

Multi-touch display using FTIR and GPU processings

Matej Gruľa*

Faculty of Mathematics, Physics and Informatics
Comenius University

Pavol Fabo†

Department of Applied Informatics
Faculty of Mathematics, Physics and Informatics
Comenius University.

Roman Ďurikovič‡

Department of Applied Informatics
Faculty of Mathematics, Physics and Informatics
Comenius University

Abstract

In this paper we present, a hardware implementation of a multi-touch display using a FTIR method [Han 2005], while augmenting a software implementation with a parallel processing using Nvidia's CUDA API. We thoroughly describe the process of building a multi-touch display using customized LCD display with an infrared lighting using LEDs for touch point detection in an infrared light spectrum. We discuss the software implementation, as well, starting with an image preprocessing and processing using NVIDIA's CUDA API and an optical tracking using GLSL and demonstrate its capabilities on a virtual drum kit implemented in OpenGL. We conclude our work with the possibilities for a future work and research.

CR Categories: D.2.2 [Design tools and techniques]: User interfaces; H.5.2 [User Interfaces]: Input devices and strategies

Keywords: human computer interaction, touch interface, multi-touch display

1 Introduction

From the beginning of the human computer interaction research, researchers have investigated methods and interfaces for naturalness of interaction with computer systems. Nowadays this research has progressed in big steps towards the systems for gesture interaction [Saffer 2008], eye gaze interaction [Morimoto and Mimica 2005], brain user interfaces [Nijholt et al. 2008], and especially tabletop multi-touch interfaces.

An overview of multi-touch systems was done by Bill Buxton [Buxton 2007], going all the way back to the 1970s when the Plato IV system, the first touch screen terminal had been introduced. This paper is a continuation of our previous work [Fabo et al. 2010]. We have improved the hardware implementation by making a stable metal-plate prototype, and enhanced the software part, by incorporating image processing and touch point detection by using GPU processing capabilities.

*e-mail: matgrula@gmail.com

†e-mail:pavol.fabo@fmph.uniba.sk

‡e-mail:durikovic@fmph.uniba.sk

The remaining of this paper is organized as follows. The next section sums up the latest research in the field of touch interaction. In the third section we discuss our hardware implementation describing materials and components used for building the multi-touch device. In the fourth section the software implementation is discussed, describing the dataflow starting with an image grabbed from a camera up to the output as a touch points used for generating Windows touch events. Finally, we conclude our work and discuss some possible future works.

2 Related work

Let us review some well demonstrated systems using multi-touch interaction. TouchLight system [Wilson 2004] enables touch-based interaction on a big vertical transparent screen, while using diffuse illumination for detecting touch points [Wilson 2004]. Hudson [Hudson 2004] finds out, that it's not required to use special diffuse illuminators. He describes the usage of LEDs as an optical sensor. His prototype of input/output device could be probably extended to the larger scale for using in touch surfaces.

Rear diffuse illumination techniques can be extended by sensing objects placed on top of the touch surface. Such systems can be implemented using special markers called fiducials, as firstly described in the reactIVision by [Jordà et al. 2006]. The extension of the idea of the fiducial markers was presented in [Antle et al. 2009], where the fiducial markers can be connected in the puzzle way, thus extending the interaction techniques. In [Kim et al. 2006], authors discuss the scheme for building touch tables, they describe the dataflow of standard algorithm for detection and tracking of the touch points. Finally [Kim et al. 2006] propose various gestures for touch based interaction with a virtual maps. In [Katz et al. 2007] authors omitted additional infrared illuminators from their hardware implementation, and proposed pure computer vision technique using four accurately placed cameras. Such a system is able to detect fingertips as well as touch points and gestures without any special set-ups. [Izadi et al. 2007] proposed optical based touch sensing in a thin form factor, enabling to build it inside the LCD screen. It is based on the combination of light emitter and an optically isolated light detector, called retro-reflective optosensor. The interaction with such a device is not limited just to the finger touches, it can detect any object reflecting infrared light. What's more objects with their own infrared emitter would be able to send data directly into the device. By masking out such infrared emitter on the object, it is possible to create a fiducial marker too. Furthermore, in [Izadi et al. 2008] a touch display is extended with an electronically switchable diffuser. Such a diffuser has two states, the first when it is transparent i.e. light can pass through, and the second when it is translucent i.e. light can pass through but only diffusely. This property is used in combination with two projectors with 60Hz refresh rate, allowing the resulting 120Hz. Projec-

tors are covered with special shutters, which in combination with switchable diffuser can produce two images on the two screens, one above the other. Another approach for using multi-touch inputs was presented by [Benko et al. 2008], where a traditional flat projection surface is replaced by a spherical surface, changing the output image space to the sphere. Using of such spherical output space is more appropriate for multiple user interaction than the traditional flat surfaces [Benko et al. 2008]. The ordinary projector has been changed to LCD screen, with higher resolution for the output images in [Motamedi 2008]. Furthermore they discuss the detection of several user interaction states like touch point detection and the event of grasping of an object with some fiducial marker. At [Rosenberg et al. 2009], the authors realized that the natural touch event has three properties. The first two are the location of the touch point and the third is the pressure of the touch event. The pressure was the main idea for building IMPAD touch device [Rosenberg et al. 2009], which is based on the force variable resistors, whose electrical resistance decreases when compressed allowing for the sensing of outside forces [Rosenberg et al. 2009].

