Smartphone Localization on Interactive Surfaces Using the Built-in Camera

Tom Cuypers Yannick Francken Cedric Vanaken Frank Van Reeth Philippe Bekaert
Hasselt University - tUL - IBBT
Expertise Centre for Digital Media, Belgium
{firstname.lastname}@uhasselt.be

Abstract

We present a technique for retreiving the position and orientation of smartphones on an interactive surface using solely their built-in camera. Additional tracking hardware such as RFID tags or cameras therefore become redundant. A pattern is projected below each detected device, which is captured by its built-in camera. From this pattern, the smartphone can efficiently extract its individual location on the surface. We show that the involved image processing is performed in real-time on current smartphones.

1. Introduction

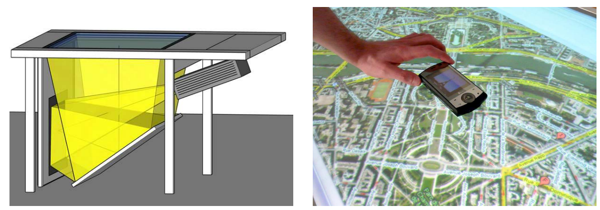
Nowadays, interactive surfaces such as multi-touch displays are becoming increasingly popular. These surfaces allow people to interact with computers in an intuitive way using multiple fingers or small electronic devices, often referred to as tangibles. Interesting examples of tangibles are smartphones, as they consist of a CPU, storage capacity, and Bluetooth/WiFi communication possibilities, opening numerous opportunities.

In order to allow useful means of interaction, the position of the tangible on the interactive surface needs to be known. A common way to achieve this is by arming the tangible with tracking hardware such as RFID tags [2]. Alternatively, one can place additional cameras above the surface to achieve vision based tracking [4].

Modern electronic devices feature a wide range of functionalities. For example, most mobile phones ship with a built-in camera and a wireless communication interface. In this work, the position as well as the orientation of the phone will be detected solely from what the built-in camera "sees", and is communicated to the interactive surface.

2. Smartphone Detection

Our multi-touch setup (Fig. 1(a)) consists of a projector, infrared (IR) camera, acrylic plate and two mirrors. The projector projects onto the backside of the acrylic plate via the two mirrors. Touchpoints on this plate are then detected by the IR camera, based on the Frustrated Total Internal Reflection technique described by Jeff Hann [1].



(a) Setup (b) Interactive Surface

Figure 1. (a) Schematic overview of our setup (b) Topview of the interactive surface

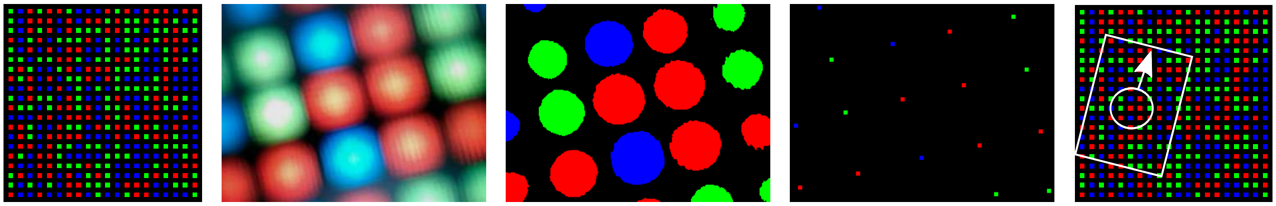
Not only finger touches, but also an initial estimate of the position of the tangible on the surface can be determined. IR intensities above a certain threshold are classified as touchpoints, while the other intensities are ignored to ensure noise elimination. Fig. 3 shows an image taken by the IR camera after a background subtraction is performed. The low intensity blobs are useful as an initial guess for the position of the tangible.

A color-encoded pattern [3] is then displayed on the surface at this position (Fig. 2(a)). This pattern consists of colored dots and is constructed in such a way that each block of 3×3 dots uniquely describes a position as well as an orientation. The dots are marked using the three base colors (red, green, blue), which makes them easy to identify and segment. Since the camera is placed very close to the projection surface, the imaged pattern will be out of focus. Due to the large relative distance between the base colors, segmentation is not affected by this issue.

Each smartphone needs to recover its own location and orientation and send it to the main application. This information is obtained in two steps. First a neighborhood of 3×3 dots needs to be recognized. Second, the neighboring dots are used to derive its location and orientation. This process is repeated to track the smartphone. When hovering over the display, we are able to robustly track the smartphone.

