Shadow Tracking on Multi-Touch Tables

Florian Echtler Manuel Huber Gudrun Klinker, PhD
{echtler,huberma,klinker}@in.tum.de

Technische Universität München - Institut für Informatik
Boltzmannstr. 3, D-85747 Garching, Germany

ABSTRACT

Multi-touch interfaces have been a focus of research in recent years, resulting in development of various innovative UI concepts. Support for existing WIMP interfaces, however, should not be overlooked. Although several approaches exist, there is still room for improvement, particularly regarding implementation of the ”hover” state, commonly used in mouse-based interfaces.
In this paper, we present a multi-touch system which is designed to address this problem. A multi-touch table based on FTIR (frustrated total internal reflection) is extended with a ceiling-mounted light source to create shadows of hands and arms. By tracking these shadows with the rear-mounted camera which is already present in the FTIR setup, users can control multiple cursors without touching the table and trigger a ”click” event by tapping the surface with any finger of the corresponding hand.
An informal evaluation with 15 subjects found an improvement in accuracy when compared to an unaugmented touch screen.

Categories and Subject Descriptors

H.5.2 [**Information interfaces and presentation**]: User Interfaces—*Input devices and strategies (e.g., mouse, touchscreen)*

Keywords

direct-touch, mouse emulation, tabletop interfaces, shadow tracking, multi-touch, FTIR

1. INTRODUCTION

In recent years, multi-touch capable input systems have increasingly been a focus of research in the UI community. Most of these systems can be categorized as direct-touch input devices, allowing the user to manipulate data objects directly on the display. In contrast, however, a large majority of input devices in everyday use are still indirect-touch devices like laptop touchpads. Moreover, like the plain computer mouse from which they have evolved, they support only a single point of interaction. Owing to this omnipresence of mice and mouse-like devices, a great percentage of existing and emerging software is based on established UI paradigms like, e.g., WIMP interfaces.
Ideally, all software that is to be used on a multi-touch system should natively provide support for these new input possibilities. However, any current setup will likely run a mix of multitouch and legacy WIMP applications, e.g. a web browser. To provide easy access to this large existing software base, a multi-touch system should offer backwards compatibility to mouse-based applications, while at the same time providing the full range of input data to multitouch-enabled software.

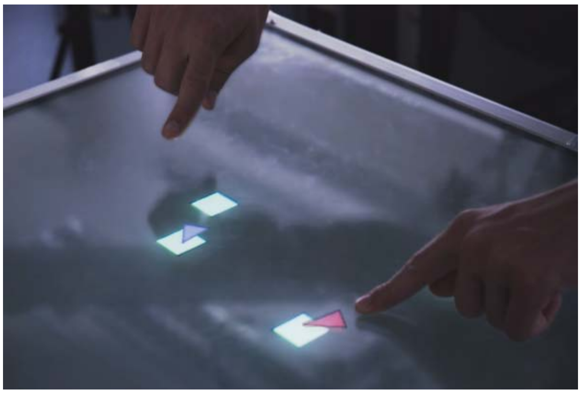

Figure 1: User controlling two cursors that are independent in position and orientation. The right hand’s cursor is in the ”hover” state (blue), while the left hand’s cursor was switched to ”click” state (red) by touching the surface.

One of the challenges posed by this goal is that most current multi-touch systems provide less data on certain aspects of the users’ actions than a mouse does. These systems usually report only one kind of interaction, a touch of the surface, which is generally interpreted to have the same function as a button click. With a mouse, on the other hand, it is possible to interact by only moving the pointer on top of an object without clicking, a technique known as hovering. Another problem is inherent to the concept of direct touch. When touching the surface, the finger itself typically occludes dozens of pixels. Particularly when aiming for small targets like, e.g., window handles, this greatly reduces the accuracy with which such an interface can be operated.

