Light-Keypad

Interaction through Coated Double Glazing

Lei Ye, Holger Schnädelbach, Steve North
Mixed Reality Laboratory, Department of Computer Science
University of Nottingham
Nottingham, UK
L.Ye@nottingham.ac.uk
Holger.Schnadelbach@nottingham.ac.uk
Steve.North@nottingham.ac.uk

***Abstract*—Interaction through glass has been demanding in industry, business, education, advertising and entertainment fields. Different technologies exist and work in varied circumstances. For example: acoustics or vibration detection, computer vision, capacitive interactive foils, infrared and others. The existing technologies tend to not work reliably through double-glazing and in outdoors conditions. This paper introduces an easy-to-implement, inexpensive solution, which is using light sensors to detect human actions and to interact with a computer. Light-Keypad can be used on any transparent medium, including through thick, coated double-glazing. This is a non-intrusive interactive method that may be used on existing objects and structures without modifying them. The prototype product has been used successfully in our current research project exploring the urban impact of digital screens.**

***Keywords*—Human-computer interaction; interaction; user interface; light sensor; keypad; urban screens; glass; double glazing; phidgets.**

I. INTRODUCTION

A. Urban Screens

Digital screens are becoming common in urban environments. Applications include signage and information kiosk solutions, advertising and more recently connected interactivity [1, 2]. Broadly speaking there are three types of installation for digital screens in public spaces: out of reach, standalone or behind glass. Most of the larger screens are placed high up and out of reach of direct interaction (i.e. touch) because of their size and their value. A large number of interactive screens are placed in specially designed and secure kiosks, weather proof when outside. A third type of screen is placed inside a commercial unit (i.e. a shop front) and behind glass. This placement offers protection from the weather and from vandalism but it does require an interaction mechanism that operates through glass.

In the course of our own research project considering interactive screens in urban spaces we are particularly interested in how networked media screens located in urban space can be designed to benefit public life, rather than merely transmit commercial content. The project has built a series of interfaces in urban neighborhoods in the UK, which use broadcast media and interactive technologies to enhance real world connections. It is a platform designed to encourage public participation and promote social cohesion by installing networked digital media screens in public areas.

To enable this, the placing of the screens required careful engagement with the communities that would be affected by that placement. For that reason alone, we approached specific venues (a public library, a cinema, a local community and an art gallery) and placed interactive screens constructed by us in their respective venues. To be able to study the urban impact of those interfaces, the deployed screens all face outwards and on to the street, requiring the aforementioned interaction through a glass pane.

B. Through-glass Interaction

At the majority of our research venues, off-the-shelve capacitance touch foils are working well [10]. These are all venues that have single glazing. However, at one of the venues, our screen is located behind 32mm-thick coated double glazing, which has a light green color. It is a Low ‘E’ glass (low emissivity glass, also named K-Glass), which is produced by introducing metal particles in the glass mix. This renders the capacitive touch foil inoperable at that location. With current building regulations, such types of shop front glazing and other types of coated glass are becoming more wide-spread and capacitive touch technologies will be difficult or impossible to use in many such circumstances.

We have considered the following alternative approaches to address this issue:

Detection of infrared, such as frustrated total internal reflection (FTIR) [3] and similar technologies [4], do not work in our case because we cannot get to the edge of the glass pane nor can install an IR camera behind the screen (our

The project was funded by the RCUK Digital Economy Programme (EP/I031413/1 & EP/I031839/1).

display is not projected). Modification of the existing building structure at our partner venues was not practical.

ZeroTouch [5] would offer high resolution touch detection but would have to be installed on the outside of the shop front, exposing a high-cost technology to the risk of vandalism and the weather.

We use FeONIC Whispering Window Speaker, which attaches at interior side of the window glass, as one of our audio interaction mechanisms. This sound transducers turn resonant surfaces, the shop window, into powerful vibration speakers. Its potential interferences and the issues of double glazing leave those technologies using acoustic vibration detections [6, 7] out of reach.

The approach of optical sensing [9], which may be integrated into a thin form factor display, due to its cost and complexity of installation, is also not suitable for our circumstances.

The infrared emitted by the Kinect [11] or similar motion trackers gets absorbed by some of the common glass coatings and the sensitivity and accuracy of Kinect drop significantly through the glass. Furthermore, direct sunlight will also blind the sensor.