3 Hardware implementation

Our implementation is based on FTIR technology and the configuration is inspired [Motamedi 2008]. We use a 1 cm thick plexiglass coupled with 27 inch Full HD LCD screen iiyama ProLite E2710HDS. For FTIR functionality the plexiglass is edge illuminated with two bars of 31 infrared LEDs generating infrared light with 850nm wavelengths. Bars are placed one on the top edge and the other on the bottom edge of the plexiglass. Under the plexiglass a customized LCD screen is placed. The customization is needed because a factory build LCDs are not transparent for infrared light. First of all, the LCD enclosing was removed. Some more LCDs layers have to be removed too. In our configuration we have removed a white reflective layer, a diffuse layer, and a Fresnel lens layer, thus leaving the LCD only with one diffuse layer and the LCD plexiglass with the LCD illumination. Our customization of the LCD is illustrated at Figure 1.

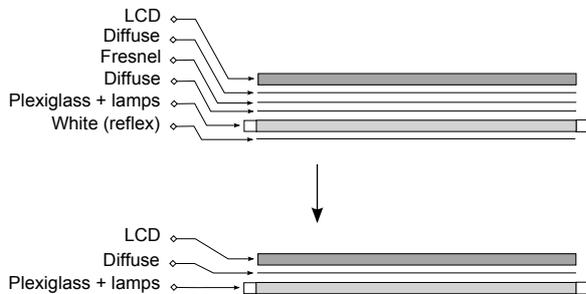


Figure 1: LCD customization. At the top, the factory built LCD, at the bottom, our customized LCD.

For capturing purposes we use a firewire camera from Pixelink which is able to capture images in higher resolution 1280 times 1014 and in 30 frames per second. This is enough for real-time interaction with touch interface since human visual processor has cycle time around 20 frames per seconds [Card et al. 1983]. Furthermore, as an illumination from LCD mixes up with the infrared illumination, we need to filter out the visible light spectrum, so the camera captures only the infrared light spectrum. This is done by using infrared band-pass filter allowing only light with wavelength between 700nm and 1000nm to pass through. Since we wanted to minimize the distance from the camera to the screen, the only way

to do so was using a wide angle lens, providing the acceptable ratio of lens distortion to the distance from the screen. The camera with wide angle lens and infrared band-filter configuration is illustrated at Figure 2.



Figure 2: Pixelink camera with wide angle lens and infrared band-pass filter attached.

Since Zaki Warfel [Warfel 2009] proposed using a paper for prototyping of physical devices, we initially used a cardboard prototype for our hardware implementation. When all raised problems were solved, we build a metal-plate prototype of our touch display. Both prototypes are illustrated at Figure 3.



Figure 3: Cardboard and metal-plate prototypes.

4 Software implementation

4.1 Calibration

The software implementation starts with camera calibration. Calibration is done using a standard calibration process described in [Bradski and Kaehler 2008] with a chessboard image at various orientations as an input image. An output is generated as a two camera matrices - intrinsic and distortion matrix. Process of generating intrinsic and distortion matrices are thoroughly described in [Bradski and Kaehler 2008].

Distortion matrix provides five parameters used to approximate radial and tangential distortion of camera lens. Radial distortion rises from the shape of the lens, because the light rays which are further from the center of the lens bent more than those closer in [Bradski and Kaehler 2008]. Fortunately, this distortion can be approximated by Taylor series, thus the radial correction can be mathematically defined by Equation 1. Where the k_1, k_2, k_3 are the distortion parameters from the distortion matrix created in calibration phase. r stands for the distance of the point from the lens center.

$$\begin{aligned}x_r &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\y_r &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6).\end{aligned}\quad (1)$$

Except of a radial distortion, there is a tangential distortion which raises from the manufacturing defects, which results in the non-parallelism of the lens. This distortion can be defined mathematically by Equation 2. Where p_1 and p_2 are parameters from the distortion matrix created in the calibration process.

$$\begin{aligned}x_{corrected} &= x_r + [2p_1 y_r + p_2 (r^2 + 2x_r^2)] \\y_{corrected} &= y_r + [p_1 (r^2 + 2y_r^2) + 2p_2 x_r].\end{aligned}\quad (2)$$

4.2 Image preprocessing

Since the camera image contains more information than the LCD screen itself, this phase is dedicated to finding the LCD screen inside the camera image. This phase is implemented purely in OpenCV, since it is done only once at the very beginning. First of all, using intrinsic and distortion matrices created in the calibration phase, we are able to remap the image, thus clearing out the distortion of the image. After the median filtering and the threshold operation, we run an OpenCV algorithm for finding contours. From the biggest contour in the image i.e. the LCD screen itself, we are able to find out the ratio of LCD screen to the LCD resolution in both x and y directions, which are used later for mapping touch points to the LCD resolution.