In this paper, we present a multi-touch system which is designed to address these problems. A multi-touch table based on FTIR (frustrated total internal reflection) is extended with an additional infrared light source mounted at the ceiling. This light source causes hands and arms to cast clearly defined shadows on the table surface. The rear-mounted infrared camera that is already available in the FTIR system can be used to track these shadows to provide proximity and orientation data for each hand. Using this data, the system can provide an independent pointer for each user’s hand that can be moved without touching the table. Subsequently, a ”click”

AVI ’08, 28-30 May , 2008, Napoli, Italy.

event can be triggered by touching the surface with any finger of the corresponding hand (see Figure 1). We have conducted an informal user evaluation with 15 subjects regarding the accuracy of our system and found it improved over an unaugmented touch screen.

2. RELATED WORK

The subject of how to accurately control a pointer on a direct-touch screen has already been investigated. In this section, we will look at existing approaches to enable direct-touch support for mouse-based applications, especially with respect to support for the fundamental UI states (”tracking/hover” and ”dragging/click”) noted by Buxton et al. [8].

Esenther et al. [9] present a solution which is based on the DiamondTouch [3] interaction surface and takes advantage of the multi-touch capability. Their *Fluid DTMouse* requires the user to touch the surface with two fingers simultaneously to control the mouse cursor, which is placed in the middle between the two contact points. Tapping with a third finger triggers a click. This technique allows to distinguish between hovering and clicking and also solves the occlusion problem. However, while effective, these methods might not be as intuitive to the first-time user as a simple ”point-and-tap” interface. For any kind of interaction, they require the users to touch the surface while moving their fingers, which may grow tiresome over time.

In a similar approach, Benko et al. [11] present several different dual-finger interaction techniques that allow the user to control the cursor with one hand while slowing down or freezing it with the second hand to enable accurate positioning. They also present a technique called *SimPress* which distinguishes between ”hover” and ”touch” states by analyzing the shape of the finger contact area. In an extensive user study, they found significant advantages over a plain touchscreen.

A different solution is offered by the SMARTBoard [1]. Their system provides a dedicated ”hover” button to switch into a state in which the user can move the pointer without triggering further interaction. Frequent mode switches are likely to be time-consuming, however, as they require the user to direct her or his attention to a completely different part of the UI.

Three other direct-touch systems which should be mentioned here are Wilson’s TouchLight [10] and PlayAnywhere [6] as well as Rekimoto’s SmartSkin [12]. Although all three systems provide proximity data about users’ hands, this feature has seen little use regarding pointer control. A possible reason is that they do not provide unambiguous differentiation between ”touch” and ”hover” states but instead have to rely on a threshold value. Users are therefore likely to trigger a click event before they actually touch the surface, which may be confusing.

Malik et al. [5] present a system which does not provide direct interaction, but offers a gesture-sensitive touchpad instead. Although it cannot deal with multiple users, the system offers two-handed input for a single user. It relies on a calibrated stereo camera, however, thereby significantly increasing the complexity of the system.

Han has presented an FTIR-based [7] direct-touch system which has contributed to the large research interest in multi-touch due to its easy construction from common off-the-shelf components, low price and back-projection support.

As our system is based on FTIR, we will include a more detailed description here. FTIR works by guiding infrared light through an acrylic glass sheet placed in front of a projection surface. The light rays are subject to total reflection at the air-material interface and are transported through the plate similar to an optical fiber. If a soft, dense material like skin touches the surface, however, total reflection is interrupted and the light illuminates the contacting object, which is now visible on the back side as a bright spot (see Figure 2). As mentioned by Han, this setup also does not provide proximity data.

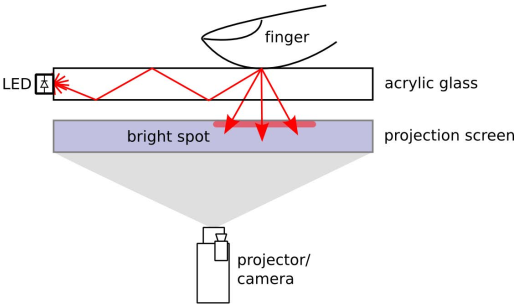

Figure 2: FTIR principle

3. TISCH[1] SYSTEM DESCRIPTION

We share the opinion that a multi-touch system can greatly benefit from supporting common mouse-based software. Our solution to the problems mentioned above is an extension to an FTIR-based interactive table that allows the user(s) to control an independent pointer with each hand. It is not necessary to keep contact with the surface for this kind of interaction, moving the hand above the table is sufficient. The surface only has to be touched if a click event is meant to be triggered.

