FlashLight: Optical Communication between Mobile Phones and Interactive Tabletops

Tobias Hesselmann
OFFIS Institute for IT
Escherweg 2
26121 Oldenburg, GERMANY
tobias.hesselmann@offis.de

Niels Henze and Susanne Boll
University of Oldenburg
Escherweg 2
26121 Oldenburg, GERMANY
{niels.henze,susanne.boll}@uni-oldenburg.de

ABSTRACT
Mobile phones can be used as mediators between users and interactive tabletops in several scenarios, including authentication and the sharing of information. Existing radio-based methods such as WiFi or Bluetooth offer a high-speed communication channel, but have serious limitations regarding the tabletop-phone-human interaction. They are not able to locate mobile phones placed on the surface, often require fairly complex coupling procedures for establishing connections, and are potentially vulnerable to eavesdropping attacks. In this paper, we present a method for establishing a bidirectional communication channel between mobile phones and vision-based interactive surfaces utilizing the built-in flashlight and camera of mobile phones and the screen and camera of vision-based tabletops. We establish an entirely visual, secure and bidirectional communication channel at a speed superior to previous vision-based approaches, enabling users to establish connections and transfer data to and from interactive surfaces using ordinary out-of-the-box hardware.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Theory and Methods, Evaluation/methodology, Benchmarking, Interaction Styles

General terms: Performance, Design, Algorithms

Keywords: Data Transfer, Communication, Interactive Surfaces, Multitouch Tabletops, Mobile Phones

INTRODUCTION
With interactive tabletops and surfaces (ITS) becoming more and more ubiquitous, the exchange of data between such systems and other ubiquitous devices, e.g. mobile phones, becomes more and more relevant as well. In the recent years, mobile phones have become very smart devices, featuring applications such as media players, internet browsers, photo viewers and many more. It is an obvious next step to use mobile phones to interact with other smart devices, such as ITS, to overcome limitations of mobile devices and to enable more sophisticated interaction methods. Many of these interactions rely on a direct communication channel, which is established between ITS and mobile phones.

At first sight, this can easily be achieved using radio-based techniques: Bluetooth and WiFi are available in most current smartphones, and most interactive surfaces can easily be equipped with corresponding hardware as well. Nevertheless, radio-based methods have certain drawbacks. They usually require fairly complex coupling procedures and are thus unsuited when the communication channel needs to be quickly established, for example when using a mobile phone to authenticate to a system. Also, for combining the advantages of mobile phones with those of ITS, e.g., by exchanging photos or videos by placing several phones on an interactive tabletop and dragging contents from one phone to another, the surface computer needs to detect the position of mobile phones on its surface, and moreover, it needs to detect which Bluetooth ID or IP address belongs to a respective mobile phone. Obviously, this cannot be easily achieved using radio-based methods. In addition, radio transmissions can be vulnerable to eavesdropping attacks, raising privacy and security concerns. It is thus preferable to develop alternative communication mechanisms, and as many of today's ITS are relying on optical touch recognition techniques, such as Diffused Illumination (DI), Frustrated Total Internal Reflection (FTIR) or Diffused Surface Illumination (DSI), using optical communication methods offers a promising alternative, which has also been the approach of the following related works.

RELATED WORK
Wilson et al. use the built-in infrared (IR) port of a mobile phone to transmit an 8-bit ID to the camera of an interactive tabletop for establishing a Bluetooth communication channel between both systems [5]. As the authors point out, the main issue with this approach is the speed of the handshaking phase: The technique relies on the devices announcing themselves via Bluetooth, inducing an initial delay before the phones become visible to the tabletop system. Also, the used transfer protocol limits the communication bandwidth to an average 2.67 bits per second (bps), leaving room for improvement. Izadi et al. present an optical touchscreen solution based on a thin array of IR sensors and emitters [2]. They developed a technique to modulate the IR emitters of the display to transmit 3-bit codes from the touchscreen to an external IR sensitive device and mention that IR based

ITS'10, November 7-10, 2010, Saarbrücken, Germany.

communication from external devices to the display would also be feasible. Unfortunately, both approaches rely on IR light as communication medium, which is no longer feasible as most modern phones are not equipped with IRDA ports anymore. Another technique to transfer data from ITS to mobile phones has been developed by [3], who use the built-in camera of a smartphone to detect information displayed by a Microsoft Surface computer. Unfortunately, no details regarding speed, bandwidth or functionality of the approach are available at the time of writing. We can only suspect that it utilizes a pattern of changing colors to transfer data between tabletops and mobile phones, and thus may employ a technique similar to our approach. Echtler et al. use a shadow tracking device and Bluetooth distance measurement to identify mobile devices on the surface of an interactive tabletop [1]. The technique does not utilize an optical data transfer between mobile device and tabletop, but relies on the detection of rectangular shapes using computer vision algorithms.

