Enabling Context Aware Multi-User Interaction with the Microsoft Kinect

David d'Angelo *
Fraunhofer IAIS

Alexander Kulik†
Bauhaus-University Weimar

Markus Schlattmann‡
AGT Group *R&D* GmbH

Manfred Bogen§
Fraunhofer IAIS

ABSTRACT

Multi-touch has become a popular technology used for numerous applications in various domains. We present a novel method based on off-the-shelf sensors for associating detected touch-points with individual users. An additional depth camera above the tabletop device tracks the users around the table and their respective hands. This environment tracking and the multi-touch sensor are automatically calibrated to the same coordinate system. We explored the resulting advantages for multi-touch applications, including the reduction of false positives and we present an application combining user aware multi-touch interaction with an immersive 3D visualization.

Index Terms: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism Animation—Virtual Reality; I.3.8 [Computer Graphics]: Applications—; H.5.6 [INFORMATION INTERFACES AND PRESENTATION]: Group and Organization Interfaces—Synchronous interaction; I.4.6 [IMAGE PROCESSING AND COMPUTER VISION]: Segmentation—Pixel classification;

1 INTRODUCTION

Most existing multi-touch (MT) systems suffer from missing context information. If a multi-touch system detects two touch points on the screen, it generally cannot distinguish whether the touch points belong to one hand, two hands or even different users. Therefore, multiple users and multiple hands can only work in the same context, which often results in interference [17, 13].

To solve these problems and allow a more natural collaboration, some previous research systems already included additional environment sensors (e.g. a ceiling mounted camera) [6, 5, 7]. However, these systems all suffered from severe limitations, either restricting the surrounding of the tabletop or even the movements/locations of the users themselves. Our novel system based on a depth camera (Microsoft Kinect) provides additional information that enables reliable and robust environment tracking. The resulting context information for detected touch-points at the interactive surface provides many new possibilities to improve the expressiveness of multi-touch gestures and realize software-supported multi-user multi-touch input coordination. In particular, we identify the following applications:

- Individual users can associate different tools to their input; thus different functions may even be operated simultaneously by cooperating users.
- Software-controlled access management eliminates involuntary interference (e.g. operations like dragging or scaling an object may block access for other users)
- Automatic partitioning of the screen and input space with respect to the users' positions.
- User oriented visualization of GUI elements improve legibility (e.g. menus or text output)
- Occlusion-awareness: relevant GUI elements can be relocated at the screen if occluded by other users.

We used a number of standalone demonstrations to explore the usability of these functionality. The features we identified as most relevant were furthermore integrated into our scientific visualization system which was tested by a consortium of experts in geology. In the next chapter we will discuss related work, followed by an explanation of our implementation in detail and a description of our combination of user aware multi-touch and an immersive VR visualization system.

2 RELATED WORK

2.1 Direct Collaboration at the Tabletop

Tabletops offer an ideal setting for collaborative interaction. Attendees can communicate face to face while exchanging information and objects on the shared horizontal surface. Consequently, this setup has long been proposed for computer mediated collaboration (e.g.: [4, 15, 14]). Developing appropriate interfaces, however, is challenging. Consider common user actions like maximizing a GUI element or panning the workspace. Direct access to global transformations is clearly beneficial for single user workplaces, but in the context of co-located collaboration results affect others too. The power and expressiveness of well established interaction patterns for graphical user interfaces easily evoke conflicts in collaborative settings (see [17, 13]).

It has been observed that territoriality plays a major role for the coordination between attendees sharing one interaction space [16, 18]. However, if a task requires joint activity at a shared focus of attention, further negotiation concepts become necessary. Ringel et al. proposed [14] a set of document sharing techniques that build on territoriality, but also on user identification and simple transitional gestures to avoid involuntary interference. Higher level coordination strategies for the negotiation of interfering user actions have been developed by Morris et al. [12]. However, none of these input coordination techniques can be implemented solely relying on multi-touch sensors. For user identification further context information is required.