The central element of our system is a multi-touch table (TISCH[1]) that provides room for 4 to 6 concurrent users.

A frosted glass plate of about 1.10 x 0.7 m is used as a back-projection surface. It is mounted on a robust aluminium frame which contains a projector, an infrared camera and a computer. An acrylic glass sheet placed on top of the projection area has 70 infrared LEDs attached around its rim to provide multi-touch input to the computer via an IR camera.

To gather proximity information, our goal was to create distinct shadows of objects on and above the surface. We have therefore mounted an additional infrared light source at the ceiling above the table. While this increases the complexity of the system and reduces its mobility, interactive tables tend to be stationary equipment which could be integrated into existing conference room tables or placed in a public area as an information booth. An additional overhead light should therefore be easy to add to such a setup.

As the illuminated areas of the surface should not interfere with the bright spots from the FTIR system, the top light source and the side-mounted LEDs are switched on alternatingly for odd- and even-numbered camera frames, thereby providing two consecutive images which will be referred to as shadow and contact image (see Figure 4). As this reduces the effective frame rate by a factor of two, a fast camera of at least 60 FPS should be used to provide a smooth user experience. To ensure accurate synchronization with the camera, a small circuit based on a PIC18 microcontroller activates the two light sources in turn and supplies them with a pulsed control current to increase total light output.

Building the top light source presented some unexpected challenges. To create hard shadows, a point light source is the ideal choice. We therefore evaluated a point light source first, consisting of a cluster of 16 infrared LEDs at different angles to be suspended over the center of the table. Unfortunately, this setup failed to illuminate more than a small fraction of the surface directly below the light source.

The reason for this effect is that light from above has to pass a total of four material-air interfaces (see also Figure 2). If each layer re-

[1] **T**angible **I**nteraction **S**urface for **C**ollaboration between **H**umans

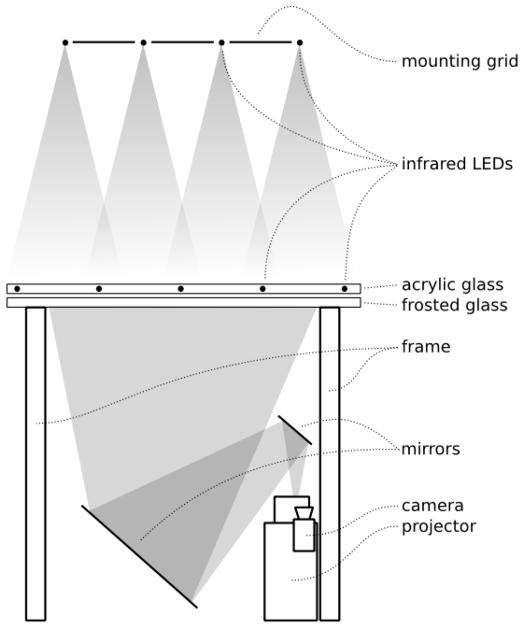

Figure 3: hardware setup

flects 15% of incoming light (a conservative assumption), the total intensity arriving at the camera already drops to $(1-0.15)^4 \approx 52\%$ of emitted light. The reflected percentage increases with decreasing angle of incidence according to Fresnel's equations. Below the critical angle of about 41°, the light transmission even drops to zero because total reflection takes effect and all light is captured in the topmost plate.

As a single point light source proved insufficient to illuminate the entire surface, we switched to a regular grid of 28 LEDs. Each LED is oriented straight downwards and mounted at a distance of 25 cm to the others. Considering the LEDs' beam width of about 20°, their intensity falloff and a distance of 1.50 m between surface and light source, the overlapping spotlights provide a sufficiently uniform illumination of the table with easily discernible shadows. For a schematic of the entire hardware setup, see Figure 3.

On the software side, we aimed for a clean separation between the input system and the end user application.

