
Seamless Integration of Mobile Devices into Interactive Surface Environments

Andreas Dippon
Technische Universität
München
Boltzmannstr. 3
85748 Garching
dippona@in.tum.de

Norbert Wiedermann
Technische Universität
München
Boltzmannstr. 3
85748 Garching
wiederma@in.tum.de

Gudrun Klinker
Technische Universität
München
Boltzmannstr. 3
85748 Garching
klinker@in.tum.de

Copyright is held by the author/owner(s).
ITS'12, November 11–14, 2012, Cambridge, Massachusetts, USA.
ACM 978-1-4503-1209-7/12/11.

Abstract

This poster abstract describes the seamless integration of uninstrumented mobile devices into an interactive surface environment. By combining a depth camera with a RGB camera for tracking, we are able to identify uninstrumented mobile devices using visual marker tracking. We describe the technical details of combining the two cameras and an example application for the integration of mobile devices.

Author Keywords

Multitouch; Depth-sensing cameras; Object tracking; Interactive Surface.

ACM Classification Keywords

H.5.2 [Information interfaces and presentation: User Interfaces. - Graphical user interfaces.]

INTRODUCTION

With the introduction of the Kinect, depth cameras got a lot of attention recently. It was shown, that such cameras can be used to detect touches on arbitrary displays [12, 2]. In addition, tangible objects on surfaces can be detected. Yet the depth data has the drawback that similar looking devices (e.g. mobile phones) can't be distinguished. This drawback can be reduced by combining the depth camera with the

built-in RGB camera and computer vision algorithms. Using this combination we can detect and distinguish different mobile devices (e.g. mobile phones, tablet computers) without any additional instrumentation. This can be used to seamlessly integrate mobile devices into an interactive surface environment.

RELATED WORK

One interesting approaches from Wilson [12] was to use a depth camera for sensing touches on an arbitrary surface. He already mentioned the possibility of tracking objects in addition to finger touches. Klompaker et al [8] developed object tracking by using a history of depth images and additional background images. With this method, they can already detect touches, arbitrary objects and touches on objects very consistently. Yet it is not possible to distinguish similar objects from each other.

Combining mobile devices and interactive surfaces has already seen a lot of attention in recent years [6, 10, 11, 4]. Wilson used a visual handshaking process on the BlueTable [11] where mobile phones have to be placed on the surface and be visible for an above mounted camera. Echtler [4] improved the system by identifying the phone using the signal strength of the bluetooth connection instead of a visual handshake. Yet both systems have the drawback that the phones have to be placed and stay on the surface, which causes privacy issues and an inferior usability of the mobile phone. Chehimi et al [1] improved the approach by creating a proxy circular area on the surface when a phone connects to it. This area is created in the middle of the surface and users have to be aware that the created proxy is representing their mobile phone. While this method is already well

integrated, the mental load on the user is increased as users are part of the registration process. Additionally the identification of the correct proxy becomes difficult when several users want to join at the same time.

Schmidt et al [9] used a different solution, where the mobile phones have to touch the surface to trigger an interaction. By using timestamps, different mobile phones can be distinguished if they don't touch the surface at the same time. The system works quite well, yet users were uncomfortable touching surfaces with their mobile devices. The same drawback can be found in the system used by Izadi et al. [7], where the built-in camera of the mobile phone is used to identify itself with the bluetooth connection to the surface.

DEPTH AND RGB TRACKING

We use the depth data for tracking objects and fingers on and above any surface by calculating the difference between the depth values in the image and a previously captured background image [12, 2]. To be able to combine the tracking data of the RGB camera and the depth camera, the two images have to be aligned in order to match marker and blob positions. As the Kinect has the RGB camera built-in, this alignment is approximately the same for all devices. As the accuracy of the tracked marker positions in the RGB image is not critical, a predefined calibration for shifting the RGB image to the depth image is sufficient. This approximation has the advantage that the system doesn't have to be calibrated for different Kinects. For the tracking of visual markers we integrated a very robust visual marker tracker from the Ubitrack library by Huber et al. [5] into the libtisch framework [3]. This combination enables us to reliably track the positions, rotations and identification numbers of visual markers in the image.

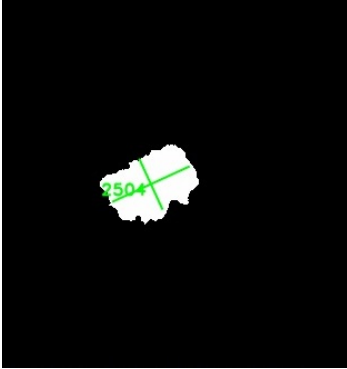


Figure 1: Detected Blob in Depth Image.



Figure 2: Detected Marker on Mobile Phone in RGB Image.

The next step is to combine the information of a marker with an object blob of the depth camera. This is done for each detected marker position by searching for the nearest blob within a modifiable search radius. If a blob is found, the marker information is added to the other characteristics of the blob. If no blob is found in the vicinity of a marker, the system can be adjusted to simply discard the marker or to automatically generate a blob on the projected position on the surface. The position of the generated marker blob won't be as accurate as an object tracked by the depth camera because of the predefined camera calibration. Yet it can still be useful especially if you want to track reflective displays (e.g. mobile phones) which often can't be detected reliably by the depth camera.