Marquard et al. recently demonstrated the benefits of a robust association between touch-points and the hands of the users [11]. Using a glove that tagged features of the hand with unambiguous optical markers, they implemented hand gesture recognition, multi-user coordination policies and adversity of drawing tools that could be associated with individual fingers. In the following section we discuss the benefits and drawbacks of multi-touch systems offering context awareness.

2.2 Context-Aware Multi-Touch Systems

DiamondTouch is a commercially available multi-touch sensor device [4] offering the association of touch input to respective users by coupling electric signals from the tabletop, through the user's

*e-mail: david.d-angelo@iais.fraunhofer.de
†e-mail: kulik@uni-weimar.de
‡e-mail: mschlattmann@agtgermany.com
§e-mail: manfred.bogen@iais.fraunhofer.de

body, into a distinct receiver for each user. Many researchers implemented multi-user coordination policies using this system (e.g. [4, 12, 14]. Unfortunately the system limits the choice of display components to front projection. It is furthermore limited to four users that must maintain physical contact to the corresponding receiver unit while avoiding to touch each other.

Recently, alternative systems have been proposed. Dohse et al. used frustrated total internal reflection (FTIR see [8]) to provide multi-touch. For association of touch-points with users, an additional camera is mounted above the FTIR tabletop display. Using this camera, the hands are tracked above the display using color segmentation or shadow tracking respectively [5]. 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.

A very promising system has been designed by Walther-Franks et al [7]. They embedded proximity sensors to the frame of the tabletop device. This system provides rough user tracking that is sufficient for many purposes. However, robust correlation of touch-points with a user's hand cannot be ensured. Touchpoints 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 her proximity.

3 System Setup

In our setup, we mounted a Microsoft Kinect device approximately 2 meters above a multi-touch table (see Fig. 1). In this configuration, the Kinect captures the entire screen area of the multi-touch system and about 50 cm of its surrounding in each direction. A standard desktop PC is used to drive the application on the multi-touch table. A second machine runs the environment tracking. Both workstations share the recorded user input data via a network using the TUIO protocol [9].

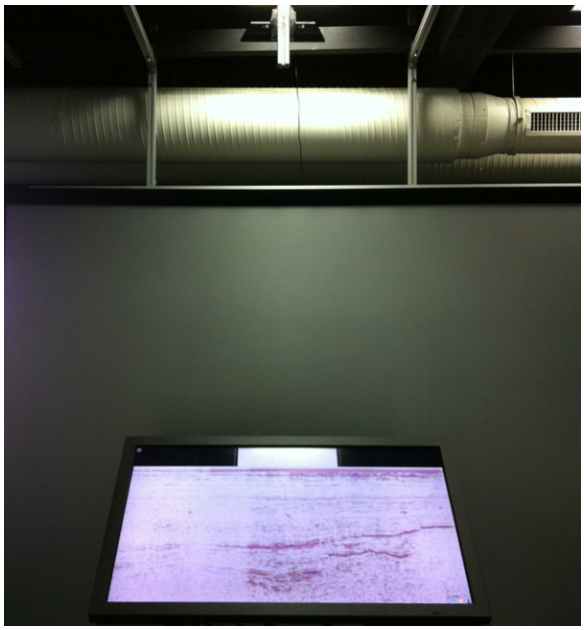

Figure 1: The arrangement of the Multi-touch table (bottom) with the Microsoft Kinect (top).