The raw camera images are acquired and analyzed in a background process. After a shadow/contact image pair has been read from the camera, both images are segmented into disjoint blobs by background subtraction, thresholding and erosion for noise removal. Size (pixel count), centroid, major and minor axes and outermost points along the major axis are now calculated for each remaining blob which is larger than the minimum size.

In a second step, illustrated in Figure 4, every touch point from the contact image is associated with its nearest shadow. Ambiguities between closely spaced touch points can now be resolved. After all data for the different blobs has been calculated, the positions are transformed by a homography [4] in a final step to compensate for the projective distortion between projector and camera image. This homography is calculated separately with a calibration tool using four point correspondences that are gathered by tapping four crosshairs in the screen corners. The transformed data for each blob is finally sent to the application(s) as a UDP packet for each processed image pair.

As a proof-of-concept and as a basis for the informal evaluation described in the next section, we use an application that displays an

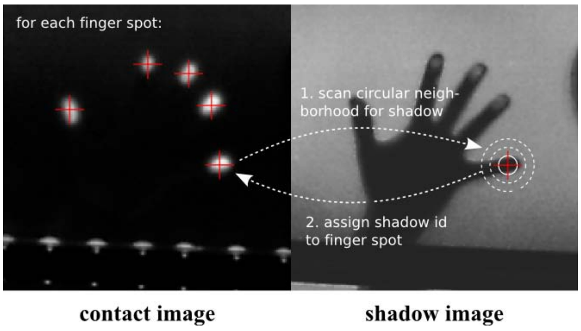
for each finger spot:
1. scan circular neighborhood for shadow
2. assign shadow id to finger spot
contact image
shadow image

Figure 4: shadow processing

arrow-shaped cursor for every detected shadow (see also Figure 1). This cursor is located near the peak of each shadow, but is shifted by an additional offset along the shadow's major axis. This prevents the user's hand from occluding the cursor. As long as the hand is not in contact with the table, the cursor is in the ”hover” state. If the surface is touched with any finger of the associated hand, a ”click” event is triggered at the location of the cursor tip. As an additional feedback to the user, the cursor changes color from blue to red. This state is maintained as long as one or more fingers from this specific hand touch the surface. When all fingers have been lifted off again, the cursor reverts to the ”hover” state. For a short video demonstrating use of our system, see [2].

4. PRELIMINARY EVALUATION

For a first evaluation of our setup, we conducted an informal targeting test. Our goal was to verify whether our system, despite its prototype state, would be able to provide increased targeting accuracy through pointer feedback as noted in other publications, e.g. by Benko et al. [11].

The targeting application requires users to activate a single randomly positioned square target on the screen. After activation, the target is replaced by a new one in a different randomly chosen location and the task is repeated. For each of the 20 repetitions, the time between two successful target activations and the distance from the target center to the touch point is recorded. The target has a size of 30 x 30 pixels, which is equivalent to a physical size of about 1 square inch on our screen.

Two different test modalities were used. In the first one, no cursors were available and the target had to be directly touched with any finger to activate it. The centroid of the contact spot was used as touch point.

In the second test, each hand held over the table surface was augmented with a cursor as described above. In this case, the touch point was at the cursor tip.

Our 15 computer-literate test subjects (4 women, 11 men, average age 26 years) had little prior experience with touch-screens. All users were told to hit the targets as fast and accurately (that is, close to the center) as possible, and in the second test, to use the hand-controlled cursor as they would use a mouse cursor.

The results of our test confirmed our expectations. Despite noticeable jitter in the cursor position, the accuracy increased by about 4 pixels: the average distance from the target center was 11.7 pixels for the first test, compared with 7.5 pixels for the second test. However, the test also exposed a drawback of our system: the time which users took to hit a target increased by a factor of two from 1.3 seconds in the first test to 2.6 seconds in the second test. Again,

this can be attributed to cursor jitter which caused users to hesitate before tapping the target.
A notable observation during our tests was that users intuitively took advantage of the cursor's variable orientation, particularly when reaching for targets close to the table edge. In this case, most subjects oriented their hand parallel to the table edge so that the cursor now pointed perpendicular to the subjects' viewing direction instead of outwards, thereby preventing occlusion.