4.3 Touch detection process

4.3.1 Background subtraction

The first phase of touch detection process is a background subtraction. Background subtraction is used for basic segmentation of the touch points. Since it is possible in our application that the scene's background could change in time, for example because of different lighting conditions, we use a running average background subtraction. This method is better than simple subtraction mainly because, the model of the background is evolving in the time. Further information on averaging background subtraction can be found at [Bradski and Kaehler 2008]. The background subtraction can be parallelized using a CUDA API, so the pixel differences can be computed simultaneously.

The exact process is following:

1. Grab the image frame from the camera.
2. Convert the image to the grayscale image using the OpenCV library.
3. Copy the image to the graphic card.
4. Flip the image vertically¹.
5. Subtract the current background from the image.
6. Update the background model according to the averaging background subtraction method.

¹The image needs to be flipped in order to eliminate the fact that camera is sensing in the opposite direction we interact. Without the flipping the user's left hand will appear as a right hand in the camera image

4.3.2 Touch point detection

After the background subtraction the image is remapped in order to clean up the distortion, and allows the detection of the touch points. The touch point detection itself is done by scale-invariant feature transform implemented on GPU by [Wu 2007].

4.3.3 Registering touch events

Since Windows 7 has built in support for touch events, we used a [vMu 2010] driver based on Microsoft WDK to support touch events. vMulti creates a virtual device inside the operating system, vMulti library then connects to the virtual device and sends messages about the touch events. The current supported events are low level Windows 7 events: TOUCH_UP and TOUCH_DOWN.

4.4 MT-Drums - Demo application

We coupled our touch detection application with an OpenGL canvas to create a standard OpenGL application that is able to recognize and match the touch point to the object in the OpenGL scene. As a demonstration we created an application simulating a drum kit. When a touch event happens the drum in the OpenGL scene for a moment increase its size to visually stimulate that it has been hit. The photo of interaction with Multi-touch Drumkit is illustrated at Figure 4.

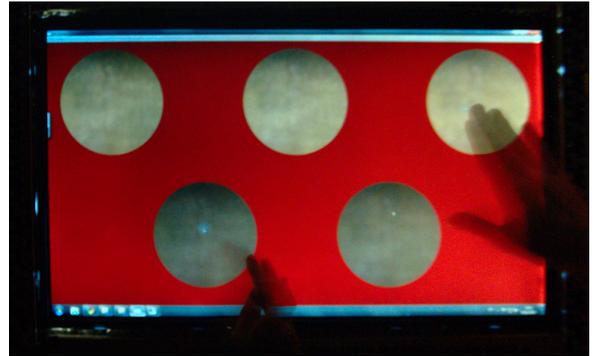


Figure 4: Interaction with MT-Drumkit application for simulating drums.

5 Conclusion and Future work

We described a basic background in building multi-touch tables, and various technologies for such devices. We presented our hardware configuration based on FTIR technology coupled with an ordinary LCD screen. Furthermore, we discuss the software parts needed for implementing touch device. We have discussed the problems in calibration, image preprocessing, background subtraction, touch point detection and registration of touch events to the operating system. To demonstrate the capability of the combination of the multi-touch tracking with the simple OpenGL scenes we have created the demo application simulating a drum kit.

As a future work, we will optimize the optical tracking, since the SIFT method, although done on GPU, is not that robust. Furthermore, we are going to support gesture recognition in order to used 2D gestures like "pinch" and zoom.

6 Acknowledgement

We would like to thanks to the Nadacia TatraBanky for funding this work, Jan Zizka for providing the camera and important advices about various cameras and camera lenses, and finally to the Faculty of Mathematics, Physics and Informatics, Comenius University, for providing the facilities and helping with the hardware implementation.

