Replicating Interactive Surfaces Using Distortion Techniques

Jooyoung Lee\textsuperscript{1}, Johannes Luderschmidt\textsuperscript{1}, Ralf Dörner\textsuperscript{1}, HyungSeok Kim\textsuperscript{2,3}, JeeIn Kim\textsuperscript{2}

Dept. DCSM, RheinMain University of Applied Sciences, Germany\textsuperscript{1}
Dept. of Internet & Multimedia Engineering, Konkuk University, Seoul, Korea\textsuperscript{2}
Institute for Media Innovation, Nanyang Technological University, Singapore\textsuperscript{3}

Abstract

In this paper we present a distortion technique that provides a focus & context view of an interactive surface’s screen on a mobile device. Simply showing a reduced version of the surface’s screen on the mobile device would not have been sufficient as UI elements could be too small to be manipulated. Users modify the region of interest (ROI) of the focus & context distortion via gestural input on their device. We employ this technique in our system that transmits interaction on the mobile device via TUIO to the interactive surface. Thus, users may interact remotely with any TUIO-based application on the surface without additional implementation effort while real-time constraints are still met.

1 Introduction

We consider a basic scenario where a user would like to interact with an interactive surface without touching it directly. For instance, a large display running an interactive application could be placed behind a shop window and a user standing in front of the window wants to interact with it. Another example is a user standing in front of a large interactive tabletop, who is not able to physically reach the far corner of the tabletop. Or a user is not willing to touch a public interactive kiosk system because of hygienic concerns. Our principal idea is to employ a mobile device (e.g., a tablet computer or smartphone) as a proxy for the interactive surface allowing for remote interaction. A usable realization of this general idea needs to provide answers to the following three questions. First, how is the large interactive surface represented on the small display of the mobile device? Second, how does the user interact with this representation? Third, what modifications of the application on the interactive surface need to be performed in order to facilitate such remote interaction? In this paper, we present a solution approach to answer all three questions. In the next section, we present our
solution after briefly reviewing related work. In section 3 we present the implementation. Section 4 gives a conclusion.

Figure 1: An example for our distortion technique: (left) Undistorted image (with marked zones). (middle) Image distorted with our focus & context technique for a tablet computer. Zones have been transformed to fit in the 9 zones. (right) Distorted image on smartphone. Note that ROI is still well recognizable.

2 Concept and Architecture

Different approaches allow users to employ mobile devices to interact with interactive surfaces. For instance a mobile device can be used as a magic lens or as a precise input device on an interactive tabletop setup (Olwal & Feiner 2009). They can be used as data input palettes for interactive whiteboards (Rekimoto 1998) or to interact with public displays via throw and tilt gestures (Dachselt & Buchholz 2009). These approaches however demand interactive surface applications that are especially adapted to work with such devices. If users want to use mobile devices for interaction with unmodified applications, they can for instance employ the TuioPad\(^1\) app. With TuioPad, devices employ the Tangible User Interface over OSC (TUIO) protocol (Kaltenbrunner et al. 2005) to transfer the interaction on the touchscreen to the interactive surface and can therefore interact with all TUIO-enabled applications. However, TuioPad only shows a grey area on which a user can interact and gives no visual clue where their touch is located on the remote surface.

Our setup introduced in (Lee et al. 2011) employs a similar approach to TuioPad that enables mobile devices with a touchscreen (like smartphones or interactive tablets) to interact with all kinds of applications on interactive surfaces with different resolutions and physical sizes. As compared to TuioPad, our system additionally provides an image of the interactive surface’s screen on the mobile device. It would not be sufficient to just scale down the big screen’s image to the mobile device’s screen size. For instance, if an interactive surface had a resolution of 30 dpi and a mobile device a resolution of 120 dpi, the UI would be four times smaller on the mobile device. Hence, details could not be recognizable anymore on the small screen and touch interaction with such small elements would be complicated. Therefore,

\(^1\) [http://code.google.com/p/tuiopad](http://code.google.com/p/tuiopad)
there is a need for a sophisticated rescaling. Our system employs the following fisheye view (Furnas 1986) like distortion. As illustrated in Figure 1 on the left, the grabbed image from the interactive surface is subdivided in nine zones. The focus zone in the center is called the region of interest (ROI) and stays undistorted. The eight other context zones are compressed to fit in the zones adjacent to the ROI. The ROI’s position can be manipulated by the user on the mobile device and can therefore be located anywhere on the screen of the interactive surface. The layout of the nine zones on the device’s screen stays fixed.

After the distortion, the image is transferred to the remote device over the network. The user interacts with the mobile device via touch. If users touch the display, the application calculates the touch position in relation to the distorted image and sends it via TUIO to the interactive surface. Therefore, all interactive surface applications based on TUIO are inherently compatible with our system. If users, however, touch the display with three fingers, they modify the ROI. Dragging the fingers on the screen changes the maximum and minimum position of the ROI. Performing a scale or pinch gesture with three fingers zooms in or out of the ROI. Figure 2 shows the application while being used by a user employing an iPad. The red rectangle depicts the ROI.

3 Implementation

We implemented our target device application as a native app for the iOS platform. The application that provides the communication between the interactive application on the surface and the iOS app is a native Mac OS X application running on the surface computer. To be able interacting via touch on the interactive surface and on the iOS device
simultaneously, we use the TUIO proxy application Throng\(^2\) that joins TUIO messages from different sources.

4 Conclusion

We have presented a fisheye view distortion technique that allows displaying the content of an interactive surface on a remote device. We grab the surface’s screen content and subdivide it in nine zones: an undistorted zone in the center (called ROI) and eight adjacent zones that are distorted to fit in the frame around the ROI. Our distortion technique allows controlling every existing TUIO application with a mobile device without the need to modify the interactive surface application.

Acknowledgements

This work was performed with the financial support by the U-City Master and Doctor Course Grant Program funded by the Korea Ministry of Land, Transport and Maritime Affairs.

Funded by the BMBF-FHProfUnt grant no. 17043X10 and the DAAD program »Partnerschaften mit Japan und Korea« grant no. 50740919.

Literaturverzeichnis


Kontaktinformationen

{ljk1201, hyuskim, jnk}@konkuk.ac.kr, {johannes.luderschmidt, ralf.doerner}@hs-rm.de

\(^2\) http://code.google.com/p/throng