The use of computer vision, e.g. OpenCV [12], to recognize gestures [8] has been proven technically difficult in challenging light conditions but would also be challenging to explain to an urban audience. Our screens are typically unattended but operated 24/7.

II. LIGHT-KEYPAD

As none of the existing technologies worked in our particular circumstances, we set out to address this with a different approach: Light-Keypad is built from a group of light sensors, which are placed behind glass facing out and an associated software driver. Principally it works with any transparent physical medium, as long as brightness changes in ambient light can be detected by the sensors. It does not matter how thick the glass is or what coating material in the glass. It works well under direct sunlight and dim illumination. And also, it may attach to existing shop window without having to adapt the building structure. Light-Keypad as reported here was iteratively developed responding to specific on-site requirements.

A. System Structure

Imagine each sensor is a key-button. By sampling instant brightness values from the light sensors, and comparing those to the average ambient brightness, the monitoring software determines which button is being pressed by a user, and a relevant interaction event will then be triggered on the screen, e.g. mouse moving up, moving left, mouse click, and etc.

Please refer to Figure 1 to see the system structure. Screen, computer and Light-Keypad are all located behind shop window. Users can only interact with the screen through the glass.

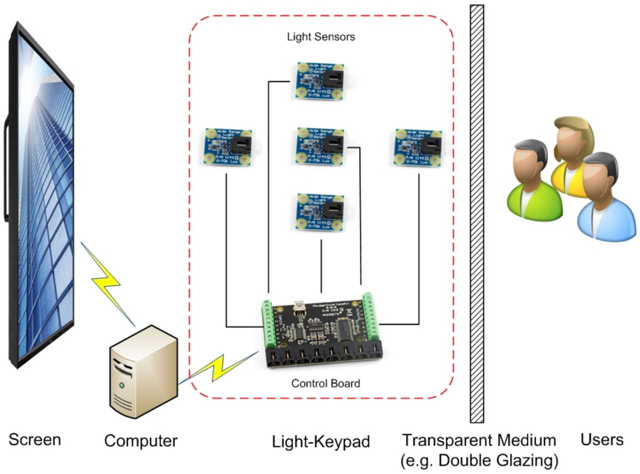


Figure 1: System structure

B. Phidgets Light Sensors

Phidgets are a set of user friendly building blocks for low cost USB sensing and control from PC [13]. The prototype of Light-Keypad is constructed from Phidgets hardware. Five light sensors are connected to a single Phidgets interface Kit 8/8/8 which connects to a Windows PC via USB. The sensor model is 1143, which can measure ambient light up to 70 kilolux (roughly equivalent to the brightness of direct sunlight). Its sensitivity to specific wavelengths of light is similar to human eyes. The sensor's output is logarithmic, so it will be more accurate at low light levels.

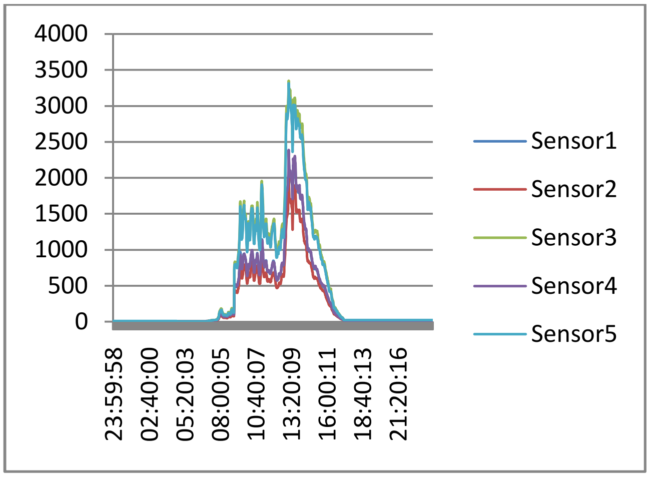


Figure 2: Brightness curves of 5 light sensors during 24 hours

(X: time line; Y: sensor value)

The formula to translate sensor reading into luminosity is:

$$Luminosity(lux) = e^{m*SensorValue+b}$$

where '*m*' and '*b*' are the calibrated values of each individual sensor. We conducted some experiments to become familiar with the response of the sensor. Attaching the sensors on the inner side of a thick double glazing and facing to street. Figure 2 is a chart that shows the sampled brightness value of the 5 light sensors during 24 hours. The shape of the curves is varied day-by-day depending on the weather conditions. We can see there is obvious individual differences among the sensors in spite of the calibrated values have been applied and these differences have to be taken account in any software developed.

