Experiences with Building a Thin Form-factor Touch and Tangible Tabletop

Shahram Izadi, Alex Butler, Steve Hodges, Darren West, Malcolm Hall, Bill Buxton, Mike Molloy
Microsoft Research Cambridge
{shahrami, dab, shodges, bibuxton, v-mimoll}@microsoft.com

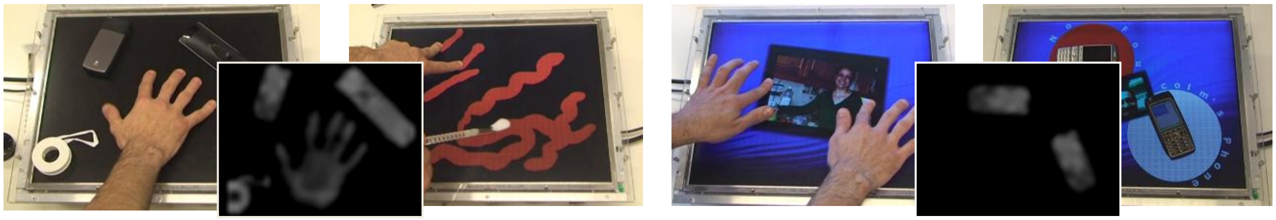

Figure 1: The ThinSight tabletop prototype, which supports detection of fingers, hands and other objects through the display. An example paint and photo manipulation application are shown in the centre. The display can also sense and communicate with active objects such as cell phones, shown right.

Abstract

In this paper we describe extensions to our work on ThinSight, necessary to scale the system to larger tabletop displays. The technique integrates optical sensors into existing off-the-shelf LCDs with minimal impact on the physical form of the display. This allows thin form-factor sensing that goes beyond the capabilities of existing multi-touch techniques, such as capacitive or resistive approaches. Specifically, the technique not only senses multiple fingertips, but outlines of whole hands and other passive tangible objects placed on the surface. It can also support sensing and communication with devices that carry embedded computation such as a mobile phone or an active stylus. We explore some of these possibilities in this paper. Scaling up the implementation to a tabletop has been non-trivial, and has resulted in modifications to the LCD architecture beyond our earlier work. We also discuss these in this paper, to allow others to make practical use of ThinSight.

1. Introduction

Tabletops offer interesting possibilities beyond traditional desktop computers. Such systems greatly benefit from multi-touch input, allowing users to interact with the surface using direct manipulation gestures [2, 5, 6]. Tabletops also naturally afford the placement of physical objects on the display surface, opening up the possibility for tangible user interfaces [4, 5]. This need for sensing touch and physical objects has motivated research into various hardware platforms (see [1] for an overview). Camera-based tabletop systems have emerged as one of the few technologies that can detect a wide variety of objects placed on the surface beyond just fingertips [4, 5, 7]. However, such systems have typically required a large optical path in front of or behind the display which resulted in a large profile – something that negatively impacts their use in many real-world scenarios.

In [3] we presented a novel technique for supporting optical sensing through thin form-factor LCDs making multi-touch and tangible interfaces more practical for deployment. We provided details of the hardware architecture and presented an early proof-of-concept prototype that supported sensing over a small portion of a vertically mounted laptop display. In this paper we focus on a second generation prototype, an interactive tabletop, which provides greater sensing fidelity across the entire display, with much improved image quality. Scaling up the implementation in this way has been non-trivial, and has resulted in modifications to the LCD architecture beyond what is described in [3]. We discuss these in order to allow practitioners to make practical use of ThinSight.

We go on to explore the capabilities of such ‘sensing displays’. Specifically, such displays have the ability to go beyond just touch, to enable tangible interaction through detection of other objects on the display surface. Our optical approach supports object detection by shape heuristics, by visual marker and by digital communication, but at varying levels of robustness. We present motivating examples of these disparate detection schemes and discuss the interplay between them. While such techniques could be supported by other approaches, with ThinSight they are supported in a much thinner form-factor making interactive tabletops more viable for real-world use.

