenCV Library. Dr. Dobb's Journal of Software Tools, 2000.

3 Nicolas Burrus. Kinect rgb demo v0.5.0 http://nicolas.burrus.name, June 2011.

4 Fraunhofer IAIS. VRGeo http://www.vrgeo.org, June 2011.

5 Chi Tai Dang, Martin Straub, and Elisabeth Andre. Hand distinction for multi-touch tabletop interaction. In Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces, ITS '09, pages 101–108, New York, NY, USA, 2009. ACM.

6 Paul Dietz and Darren Leigh. Diamondtouch: a multiuser touch technology. In Proceedings of the 14th annual ACM symposium on User interface software and technology, UIST '01, pages 219–226, New York, NY, USA, 2001. ACM.

7 K. C. Dohse, Thomas Dohse, Jeremiah D. Still, and Derrick J. Parkhurst. Enhancing multi-user interaction with multi-touch tabletop displays using hand tracking. In Proceedings of the First International Conference on Advances in Computer-Human Interaction, ACHI '08, pages 297–302, Washington, DC, USA, 2008. IEEE Computer Society.

8 Florian Echtler, Manuel Huber, and Gudrun Klinker. Shadow tracking on multi-touch tables. In Proceedings of the working conference on advanced visual interfaces, AVI '08, pages 388–391, New York, NY, USA, 2008. ACM.

9 Jefferson Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In Proceedings of the 18th annual ACM symposium on User interface software and technology, UIST '05, pages 115– 118, New York, NY, USA, 2005. ACM.

10 Matthew Hirsch, Douglas Lanman, Henry Holtzman, and Ramesh Raskar. Bidi screen: a thin, depth-sensing lcd for 3d interaction using light fields. In ACM SIGGRAPH Asia 2009 papers, SIGGRAPH Asia '09, pages 159:1–159:9, New York, NY, USA, 2009. ACM.

11 Eva Hornecker, Paul Marshall, Nick Sheep Dalton, and Yvonne Rogers. Collaboration and interference: awareness with mice or touch input. In Proceedings of the 2008 ACM conference on Computer supported cooperative work, CSCW '08, pages 167–176, New York, NY, USA, 2008. ACM.

12 Shahram Izadi, Harry Brignull, Tom Rodden, Yvonne Rogers, and Mia Underwood. Dynamo: a public interactive surface supporting the cooperative sharing and exchange of media. In Proceedings of the 16th annual ACM

symposium on User interface software and technology, UIST ’03, pages 159–168, New York, NY, USA, 2003. ACM.

13 M. Kaltenbrunner, T. Bovermann, R. Bencina, and E. Costanza. Tuio - a protocol for table-top tangible user interfaces. In Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation, 2005.

14 Bastian Leibe, Thad Starner, William Ribarsky, Zachary Wartell, David M. Krum, Brad Singletary, and Larry F. Hodges. The perceptive workbench: Toward spontaneous and natural interaction in semi-immersive virtual environments. In Virtual Reality, pages 13–20, 2000.

15 Citron GmbH. dreaMTouch http://www.citron.de/, July 2011

16 Nicolai Marquardt, Johannes Kiemer, and Saul Greenberg. What caused that touch?: expressive interaction with a surface through fiduciary-tagged gloves. In ACM International Conference on Interactive Tabletops and Surfaces, ITS ’10, pages 139–142, New York, NY,USA, 2010. ACM.

17 Chreston Miller, Ashley Robinson, Rongrong Wang, Pak Chung, and Francis Quek. Interaction techniques for the analysis of complex data on high-resolution displays. In Proceedings of the 10th international conference on Multimodal interfaces, ICMI ’08, pages 21–28, New York, NY, USA, 2008. ACM.

18 Meredith Ringel Morris, Anqi Huang, Andreas Paepcke, and Terry Winograd. Cooperative gestures: multi-user gestural interactions for co-located groupware. In Proceedings of the SIGCHI conference on Human Factors in computing systems, CHI ’06, pages 1201–1210, New York, NY, USA, 2006. ACM.

19 Meredith Ringel Morris, Kathy Ryall, Chia Shen,Clifton Forlines, and Frederic Vernier. Beyond”social protocols”: multi-user coordination policies for collocated groupware. In Proceedings of the 2004 ACM conference on Computer supported cooperative work,CSCW ’04, pages 262–265, New York, NY, USA, 2004. ACM.

20 ROS.ORG. http://www.ros.org/wiki/kinect_calibration/technical, July 2011.

21 Peter Peltonen, Esko Kurvinen, Antti Salovaara, Giulio Jacucci, Tommi Ilmonen, John Evans, Antti Oulasvirta, and Petri Saarikko. It’s mine, don’t touch!: interactions at a large multi-touch display in a city centre. In Proceeding of CHI08, pages 1285–1294, New York, NY, USA, 2008. ACM.