EXAMPLE: DATATRANSFER WITH PHONES

User Experience

Friends want to share and exchange pictures from their mobile phones with each other on a large interaction surface. They simply start the application on their mobile phone to connect to the surface display. By holding their mobile phone above the surface, it gets automatically registered by the depth camera which is mounted above the interactive surface. During the registration of the mobile phones, a visual representation of each phone is generated on the surface at the position of the corresponding phone. The users can now select pictures on their mobile phones that they want to share with their friends without the requirement to be tracked anymore. The selected pictures will be shown on the surface next to the users' mobile phone representation. The friends can now look at and interact (zoom, rotate, move) with

the pictures on the interactive surface. If they want to copy the picture to their phone, they simply can drag it on their mobile phone representation on the surface. Alternatively they can hold their mobile phone above a picture and press a copy button on the mobile application.

Implementation

We developed an Android application (AA) and an application for the interactive surface (ISA). The user simply needs to install and run the Android application to be able to instantly work with the interactive surface. When started, the AA sends a greeting message to the already running ISA. The ISA responds by sending an available random marker identification number (ID) from a predefined ID list to the device and locks this marker ID. The AA generates a visual marker from this ID, shows it on its display and waits for a confirmation message of the ISA. As soon as the ISA detects the marker and combines it with a blob (or creates a new blob for this marker), it sends a confirmation message to the mobile phone that the connection is established and the application is ready to use. Additionally the ISA creates a visual representation for the mobile phone on the position of the blob with the corresponding marker ID. After receiving the confirmation message of the ISA, the AA shows a menu with different pictures instead of the generated marker. Then the user can select pictures which should be shown on the interactive surface. The selected pictures are shown on the interactive surface and can be scaled, moved and rotated by touch interaction. In order to copy pictures to another phone, the pictures can simply be dragged onto the visual representation of another phone which was created during the registration process for each mobile phone. Pictures can also be copied by holding



Figure 3: User interacts with his mobile phone and the interactive surface.

a mobile phone above a picture and pressing the copy button in the AA. This will show the previously generated marker on the display of the mobile phone until it is detected by the ISA. The ISA then sends the picture and as soon as the mobile phone receives the data, it will show the pictures menu again. If a user wants to leave, the user can simply quit the AA which will send a farewell message to the ISA. The visual representation and all data coming from the corresponding phone will be removed from the interactive surface and the locked marker ID will be available again for another phone.

DISCUSSION

For the identification process of a mobile device it is sufficient, that the marker is detected in only one frame. Therefore this process is often not even recognized by the users and enables a seamless integration of mobile devices. Due to the simple registration process of the mobile devices, this method could also be very well integrated into games (e.g. card games) that use the interactive surface as the playground and the mobile devices as private displays to show the cards of a player [10].

CONCLUSION

We presented a technique to seamlessly integrate uninstrumented mobile devices into an interactive surface environment. We believe that the combination of the two cameras has a lot of potential in upcoming applications and think that the described fundamentals help other researchers to develop further ideas.

ACKNOWLEDGMENTS

Part of the CRUMBS project, funded by BMWi.

References

- [1] Chehimi, F., and Rukzio, E. Throwing Gesture as a Way for Photo Sharing between Mobile Phones and Interactive Tables. In *4th Intermedia Open Forum* ('10).
- [2] Dippon, A., and Klinker, G. KinectTouch: Accuracy Test for a Very Low-Cost 2.5D Multitouch Tracking System. In *ITS '11* (Nov. 2011).
- [3] Echtler, F., and Klinker, G. A multitouch software architecture. In *NordiCHI '08* (Oct. 2008).
- [4] Echtler, F., and Klinker, G. Tracking mobile phones on interactive tabletops. In *MEIS '08* (Sept. 2008).
- [5] Huber, M., Pustka, D., Keitler, P., Echtler, F., and Klinker, G. A system architecture for ubiquitous tracking environments. In *ISMAR '07* (Nov. 2007).
- [6] Izadi, S., Brignull, H., Rodden, T., Rogers, Y., and Underwood, M. Dynamo: A public interactive surface supporting the cooperative sharing and exchange of media. In *UIST '03* (Nov. 2003).
- [7] Izadi, S., Butler, A., Hodges, S., West, D., Hall, M., Buxton, B., and Molloy, M. Experiences with building a thin form-factor touch and tangible tabletop. In *Tabletop 2008* (Oct. 2008).
- [8] Klompmaker, F., Nebe, K., and Fast, A. dSensingNI: A Framework for Advanced Tangible Interaction using a Depth Camera. In *TEI '12* (Feb. 2012).
- [9] Schmidt, D., Seifert, J., Rukzio, E., and Gellersen, H. A Cross-Device Interaction Style for Mobiles and Surfaces. In *DIS '12* (June 2012).
- [10] Shirazi, A. S., Döring, T., Parvahan, P., Ahrens, B., and Schmidt, A. Poker surface: Combining a multi-touch table and mobile phones in interactive card games. In *MobileHCI '09* (Sept. 2009).
- [11] Wilson, A., and Sarin, R. Bluetable: connecting wireless mobile devices on interactive surfaces using vision-based handshaking. In *GI '07* (May 2007).
- [12] Wilson, A. D. Using a depth camera as a touch sensor. In *ITS '10* (Nov. 2010).