2. Extensions to the ThinSight hardware

The proof-of-concept prototype which was reported in [3] uses ThinSight sensor boards to cover a small area in the centre of a laptop LCD panel. We tiled a larger array of these PCBs (30 in total) behind a standalone 21” desktop display panel for our tabletop prototype as shown in Figure 2. Besides the increased size of the display, other benefits are improved viewing angle, brightness and contrast.

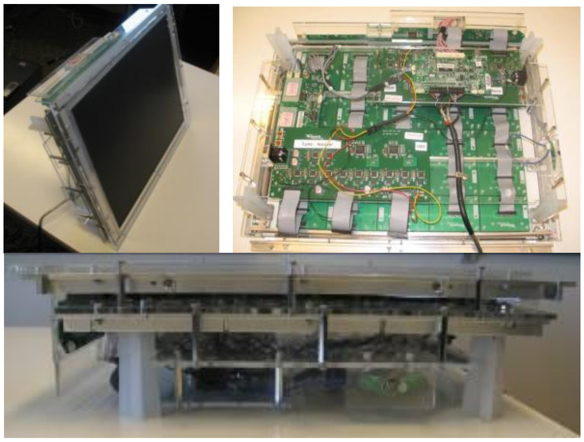

Figure 2: The ThinSight hardware as viewed from the sides and behind. 30 PCBs (in a 5x6 grid) are tiled with columns interconnected with ribbon cable and attached to a hub board for aggregating data and inter-tile communication.

With the increased number of PCBs, an extension to the original hardware architecture is needed to aggregate the data from each tile and use a single channel to transmit the full sensor image to the PC. This is done in our implementation using a custom ‘hub’ PCB based on an FPGA. The hub is also used to synchronize the tiles, allowing rows on each tile to be scanned in lock-step. This is crucial in limiting optical interference between adjacent PCBs.

One main issue we encountered when moving to the larger display concerned the stack of materials used within the LCD panel. The combination of the diffuser and the brightness enhancing film (BEF) in the desktop panel (see Figure 3) causes excessive attenuation of the Infrared (IR) signal – something we did not observe with our earlier work using laptop LCDs. Removing these materials solves the problem, but in turn degrades the displayed image significantly. Firstly, removing the BEF reduces brightness and contrast of the displayed image unacceptably. Secondly, without a diffuser the image appears to ‘float’ in front of the backlight and at the same time the position of the IR emitters and detectors can be seen in the form of an array of faint dots across the entire display.

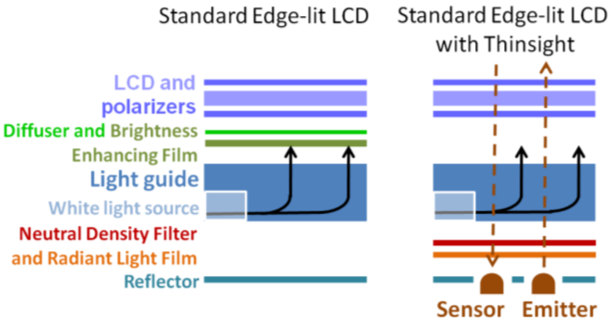

Figure 3: Typical LCD edge-lit architecture shown left (see [3] for a fuller description). The LCD comprises a stack of passive and active optical elements. White light sources are placed around the edge in the back of the panel. A white reflector and clear light guide help shine visible light towards the front of the panel. The films help scatter this light uniformly and enhance brightness. However these also cause excessive attenuation of the IR signal. In ThinSight shown right, these are replaced by two other suitable films, placed behind the light guide to minimize attenuation and also reduce noise caused by LCD flexing upon touch.

To solve these problems, we have experimented with many alternatives to the standard BEF and diffuser layers – including tracing paper, acetate sheets coated in emulsion paint, spray-on frosting, thin sheets of white polythene and mylar. We have found most materials to be unsuitable either because of a lack of IR transparency or an inability to uniformly scatter visible light (both are necessary to obtain high quality sensor data whilst maintaining a bright, high contrast display that hides the ThinSight sensors).