3.1 Automatic Calibration

The internal calibration of the different sensors embedded in the Kinect (color, IR, depth) must be performed only once as their relation does not change over time. We use the method published by Nicolas Burrus for this purpose [3]. The extrinsic parameters defining the relation between the display area and one of the Kinect sensors, instead, must frequently be re-calibrated. These parameters have to be re-computed every time the multi-touch table or the Kinect has been moved. Due to the adaptability of our assembly this happens quite often. To this end we implemented a fully automatic calibration routine without the necessity of any additional calibration object such as a printed chessboard pattern. The screen itself is employed to display a calibration pattern (chessboard) that can be recorded by the color sensor of the Kinect. From this image data, the transformation matrices can be derived using the calibration method available in OpenCV [2]. Note that using the color sensor of the Kinect is mandatory for this procedure due to the fact that the calibration pattern on the display is not visible for the infrared or depth sensor of the Kinect. If desired, the calibration can be triggered automatically if the 3 axis accelerometer attached to the multi-touch table reports a change of inclination angle.

3.2 Combined Segmentation

The sensors of the Microsoft Kinect enable different ways of segmenting the foreground (user body parts) from the background (e.g. floor and display). However, every method has its drawbacks. Using the color information e.g. for skin color segmentation or background subtraction severely restricts the colors allowed in the background. As this includes the display content itself such a restriction is a no-go criterion. Note that cancelling the light from the displays by means of polarization filters as in [5] is not feasible in such a highly adaptive setup. Using depth information, instead, for segmentation between fore- and background has several advantages. Only objects occluding the view of the Kinect must be avoided. However, further issues have to be solved for robust operation. The imprecision and quantization of depth values obtained by the Kinect impede precise depth segmentation close to the display surface. Furthermore, the effective image resolution of the depth image is low (approximately 320x240), wherefore small body parts (e.g. fingers or small hands) tend to vanish in the segmentation. These problems are illustrated in Fig. 2.

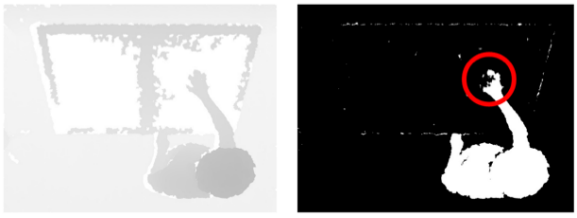

Figure 2: Depth-based segmentation leads to problems near the display surface. Left: depth image. Right: depth-based segmentation with missing foreground inside the red circle.

A third way of segmentation is using a background subtraction algorithm on the infrared (IR) intensity information from the Kinect. This has the advantage of a higher resolution (640x480) and works also close to the display surface as the display does not emit infrared light. However, 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 strongly affect the segmentation result (see Fig. 3)

Because of these problems we finally came up with a combination of depth and infrared segmentation. Using a logical or-operation, we combine the depth-based segmentation of the entire image with the infrared segmentation, but only inside the image region corresponding to the display surface. Fig. 4 illustrates the resulting segmentation performance.

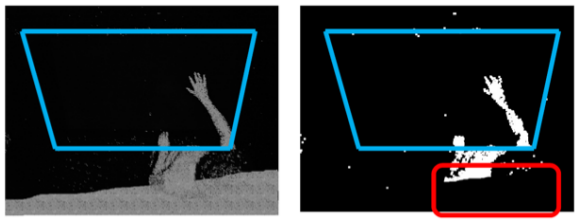

Figure 3: Infrared-based segmentation can induce problems around the display. The blue rectangle indicates the display region. Left: IR image. Right: IR-based segmentation with missing foreground in the red rectangle.

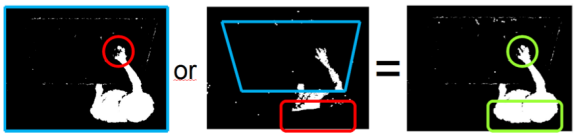

Figure 4: A combination of depth segmentation (left image) and IR-intensity segmentation (middle image) leads to better segmentation results (right image) and solves the respective problems (see regions in green circles)

3.3 User Separation and Tracking

After the segmentation, only noise and those regions belonging to the users remain in the image. In this image we search for the largest connected components. A simple threshold on the minimum component size filters artifacts of noise. The remaining connected components (CC) can be assumed to correspond to a single user. They are assigned a user ID and tracked over time. A drawback of this approach is that as soon as two users touch or occlude each other, their separate CCs will merge. Thus, we apply a second processing step to separate the user regions also if a CC comprises multiple users. We exploit the depth information provided by the the Kinect camera to identify the upper body of each user. Our algorithm searches for height peaks within each connected component that are at least 40 cm above than the known height of the tabletop surface. If only one of such peaks exists, the entire component is interpreted as a single user region. Otherwise, the region is separated into individual areas surrounding each peak using a simple heuristic based on frame coherence.