The brightness curves on a particular sensor while it is being covered and uncovered are shown in Figure 3. The sensor value drops distinctly while an object is covering on the sensor from the outer side of the glass. This property can be used to simulate a key-press action, assuming a user put palm on the sensors trough the glass.

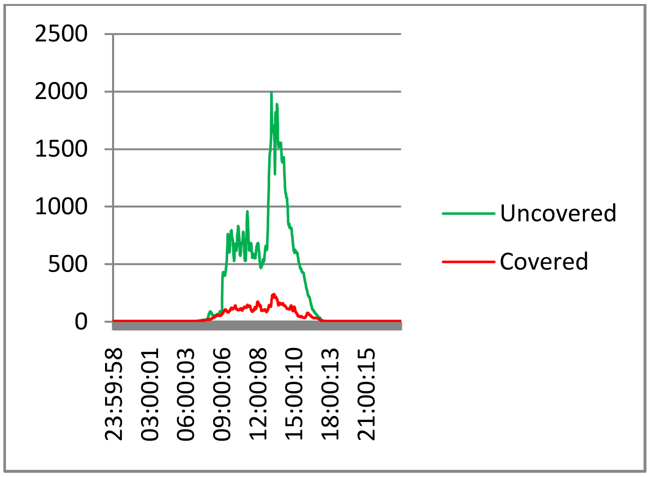


Figure 3: Brightness curves of 1 light sensor during 24 hours while covered and uncovered

C. *Software Development and Calibration*

The monitoring software is written in Python. It is using the Phidgets API to access the interface kit and the sensors. The monitoring software detects the brightness values changing on each sensor, determining which sensor has been covered. In our particular case, the developed software then operates like a mouse driver, allowing the movement of the mouse pointer on screen and the selection of on-screen buttons.

1) *Relative Sensor Brightness*

In order to reduce the effect, which is caused by the existing individual sensitivity differences of the sensors (please refer to Figure 2), a proportional value which indicates the relative brightness is used to decide whether a key is being pressed. When the relative brightness value is lower than a certain threshold, then the key is being pressed. Relative brightness for each sensor is calculated as follows:

$$Relative\ Brightness = \frac{Instant\ Brightness}{Average\ Ambient\ Brightness}$$

2) *Multi-covering Sensors*

During prototyping a second issue became evident that required additional calibration. In interaction, it frequently occurred that more than one sensor dropped their output values. This was caused by more than one sensor being covered or partially covered, for example covered by an elbow of the person interacting. Therefore it is possible that more than one sensors' value drop below the threshold during a sampling loop. In this case, a predefined priority table, which is designed for our particular situation, will be used to judge which key is exactly being pressed. Figure 4 illustrates that 3 keys are covered by user at same time. On the assumption that the brightness values of 'Up' and 'Right' are below the threshold. According to the priority table in right side, the 'Up' key, which is the 1st priority, will be considered as being pressed.

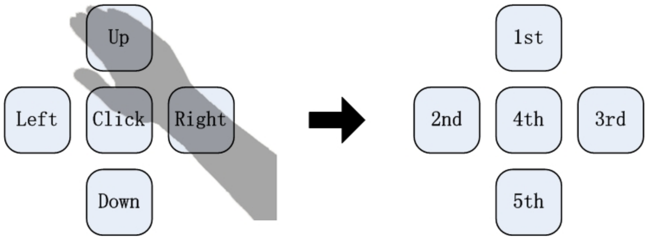


Figure 4: Priority table of light keypad, which is based on the analysis of users' postures while operating

A graphic interface was developed to visualize the relative brightness value for each sensor in real-time, where we can observe the multi-covering. From Figure 5 we can see the top key is being pressed, indicated by a red frame.