5. CONCLUSION AND FUTURE WORK

Our system provides a useful addition to existing FTIR-based multi-touch setups in order to provide intuitive mouse emulation, including support for the "hover" state and precise targeting. Although the required hardware slightly increases in complexity (additional ceiling-mounted infrared light and control circuit), still only a single camera and calibration is required.

Although our evaluation was not rigidly controlled, some conclusions can be drawn. Our technique is able to provide a noticeable increase in pointing accuracy at the expense of targeting speed. Accuracy as well as usage speed are likely to increase further when the cursor motion jitter is reduced. To this end, a Kalman filter can be employed to make cursor position and orientation less sensitive to camera noise.

Moreover, the system is highly intuitive. Most users understood the system immediately after first holding their hand over the table surface and observing the associated cursor. This is an advantage that should not be underestimated, as interactive surfaces are often deployed in public or semi-public scenarios where little or no prior instruction is available to users.
As the overhead light source currently requires a (semi-)permanent installation, an application like a public information booth is one of the best suited scenarios for our system. Here, a mix of legacy software (e.g. a web browser) and multi-touch applications (e.g. casual games) is likely to be used. While conventional applications can then be controlled with one or more pointers as usual, multi-touch software can be used with direct touch interaction.
However, if a mobile solution is absolutely mandatory, the light source could be attached directly to the table like a canopy. While the unavoidable poles might hinder users, a different solution which relies entirely on environment light could be envisioned. In this scenario, an existing ceiling lamp could provide the necessary illumination.

Another aspect of this system is that it provides support for tangible user interfaces [13]. Objects on the surface create shadows, but no contact spots (with the exception of some very soft plastics). These shadows could be classified according to their size and ratio between major and minor axis, thereby providing, e.g., physical handles for widgets.

Finally, we have not yet explored the possibilities which are offered by our system's ability to assign contact spots to a certain hand. Recent laptop touchpads like those installed in MacBooks allow the user to tap with one, two or three fingers simultaneously to perform a left click, right click or scrolling operation. While other multi-touch surfaces would not be able to distinguish such a gesture from closely spaced gestures with two hands, our setup can easily be extended to support such interactions. This could also improve intuitive usability, as laptop users are probably already well accustomed to these gestures.

6. REFERENCES

[1] Smart Technologies. SMART Board. `http://www.smarttech.com/SmartBoard`.

[2] F. Echtler. Shadow tracking demonstration video. `http://campar.in.tum.de/personal/echtler/avi08-shadowtrack.avi`.

[3] P. Dietz and D. Leigh. DiamondTouch: a multi-user touch technology. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 219–226, New York, NY, USA, 2001. ACM Press.

[4] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[5] S. Malik and J. Laszlo. Visual touchpad: a two-handed gestural input device. In *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces*, pages 289–296, New York, NY, USA, 2004. ACM Press.

[6] A. Wilson. PlayAnywhere: a compact interactive tabletop projection-vision system. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 83–92, 2005.

[7] J. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118, New York, NY, USA, 2005. ACM Press.

[8] W. Buxton, R. Hill, and P. Rowley. Issues and techniques in touch-sensitive tablet input. In *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 215–224, New York, NY, USA, 1985. ACM.

[9] A. Esenther and K. Ryall. Fluid DTMouse: better mouse support for touch-based interactions. In *AVI '06: Proceedings of the working conference on Advanced visual interfaces*, pages 112–115, New York, NY, USA, 2006. ACM Press.

[10] A. Wilson. TouchLight: an imaging touch screen and display for gesture-based interaction. In *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces*, pages 69–76, New York, NY, USA, 2004. ACM Press.

[11] H. Benko, A. Wilson, and P. Baudisch. Precise selection techniques for multi-touch screens. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1263–1272, New York, NY, USA, 2006. ACM Press.

[12] J. Rekimoto. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 113–120, New York, NY, USA, 2002. ACM Press.

[13] H. Ishii and B. Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *CHI '97: Proceedings of the Conference on Human Factors in Computing Systems*, pages 234–241, 1997.