22 Meredith Ringel, Kathy Ryall, Chia Shen, Clifton Forlines, and Frederic Vernier. Release, relocate, reorient, resize: fluid techniques for document sharing on multiuser interactive tables. In CHI ’04 extended abstracts on Human factors in computing systems, CHI EA ’04, pages 1441–1444, New York, NY, USA, 2004. ACM.

23 Volker Roth, Philipp Schmidt, and Benjamin Güldenring. The ir ring: authenticating users’ touches on a multi-touch display. In Proceedings of the

23nd annual ACM symposium on User interface software and technology, UIST '10, pages 259–262, New York, NY, USA, 2010. ACM.

24 Kathy Ryall, Clifton Forlines, Chia Shen, and Meredith Ringel Morris. Exploring the effects of group size and table size on interactions with tabletop shared display groupware. In Proceedings of the 2004 ACM conference on Computer supported cooperative work, CSCW '04, pages 284–293, New York, NY, USA, 2004. ACM.

25 Stacey D. Scott, Karen D. Grant, and Regan L. Mandryk. System guidelines for co-located, collaborative work on a tabletop display. In Proceedings of the eighth conference on European Conference on Computer Supported Cooperative Work, pages 159– 178, Norwell, MA, USA, 2003. Kluwer Academic Publishers.

26 Stacey D. Scott, M. Sheelagh, T. Carpendale, and Kori M. Inkpen. Territoriality in collaborative tabletop workspaces. In Proceedings of the 2004 ACM conference on Computer supported cooperative work, CSCW '04, pages 294–303, New York, NY, USA, 2004. ACM.

27 Chia Shen, Katherine Everitt, and Kathleen Ryall. Ubitable: Impromptu face-to-face collaboration on horizontal interactive surfaces. In Ubicomp, pages 281– 288, 2003.

28 Chia Shen, Frederic D. Vernier, Clifton Forlines, and Meredith Ringel. Diamondspin: an extensible toolkit for around-the-table interaction. In Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '04, pages 167–174, New York, NY, USA, 2004. ACM.

29 Jason Stewart, Benjamin B. Bederson, and Allison Druin. Single display groupware: a model for copresent collaboration. In Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit, CHI '99, pages 286–293, New York, NY, USA, 1999. ACM

30 Norbert Streitz, Thorsten Prante, Christian Müller-Tomfelde, Peter Tandler, and Carsten Magerkurth. Roomware&#169;: the second generation. In CHI '02 extended abstracts on Human factors in computing systems, CHI EA '02, pages 506–507, New York, NY, USA, 2002. ACM.

31 Peter Tandler, Thorsten Prante, Christian Mueller- Tomfelde, Norbert Streitz, and Ralf Steinmetz. Connectables: dynamic coupling of displays for the flexible creation of shared workspaces. In Proceedings of the 14th annual ACM symposium on User interface software and technology, UIST '01, pages 11–20, New York, NY, USA, 2001. ACM.

32 Jens Teichert, Marc Herrlich, Benjamin Walther-Franks, Lasse Schwarten, and Markus Krause. User detection for a multi-touch table via proximity sensors. Proceedings of the IEEE Tabletops and Interactive Surfaces, 2008.

33 The Continuous Interaction Space: Interaction Techniques Unifying Touch, Gesture On, and Above a Digital Surface. The perceptive workbench: Toward spontaneous and natural interaction in semi-immersive virtual envi-

ronments. In In Proceedings of the 13th IFIP TCI3 Conference on Human Computer Interaction – INTERACT 2011, Pages 5–9, 2011.

34 Edward Tse, Jonathan Histon, Stacey D. Scott, and Saul Greenberg. Avoiding interference: how people use spatial separation and partitioning in sdg workspaces. In Proceedings of the 2004 ACM conference on Computer supported cooperative work, CSCW ’04, pages 252– 261, New York, NY, USA, 2004. ACM.

35 Andrew 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.

36 Mike Wu and Ravin Balakrishnan. Multi-finger and whole hand gestural interaction techniques for multiuser tabletop displays. In Proceedings of the 16th annual ACM symposium on User interface software and technology, UIST ’03, pages 193–202, New York, NY, USA, 2003. ACM.

37 OpenKinect project. http://openkinect.org/wiki/Main_Page, June 2011

38 Benko, H., Saponas, T.S., Morris, D., and Tan, D. Enhancing input on and above the interactive surface with muscle sensing. Proc. of ITS, ACM (2009), 93-100.

39 S. Camille Peres, Vickie Guye, and Magdy Akladios. Geophysical software ergonomics: Objective measures for evaluation, June 2011.

40 Micosoft. Microsoft Kinect, http://www.xbox.com/kinect, June 2011.