Pure optical IR cold mirrors were found to work, but are fragile and expensive and we were unable to source parts large enough to cover the entire display, resulting in seams which were visible to the user. The most effective solution we have found to date is the use of Radiant Light Film by 3M (part number CM500), which largely lets IR light pass through whilst reflecting visible light without the disadvantages of a true cold mirror. This was combined with the use of a grade ‘0’ neutral density filter, a visually opaque but IR transparent diffuser, to even out the distribution of the backlight and at the same time prevent the 'floating' effect. The surface quality of this Radiant Light Film is critical since minor imperfections (e.g. wrinkles or bubbles) are highly visible to the user. Laminating it onto a thin PET ensures a smooth surface. One final modification to the LCD architecture was to deploy these films behind the light guide. This minimized attenuation further, and reduced the noise due to the films deforming if the LCD flexes upon user touch.

As shown in Figures 1 and 4, the imaging capabilities of the tabletop surface are extremely compelling. The increased sensor area allows us to get a much better feel of fidelity of sensing and hence the practicality of the approach than was possible with our laptop-based proof of concept. Fingers, hands and other objects on the surface are clearly identifiable.

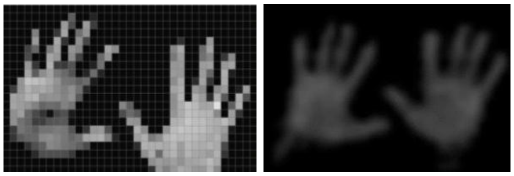

Figure 4: The raw ThinSight sensor data shown left and after interpolation and smoothing right.

3. Object detection

Physical objects are an important aspect of tangible tabletops, yet cannot be readily detected using capacitive or resistive sensing technologies. Optical sensing, on the other hand, readily supports the detection of objects beyond fingertips – essentially they can simply be observed through the display. To demonstrate some of the capabilities that result, we have built an example paint application that allows users to paint digitally directly on the surface using both fingertips and real paint brushes (see Figure 5). The latter works because ThinSight can detect the brushes' white bristles (which reflect IR). There is no need to distinguish finger from brush – simple detection of *object presence* is sufficient in this case because all objects can be treated as 'paint input'.

The paint application also supports a more sophisticated scenario where a transparent physical object is placed on the display surface, under which a palette of colors is rendered. The user can change color by 'dipping' either a fingertip or a brush into the appropriate well in the palette. We now need to distinguish between the different *object class* – the finger or brush tip and the palette. In this example, by simply looking at the shape of the connected components in the sensor image, we can distinguish palette from paint input – the latter is visible as a smaller ellipsoid. In practice there may be many scenarios where this course type of object class detection based on shape heuristics can be useful, and it is one that ThinSight supports fairly robustly.

Figure 5: Tangible input on ThinSight. Top: a paint brush, a physical palette, a visually tagged dial (sensor image far right). Bottom: An active IR tag detected and controlled by touching the digital ThinSight surface.

In some instances, however, it might not be possible to disambiguate physical objects in this way. For example consider a physical dial which can be used in the paint application to select different 'tools', e.g. to choose between eraser and paint modes, as shown in Figure 5. In this case, using shape may not be sufficient alone because the object is smaller and hence has a chance of being mistaken for a brush or finger. Also, the shape may not give accurate orientation information. To overcome these issues, we can *tag* the underside of the object i.e. augment it with a simple *passive* visual marker to support detection. Tagging therefore helps disambiguate the detection of different object classes, and also provides more accurate position and orientation information.

If we want to further extend the paint application to support more than one physical dial – introducing a second one to control the intensity of the paint for example, then it is no longer sufficient to simply differentiate object class. In this case we need a way of detecting *object instance*. This requires unique visual markers for each instance to be identified e.g. as described in [4, 7]. When using visual tags, the restricted sensing resolution of the current ThinSight prototype clearly limits the ability to reliably detect many unique tags. We have therefore explored other options that play to the strengths of the system, leveraging active digital communication between an object and the display.