2.1. Dot Segmentation

The image captured with the built-in camera (Fig. 2(b)) is analyzed to retreive a 3×3 group of projected color dots. In order to distinguish between different dots, the input images are intensity thresholded and segmented (Fig. 2(c)).



(a) Pattern (b) Input (c) Color segmentation (d) Point detection (e) Localization

Figure 2. An overview of the point segmentation algorithm

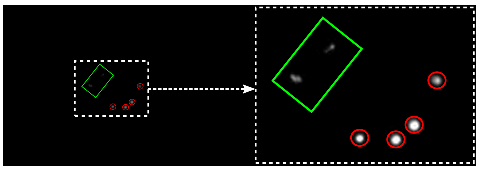


Figure 3. IR camera image after background subtraction. Red circles: touchpoints by fingers; Green rectangle: smartphone location

From the centers of the segments (Fig. 2(d)), a 3×3 grid of dots is extracted. The angle modulo 90 degrees between the grid and the captured image is used in the next step to retreive the orientation of the smartphone.

2.2. Retrieving Location and Orientation

The retreived squared grid of colored dots can be translated into a unique position and orientation within the predefined pattern. Without taking the dot colors into account, we are only able to determine the grid angle up to $k \times 90$ degrees. In a preprocessing step, we therefore build a tree that associates a window of nine colors to a specific position *as well as* a rotation factor $k \in \{0, 1, 2, 3\}$. These parameters are stored in the tree leafs at depth level nine (Fig. 4). The angle found in the previous step is finally added to the looked-up orientation to obtain the exact orientation.

3. Results and Discussion

In the experiments of this ongoing work, we used the HTC Touch Cruise Smartphone as it comes with a built-in camera and a Bluetooth and WiFi transceiver.

The segmentation and localization algorithm is implemented directly on the smartphone. It takes 23 ms to extract the dots in an 160×120 image. Notice that this resolution suffices because of the severe camera defocus. Although the segmentation is fast and robust, the camera currently needs to be held at least 5 mm above the display surface as deblurring is not yet integrated in our application.

Creating and using the look-up tree allows for fast localization. Related to the memory footprint, the tree of the full 20×20 dot pattern from Fig. 2(a) consists of 4340 nodes. The memory complexity of this datastructure is $O(n)$.

4. Future Work

In order to increase the robustness against camera defocus blur, we are currently looking into measuring the point

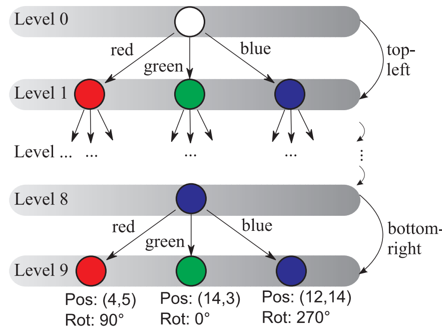


Figure 4. Look-up tree for efficiently recovering location and orientation of 3 by 3 neighbor dots.

spread function and deconvolving the input images. This can be done by projecting temporally shifting striped patterns below the camera [5] or spatially observing a single imaged dot. In the future we will extend our technique to allow 3D tracking of the smartphone when lifting/tilting it above the surface.

Acknowledgements

The authors (Cuypers Tom and Philippe Bekaert) acknowledge financial support by the European Commission (FP7 IP " 2020 3D media"). We gratefully express our gratitude to the European Regional Development Fund (ERDF), the Flemish Government and the Flemish Interdisciplinary institute for Broadband Technology (IBBT), which are funding part of the research at the Expertise Centre for Digital Media. We also would like to thank our colleagues for their time and effort in this work.

References

[1] J. Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST*, pages 115–118, 2005.

[2] A. Olwal and A. D. Wilson. Surfacefusion: unobtrusive tracking of everyday objects in tangible user interfaces. In *GI*, pages 235–242, 2008.

[3] V. Scholz and M. Magnor. Texture replacement of garments in monocular video sequences. In *EGSR*, pages 305–312, 2006.

[4] A. D. Wilson and R. Sarin. Bluetable: connecting wireless mobile devices on interactive surfaces using vision-based handshaking. In *GI*, pages 119–125, 2007.

[5] L. Zhang and S. Nayar. Projection defocus analysis for scene capture and image display. *SIGGRAPH*, pages 907–915, 2006.