APPROACH

The related work mostly relies on radio based methods for the actual data transfer from phone to tabletop or vice versa, or on the transmission of infrared light, which is not feasible using most current smartphones. In contrast, our approach relies on an entirely optical transmission of information, exploiting built-in features of both common tabletop systems and mobile phones. Using the built-in camera and flash of mobile phones, as well as the built-in camera and display of interactive tabletops, we establish an optical, bidirectional channel at a speed superior to previous vision-based approaches. We are also able to establish the channel in a fast handshaking phase. Radio-based methods, such as WiFi or Bluetooth, are not required for communication. Our approach works with common hardware and does not need additional equipment. It is also fairly secure against eavesdropping attacks: The phone is placed directly on the surface of the tabletop and only the area below the phone is used for communication. It should be mentioned though, that depending on the position of the phone's camera and flash, information may still leak out in the form of visible light when communicating. Nevertheless, in contrast to radio-based methods, such security lapses become instantly visible to users, enabling them to take appropriate countermeasures.

As a fully optical approach, our method opens up possibilities for communication in scenarios where radio-based traffic is unsuited or impossible, including

- User authentication in security critical settings, e.g. ATMs or access control systems.
- Areas where radio-based communication is prohibited, e.g. aircrafts or hospitals.
- Public areas, where many people tend to deactivate Bluetooth and WiFi for security reasons and to save battery power.

In the following, we will shortly describe the technical setup used and then explain the two techniques used to transmit data from mobile phones to the tabletop and vice versa.

Technical Setup

For development and evaluation purposes, we used an interactive tabletop developed at our institutes and a Google Nexus One smartphone (see Fig. 1). The tabletop system utilizes a firewire camera with a frame rate of 60 FPS at 640x480. Diffused Surface Illumination is used for detecting touches on the surface. The display consists of a projector and a rear projection screen. It has a viewable screen size of 70x112 cm (52") at a resolution of 1280x800 (16:10). The Nexus One features a 25 FPS camera and a flash built into the back of the phone. The flash features a high-brightness LED which can be triggered by software.

Figure 1: OFFIS Interactive Table and Google Nexus One smartphone

TABLETOP TO MOBILE PHONE

To transmit data from tabletop to mobile phone we utilize the camera built into virtually any smartphone. We assume that the tabletop can recognize that an object is placed on its surface. In our implementation, we use a simple computer vision algorithm to achieve this. To start the communication, the tabletop illuminates the area below the phone. As the phone's camera images are naturally very blurry when placed on the tabletop, visual markers or other textured images cannot be used to transmit information. Alternatively, we used constant colors filling the area below the phone's camera.

Color based encoding

Since dark colors reduce the camera's frame rate to approximately 10 FPS, due to automatic adjustments of white balance and shutter speed, they cannot reasonably be used for data transmission at a decent bandwidth. In addition bright colors, such as white, also distract the phone's camera. Due to automatic white balance adjustments, a pure white can e.g. become a shade of blue in the recorded image.

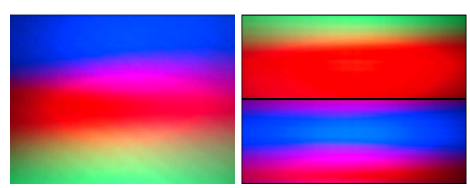

Figure 2: Single camera images (left) and two consecutive camera images (right) at 50Hz.

Nevertheless, the three extremes of the RGB color space (red, green and blue) can be differentiated very well and allow a robust classification of a pixel's color. As shown in Figure 2 (left), transitions from one primary color to another are very blurry, e.g. a color change from blue to red results in a smooth transition from blue via purple to red. Similar effects occur for every two primary RGB colors. Also, the colors are heavily affected by the camera's automatic white balance adjustments, e.g. purple images are hardly distinguishable from red or blue depending on the previous colors. To transmit information, we experimented with encodings using up to six colors, but it quickly became clear that due to high error rates at fairly low throughput (only up to 10 FPS), it was more reasonable to stick to the the three primary colors. The resulting encoding is described in the following.

Bit-level encoding using sub-frame resolution

According to Shannon's sampling theorem [4], information can not be transmitted error-free with more than half of the camera's frame rate (when using the average color of the entire camera image). With the camera's maximum sampling rate of 25 FPS, the maximum theoretical bandwidth without errors would result in a fairly slow 12.5 FPS. To increase the rate at which color changes can be detected, we exploit the characteristics of today's smartphones: A digital signal processor (DSP) continuously reads the color values from the camera sensor and copies them to the phone's memory. If the recorded scene changes from one color to another while the DSP reads from the sensor, the first part of the read image contains the first color and the second part of the image contains the second color. Thus, the resulting image does not contain a snapshot of the recorded scene at a discrete point in time but a continuous stream of the presented colors. At 25 FPS the gap between two consecutive camera images is shorter than 2ms or 2% of the camera image (see Figure 2, right). In order to achieve sub-frame resolution, the camera image is processed row-by-row using each row's average color. To increase the robustness to noise and to avoid false results at color transitions, a sliding window with four consecutive rows (3.33% of the entire image) is used. A color is only detected if all four rows in the window share the same color. Thus, it is possible to detect colors even if the area below the phone's camera changes its color two times within one frame.