3.1 Active detection

One approach to detection of object instance is to replace the use of passive visual tags with an *active* *tag* – a cheap and small electronic device that can either be used stand-alone or can be attached to an object and which can emit an IR code to uniquely identify it or the associated object. This code, which also locates the position of the object on the surface, is readily detected by ThinSight. If an IR receiver is built into the tag, it becomes possible to send data back to the tag by controlling the individual IR emitters in ThinSight. In Figure 5, we show an example active tag in use with the tabletop. Here we can switch the LED on the tag on or off and control its color by sending IR commands from ThinSight to the tag. We can also uniquely identify the tag by modulating an IR signal from the tag to the display.

Another interesting possibility related to the active tags described above is the ability to detect certain existing electronic devices such as mobile phones on the surface without physical augmentation of the device. These could be considered as *active objects* that are untagged. We have developed a new object detection scheme for uniquely identifying a mobile device with an embedded camera and Bluetooth (BT). This works without these devices needing to be tagged in any way, only client software is required for the device. The approach works differently to [8] which leverages the IR port or display of the device.

The scheme (Figure 6), works by running a simple client application on each of the mobile devices that listens for BT requests. When one of the devices is placed on the surface, the object class is detected passively using simple shape heuristics. At this point the system knows the object class (mobile device) and location, but it does not yet know the object instance; this is required before any data can be sent between the device and the display. There may be a number of candidate devices within BT range, and it is necessary to determine the one that has been placed on the surface. To do this, the system first connects to each of the candidates (running our client application). A unique color is then flashed underneath each device detected on the display surface. Each of the clients triggers the onboard camera capturing an image in the device's field of view. A histogram of color distribution in the image is then returned to the ThinSight PC that initiated the request.

In practice this works well for many camera phones. The camera does not need to be in focus, but rather only needs to be facing down on the surface. If there is any ambiguity in detection, another color can be displayed under the mobile device to resolve this. Once a definite match has been found, the surface can use the Bluetooth MAC address to perform communication with the device in question e.g. to copy recently taken photographs to the surface.

Figure 6: Synchronizing an untagged mobile device with the ThinSight display using the device's camera and Bluetooth.

As we start to explore additional uses for ThinSight (beyond our simple demonstrators described here), we think it will be useful to keep in mind the basic differences between *passive* and *active* tags\objects and detection through *presence*, *class* and *instance*.

4. Conclusions

In this paper we have presented our experiences to date with using ThinSight for interactive tabletops. What is unique about this approach is that it supports thin form-factor tabletops with both multi-touch and object sensing capabilities. In this paper we have uncovered deeper aspects of our approach, in particular describing some of the subtleties of scaling the system to larger desktop LCDs, and highlighting the strengths and weaknesses for ThinSight to support diverse object detection schemes.

5. References

[1] Bill Buxton, 2007. http://www.billbuxton.com/multitouchOverview.html

[2] Paul Dietz and Darren Leigh. DiamondTouch: a multi-user touch technology. In ACM UIST 2001.

[3] Hodges, S. et al. ThinSight: Versatile Multi-touch Sensing for Thin Form Displays. In ACM UIST 2007.

[4] Sergi Jordà, Martin Kaltenbrunner, Günter Geiger, Ross Bencina, 2005. The reacTable. *International Computer Music Conference (ICMC2005)*, 2005.

[5] Microsoft Surface, http://www.surface.com

[6] Rekimoto, J., SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In ACM CHI 2002.

[7] Andrew D. Wilson, 2005. PlayAnywhere: A Compact Interactive Tabletop Projection-Vision System. In ACM UIST 2005.

[8] Andrew D. Wilson and Ramin Sarin., 2007. BlueTable: Synchronizing mobile devices with surfaces. In ACM GI 2007.