References

- ANTLE, A. N., MOTAMEDI, N., TANENBAUM, K., AND XIE, Z. L. 2009. The eventtable technique: distributed fiducial markers. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, ACM, New York, NY, USA, TEI '09, 307–313.
- BENKO, H., WILSON, A. D., AND BALAKRISHNAN, R. 2008. Sphere: multi-touch interactions on a spherical display. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, ACM, New York, NY, USA, UIST '08, 77–86.
- BRADSKI, D. G. R., AND KAEHLER, A. 2008. *Learning opencv, 1st edition*, first ed. O'Reilly Media, Inc.
- BUXTON, B., 2007. Multi-touch systems that i have known and loved.
- CARD, S. K., NEWELL, A., AND MORAN, T. P. 1983. *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA.
- FABO, P., ĐURIKOVIČ, R., AND NISHITA, T. 2010. Multi-touch display using the combination of ftir and hd lcd display. In *Proceedings of the 13th International Conference on Humans and Computers*, University of Aizu Press, 11–14.
- HAN, J. Y. 2005. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, ACM, New York, NY, USA, UIST '05, 115–118.
- HUDSON, S. E. 2004. Using light emitting diode arrays as touch-sensitive input and output devices. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, ACM, New York, NY, USA, UIST '04, 287–290.
- ISHII, H., AND ULLMER, B. 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, New York, NY, USA, CHI '97, 234–241.
- IZADI, S., HODGES, S., BUTLER, A., RRUSTEMI, A., AND BUXTON, B. 2007. Thinsight: integrated optical multi-touch sensing through thin form-factor displays. In *Proceedings of the 2007 workshop on Emerging displays technologies: images and beyond: the future of displays and interacton*, ACM, New York, NY, USA, EDT '07.
- IZADI, S., HODGES, S., TAYLOR, S., ROSENFELD, D., VILLAR, N., BUTLER, A., AND WESTHUES, J. 2008. Going beyond the display: a surface technology with an electronically switchable diffuser. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, ACM, New York, NY, USA, UIST '08, 269–278.
- JORDÀ, S., KALTENBRUNNER, M., GEIGER, G., AND ALONSO, M. 2006. The reactable: a tangible tabletop musical instrument and collaborative workbench. In *ACM SIGGRAPH 2006 Sketches*, ACM, New York, NY, USA, SIGGRAPH '06.
- KATZ, I., GABAYAN, K., AND AGHAJAN, H. 2007. A multi-touch surface using multiple cameras. In *Advanced Concepts for Intelligent Vision Systems*, J. Blanc-Talon, W. Philips, D. Popescu, and P. Scheunders, Eds., vol. 4678 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 97–108. 10.1007/978-3-540-74607-2_9.
- KIM, S.-G., KIM, J.-W., BAE, K.-T., AND LEE, C.-W. 2006. Multi-touch interaction for table-top display. In *Advances in Artificial Reality and Tele-Existence*, Z. Pan, A. Cheok, M. Haller, R. Lau, H. Saito, and R. Liang, Eds., vol. 4282 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 1273–1282. 10.1007/11941354_132.
- LOWE, D. 1999. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*.
- MORIMOTO, C. H., AND MIMICA, M. R. M. 2005. Eye gaze tracking techniques for interactive applications. *Comput. Vis. Image Underst.* 98 (April), 4–24.
- MOTAMEDI, N. 2008. Hd touch: multi-touch and object sensing on a high definition lcd tv. In *CHI '08 extended abstracts on Human factors in computing systems*, ACM, New York, NY, USA, CHI '08, 3069–3074.
- NIJHOLT, A., TAN, D., ALLISON, B., DEL R. MILAN, J., AND GRAIMANN, B. 2008. Brain-computer interfaces for hci and games. In *CHI '08 extended abstracts on Human factors in computing systems*, ACM, New York, NY, USA, CHI '08, 3925–3928.
- NUI, G. A. 2009. *Multitouch technologies*.
- ROSENBERG, I. D., GRAU, A., HENDEE, C., AWAD, N., AND PERLIN, K. 2009. Impad: an inexpensive multi-touchpressure acquisition device. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, ACM, New York, NY, USA, CHI '09, 3217–3222.
- SAFFER, D. 2008. *Designing Gestural Interfaces: Touchscreens and Interactive Devices*. O'Reilly Media, Inc.
- TOMFELDE, C. M., Ed. 2010. *Tabletops - Horizontal Interactive Displays*. Springer.
2010. vmulti - virtual multiple hid driver.
- WARFEL, T. Z. 2009. *Prototyping: A practitioner's guide*. Rosenfeld Media.
- WIGDOR, D., FORLINES, C., BAUDISCH, P., BARNWELL, J., AND SHEN, C. 2007. Lucid touch: a see-through mobile device. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, ACM, New York, NY, USA, UIST '07, 269–278.
- WILSON, A. D. 2004. Touchlight: an imaging touch screen and display for gesture-based interaction. In *Proceedings of the 6th international conference on Multimodal interfaces*, ACM, New York, NY, USA, ICMI '04, 69–76.
- WU, C., 2007. Siftgpu - a gpu implementation of scale invariant feature transform.