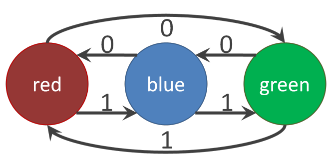

Figure 3: Bit encoding using color transitions.

As shown in Figure 2 the color edges are very blurry and the area covered by the transition between two colors can be larger than the actual colors at high FPS. Thus, the duration a color is visible to the phone is not fixed. As the clock pulse edges are blurry and the cycle length varies, we decided to encode bits not using the actual colors recognized, but based on the transitions between colors. The finite-state machine shown in Figure 3 shows the bit encoding used for our approach. For example, a transition from red to blue results in a '1', while a transition from green to blue results in a '0'.

Evaluation

To determine the speed and the error rates that can be achieved we conducted an evaluation using the technical setup described before. We implemented a test program on the tabletop that transmits bit sequences at different frequencies. A sequence containing 210 bits of white noise was used and we repeated the test 4 times for each cycle length. Our tests showed that the phone needs an initial synchronisation phase until the camera adapts its parameters to the tabletop, in particular regarding exposure time and white balance. At bitrates higher than 20 bps, approximately 100ms are needed. Thus, we discarded the first 10 bits during this phase in the test sequence. The bitrates between 20 and 66.67 FPS as well as the corresponding error rates are shown in Figure 4: Virtually no errors occur at speeds up to 33bps, but when increasing the transmission speed to up to 41.67 bps, the error rate slowly rises to up to 3.5 errors / 200bits, which may still be compensated using error correction mechanisms. When further increasing the transmission speed, the original bit sequence becomes virtually unrecognizable at more than 13 errors / 200 bits. To compare the tabletop results to a normal computer display, we also repeated the test using a MacBook Pro's LCD display, achieving slightly better results (see Figure 4).

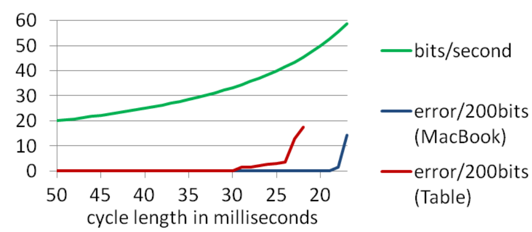

Figure 4: Speed and error rate, Tabletop to phone

Based on our results we conclude that, using our test setup, data can be robustly transmitted at rates of up to 33 bps. At this rate the color still spans at least 22% of the camera image and we assume that the phone is able to achieve even higher bitrates. Using a sliding window with 3.33% of the camera image and considering the gap between two consecutive images it might be possible to achieve bitrates up to 150 bps. With regards to the better results on a MacBook, we suspect that the limiting factor is not the phone's camera but the used display.

MOBILE PHONE TO TABLETOP

There are several options to send visual information from the mobile phone to the tabletop. One is to use the display of the phone to send information, similar to the approach presented in the tabletop-to-phone direction. This technique was also explored by [5]. As they point out, one particular disadvantage of that approach is that the transmission application temporarily needs to take control of the display. Moreover, most mobile phones do not feature a front camera, rendering bidirectional optical communication impossible.

In contrast, we trigger the flash built into most current smartphones to transmit information optically to the camera of the tabletop. The advantage of this approach is that it does not require any custom hardware and theoretically features a high data rate, as most modern camera flashes use LEDs, which can typically blink at a fairly high frequency. Naturally, flash lights are also very bright, and are thus easily recognizable by most cameras. In our case, the tabletop camera could easily recognize the flashes of the mobile phone, even with an infrared band-pass filter attached to the lens of the camera, which is a typical setup in most vision-based tabletop systems. To detect the blinking of the phone's flash, we processed the video stream of the tabletop camera to filter out points with a high level of brightness, which worked very robustly in our setup.

Bit level encoding

The flash of a mobile phone can typically take one of two states: On and off. Thus, it is necessary to use a binary code to transmit bits over the optical channel. In [5], pulse-width modulation (PWM, see top of Figure 5) was used to encode '0's and '1's using different lengths of LED pulses, i.e. 150ms for a '0' and 300ms for a '1', with 150ms breaks in-between pulses. The benefit of PWM is that it is quite robust and resistant against timing problems. Nevertheless, it also is fairly time consuming and wastes precious transmission time in contrast to other approaches.

