Multi-touch display using FTIR and GPU processing

Matej Gruľa[1], Pavol Fabo[2], Roman Ďurikovič[2]

[1]Faculty of Mathematics, Physics and Informatics, Comenius University
[2]Department of Applied Informatics, Faculty of Mathematics, Physics and Informatics, Comenius University

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, eye gaze interaction, brain user interfaces, and especially tabletop multitouch interfaces.

Hardware implementation

Our implementation is based on FTIR technology and the configuration is inspired [3]. 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. 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.

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. The camera's lens is also covered with a infrared band-pass filter to filter out the visible spectrum of the light and leaving just the infrared spectrum. Further more, we have used a wide angle lens in order to decrease the distance from the camera to the screen significantly. Both band-pass filter and wide angle lens are illustrated in Figure 1. Using this described configuration we have build two prototypes. The first build from the cardboard and the second from the solid metal plates. Both prototypes are illustrated in Figure 2.

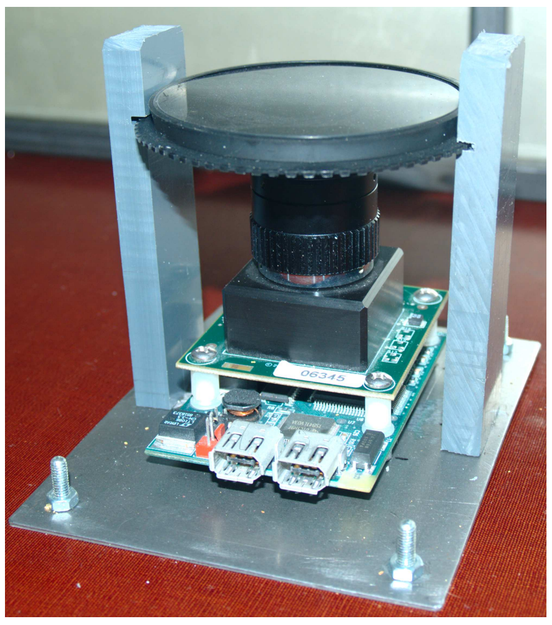

Figure 1: Pixelink camera with wide angle lens and infrared band-pass filter attached.

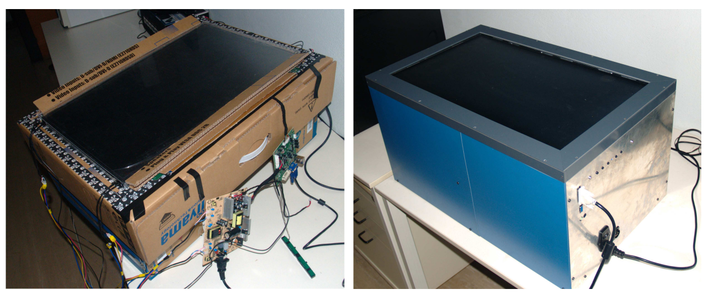

Figure 2: Cardboard and metal-plate prototypes.

Software implementation

The first phase of our implementation is calibration and image processing. The camera is calibrated using a chessboard, which allows us to remove radial and tangential distortion in the output image from the camera. After calibration a basic image preprocessing is done, where we use median filtering and threshold operation. On the binary image we run a contour finding method implemented in OpenCV in order to find the biggest contour i.e. the LCD screen.

The second phase is a touch detection algoritm, which actually uses the CUDA API. Firstly a background subtraction specifically the running average background subtraction method is run, in order to differenciate the background from the actual touch events. For touch point detection we use a scale-invariant feature transform implemented on GPU by [6].

Finally the touch events are registered with the operating system using a [1] driver based on Microsoft WDK. We currently support basic low level Windows 7 touch events: TOUCH_UP, TOUCH_DOWN, and TOUCH_MOVE.

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. The photo of the interaction with Multitouch Drumkit is illustrated at Figure 3.

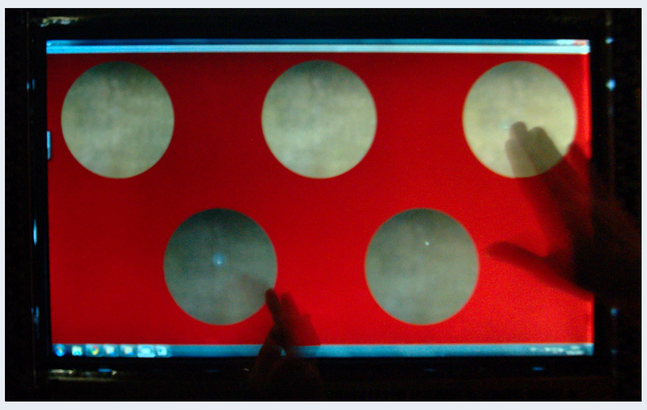

Figure 3: Interaction with MT-Drumkit application for simulating drums.

Acknowledgment

Our work wouldn't be possible without a kind help from Comenius University, and people in the Engineering lab. Jan Zizka has provided the camera used in the implementation, and Nadacia Tatrabanky has funded this project.

References

[1] vmulti - virtual multiple hid driver, 2010.

[2] N. Motamedi et al.
Hd touch: multi-touch and object sensing on a high definition lcd tv.
CHI '08, 2008.

[3] C. Wu.
Siftgpu - a gpu implementation of scale invariant feature transform, 2007.