3.4 Associating Touch-Points to Users and Hands

Finally, for each touch-point the corresponding user and hand have to be estimated. To this end, we project the touch-points to the image containing the user regions using the transformation matrix derived from the calibration (see Fig. 5). If a touch-point is located inside a user region, the corresponding user ID is transcribed directly. Otherwise the closest user region is computed using a breadth-first search. We further distinguish both hands of a user based on the geodesic distances of touch-points in the graph representing the user component. Our procedure builds on the Dijkstra algorithm to find the shortest path between the touch-points assigned to one user. The procedure stops either if all the other touch-points of the respective user have been found or if the Dijkstra radius exceeds a certain threshold Δ (30 cm). The resulting clusters are interpreted as individual hands. Unlike euclidean distances, geodesic distances support robust clustering even if both hands are in close proximity (unless they merge to one region in the camera images).

3.5 Identification of Involuntary Touches

If more than two clusters are found, we ignore those with the lowest frame coherence as the respective touch events most likely occurred

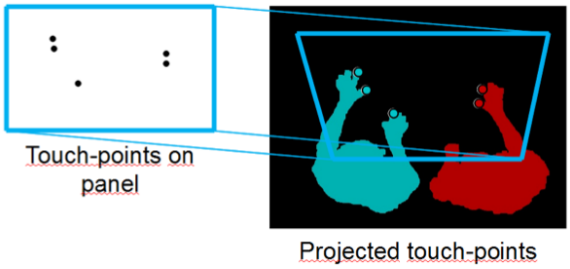

Figure 5: Projection and assignment of the touch-points (black dots on the left) to the image containing the user regions (right). The blue rectangle indicates the area of the display surface. The projected touch-points (colored dots with black margin on the right) are assigned to the closest user region.

involuntarily. We also classify a touch-point to be involuntary if the area of the touch footprint on the multi-touch panel is larger than a certain threshold Δ (5 cm^2). This way, touches too large to originate from a finger or soft-touch stylus are ignored. This simple thresholding allows users to rest their the hands on the tabletop surface while operating with finger or stylus. A one-time invalid touch-point stays invalid. We also tried other mechanisms allowing touch-points to become valid again such as (adaptive) hysteresis thresholding. However, we observed this simple mechanism to work best. We observed that voluntary touches were hardly ever classified as involuntary.

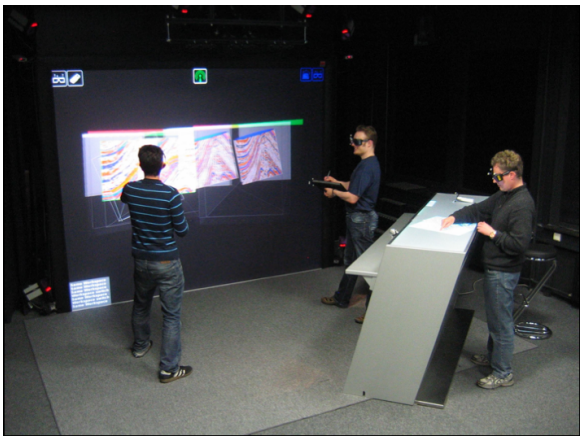

Figure 6: Three users simultaneously using a geoscientific application, which combines a user aware multi-touch system with an 3D VR system for seismic interpretation tasks.