We used a non-return-to-zero (NRZ) line code in our approach, in which high and low bits are represented by turning the flash on and off within a defined time frame (see Figure 5, bottom). Using NRZ, even the non-lit flash carries information, i.e. it is interpreted as a '0'. Also, there are no breaks between separate bits. As an example (see Figure 5), using the PWM approach of [5], it takes 2850ms to transmit 8 bits of changing '0's and '1's. Assuming 150ms pulse lengths, NRZ is able to transmit 19 bits in the same time, which is more than twice as fast. In addition, PWM transmission times get worse with the amount of '1's transmitted, while the transmission time stays constant using NRZ.

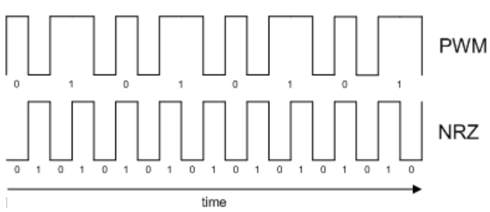

Figure 5: Pulse Width Modulation (PWM) vs. Non-Return-To-Zero (NRZ) bit encoding

Evaluation

To test the throughput and stability of the flash-based communication, we evaluated the system in situ and implemented a small test application on the phone, which transmits data to the tabletop using the flashing technique. Every test run was repeated at least four times. We configured the initial length of one bit to 50ms and then gradually increased the communication speed, measuring data throughput and error rate. Figure 6 shows the results of the evaluation. The maximum throughput without errors could be achieved at 48ms cycles, resulting in a transmission speed of approx. 20 bps. At 47ms and 46ms, the error rate was still fairly low, with 0.4 and 0.8 errors per 256 bits at a speed of approx. 21 bps. Starting at 45ms cycles (corresponding to 22 bps), the connection started to seriously degenerate, with a high error rate of 28 errors per 256 bits.

According to [4], at a camera sampling rate of 60 FPS, the maximum error-free transmission rate is 30 bps. Nevertheless, under realistic conditions, the sampling rate of the camera is often not constant, leading to lower bandwidth and communication errors. Again, factors such as automatic adjustments of white balance, exposure or sensitivity can destabilize the frame rate, as well as CPU load, thread scheduling and the software processing of camera images to detect flashes in the video stream. Nevertheless, we can conclude that even without further optimizations, we were able to achieve a virtually error-free transmission speed of two thirds of the theoretical maximum.

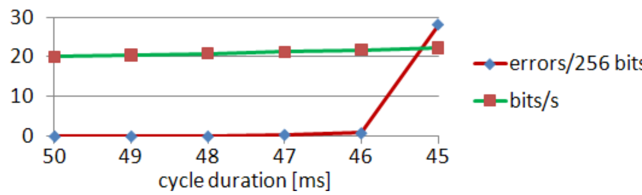

Figure 6: Speed and error rate, Phone to Tabletop

CONCLUSION

We have developed a method to enable bidirectional data transfer between mobile phones and vision-based tabletops, which works with common hardware setups and is solely based on optical transmission techniques. In contrast to previous approaches, we were able to increase the communication bandwidth to 30 bps from tabletop to mobile phone, and up to 20 bps in the reverse direction. As our technique also does not rely on a radio-based reverse channel (e. g. Bluetooth or WiFi), such technologies can be completely omitted in scenarios which do not require higher data rates. The technique is also fairly secure, as the communication between tabletop and mobile phone cannot be hijacked without being visible to the user. Enabling an entirely vision-based transfer of data between the involved devices, our technique opens a wide range of possibilities for interaction using mobile phones and interactive surfaces.

ACKNOWLEDGEMENTS

This paper is supported by the European Commission within the InterMedia project (038419).

REFERENCES

1. F. Echtler and G. Klinker. Tracking mobile phones on interactive tabletops. In *GI Jahrestagung (1)*, pages 285–290, 2008.
2. Shahram Izadi, Steve Hodges, Alex Butler, Darren West, Alban Rrustemi, Mike Molloy, and William Buxton. Thinsight: a thin form-factor interactive surface technology. *Commun. ACM*, 52(12):90–98, 2009.
3. Microsoft. Mobile connect sample application. http://tech net.microsoft.com/en-us/library/ee692087(Surface.10).aspx, 2010.
4. C. E. Shannon. Communication in the presence of noise. *Proc. Institute of Radio Engineers*, 37(1):10–21, 1949.
5. A. D. Wilson and R. Sarin. Bluetable: connecting wireless mobile devices on interactive surfaces using vision-based handshaking. In *Proc. GI '07*, pages 119–125, 2007.