4 Application

We developed a geoscientific application, which combines an immersive environment with the user aware multi-touch system described in the previous section (see Fig. 6). The system provides a 3D as well as a 2D interface transparently sharing the same database. For example, a user can create seismic slices in the 3D environment and they are automatically linked with with 2D multi-touch system, where they can be interpreted. Using this setup, tasks like extraction and positioning of seismic slices are done by using the immersive 3D interface, whereas the 2D multi-touch interface is used for interpretation tasks, like fault or horizon picking. While using the 2D interface, the immersive environment serves as a 3D reference for the 2D task execution. All interpretations made on the multi-touch display are automatically synced and displayed in

the 3D environment. This way a user can always check if the 2D interpretation still fits with the whole 3D picture. Since all users are tracked in the 3D environment, and our algorithm matches the hands respectively the fingers to users, all annotations can be bound to the specific users.

5 Conclusion and Future Work

We presented a novel method to achieve context awareness for large interactive tabletops based on the sensor data from a off-the-shelf depth camera. This method includes automatic recalibration, sensor-fused segmentation, separation of touching users, robust hand identification based on geodesic distances and a detection method for involuntary touches. In addition, we showed an application where we successfully integrated such a system with an immersive multi-user environment.

In the future we are planing to extend our framework to support a continuous interaction space as suggested in [10, 1]. This also includes using additional hard- and software to discern the hands and arms of the users above the tabletop.

Acknowledgements

This work was funded by the VRGeo (Virtual Reality for the Geosciences) Consortium. The authors want to thank all VRGeo members for their support and interesting discussions in the context of this work.

References

[1] H. Benko and A. D. Wilson. Depthtouch: Using depth-sensing camera to enable freehand interactions on and above the interactive surface. Technical Report MSR-TR-2009-23, Microsoft Research Technical Report, East Lansing, Michigan, March 2009.

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

[3] N. Burrus. Kinect rgb demo v 0.5.0. `http://http://nicolas.burrus.name/`, June 2011.

[4] P. Dietz and D. Leigh. Diamondtouch: a multi-user 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.

[5] K. C. Dohse, T. Dohse, J. D. Still, and D. 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.

[6] F. Echtler, M. Huber, and G. 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.

[7] B. W. Franks, L. Schwarten, J. Teichert, M. Krause, and M. Herrlich. User Detection for a Multi-touch Table via Proximity Sensors. In *IEEE Tabletops and Interactive Surfaces 2008*. IEEE Computer Society, 2008.

[8] J. 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.

[9] M. Kaltenbrunner, T. Bovermann, R. Bencina, and E. Costanza. Tuio: A protocol for table-top tangible user interfaces. In *6th International Gesture Workshop*, 2005.

[10] B. Leibe, T. Starner, W. Ribarsky, Z. Wartell, D. Krum, B. Singletary, and L. Hodges. The perceptive workbench: Toward spontaneous and natural interaction in semi-immersive virtual environments. In *Proceedings of the IEEE Virtual Reality 2000 Conference*, VR '00, pages 13–, Washington, DC, USA, 2000. IEEE Computer Society.

[11] N. Marquardt, J. Kiemer, and S. 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.

[12] M. R. Morris, A. Huang, A. Paepcke, and T. 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.

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

[14] M. Ringel, K. Ryall, C. Shen, C. Forlines, and F. Vernier. Release, relocate, reorient, resize: Fluid techniques for document sharing on multi-user interactive tables. In *Abs. of the ACM Conference on Human Factors in Computing Systems*, pages 1441–1444. ACM Press, 2004.

[15] S. D. Scott, K. D. Grant, and R. 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.

[16] S. D. Scott, M. Sheelagh, T. Carpendale, and K. 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.

[17] J. Stewart, B. B. Bederson, and A. Druin. Single display groupware: a model for co-present 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.

[18] E. Tse, J. Histon, S. D. Scott, and S. 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.