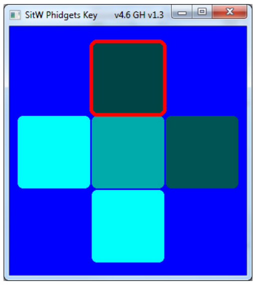


Figure 5: A screenshot of the relative brightness distribution map of the Light-Keypad at a certain moment

The threshold value is varied depending on the environment where the Light-Keypad is working, e.g. indoors or outdoors, daytime or nighttime. It can be easily figured out by onsite testing or by analyzing data log.

3) *Environmental Brightness*

Finally, average ambient brightness is the reference of calculating relative brightness. However, ambient light keeps on changing during a day. From morning to noon and from afternoon to midnight, the light brightness, which is also affected by weather conditions, is very different. For the stabilization of key-pressing recognition, an automatic calibration mechanism is used to decide the average ambient brightness.

By collecting serial sensor values with certain interval, ignoring those values that change significantly, an average brightness of ambient light can be determined by

$$Average\ Ambient\ Brightness = \frac{\sum_{i=1}^{n} Instant\ Brightness_i}{n}$$

Where n is the number of samples. Brightness values are stored in an FIFO queue. Only the latest n samples will be used for making a decision. There is a private FIFO queue for each sensor. Therefore the average ambient brightness of a certain sensor is absolutely independent. This method is used to eliminate the effects from the existing individual sensitivity difference of sensors.

The number of samples and the interval of sampling are configurable according to application requirements. In our project, n is 12 and sampling interval is 10 seconds. Lower number of samples and shorter sampling intervals will cause a quicker reaction to the changing of illumination conditions, but may also increase the chance of being interfered by accidental events.

D. *Hardware Installation*

For the installation on the inner side of double-glazing, sensors and interface are protected by plastic containers which were built using our 3D printer. These containers are attached on the glass by small sticky mounting pad. Figure 6 shows the 3D model of sensor container and a light sensor. Individual Light-Keypad can be mounted, relocated and removed easily, allowing for a very flexible and a non-intrusive installation.

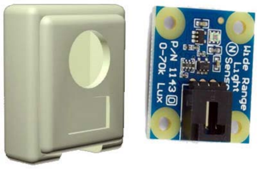


Figure 6: A sensor container and a light sensor 1143

Figure 7 shows the actual installed Light-Keypad at the venue, one of our project collaborators. The double glazing there is approximate 3 x 3 square meters in size, and is 32mm in thickness. This consists of: 12mm clear toughened outer pane, 14mm argon filled cavity with silver spacers and 6mm clear toughened inner pane hard coat low 'E'.

III. INTERACTION WITH LIGHT-KEYPAD

A. *Reflection on Technology*

Light-Keypad has now been installed for 12 months and it is working whenever the venue shutter is open. This is typically from 9am to 6pm Mon-Sat, and has therefore included periods of 'urban' darkness in the winter months.

Technically it works in the vast majority of environmental conditions. There are conditions, such as rapid changes in light from very bright to dark for example, where the environmental calibration cannot keep up. This results in erratic pointer behavior, while this occurs only for short periods of time.

B. *Reflection on Interaction*

From an interaction point of view, Light-Keypad works very differently from a touch screen. It is at an offset, i.e. it is not direct interaction. For that reason, interaction is slower. It requires end-users to make reference between their interaction on Light-Keypad and events on screen and this translation can cost time. Furthermore, due to the limited amount of sensors, the Light-Keypad can only deal with simple events.

C. *Experiences*

At the time of writing, Light-Keypad has been used with all of our more than 10 experiences, e.g. SoundShape, ScreenGram, MomentMachine, and etc. Just to give a flavor of the types of interaction it enables.

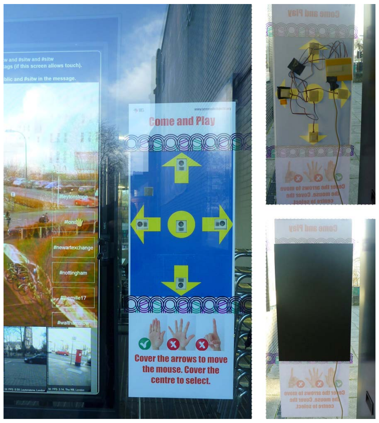

Come and Play
Cover the arrows to move the mouse. Cover the centre to select.

Figure 7: An actual example of installed Light-Keypad. Left is front view; up-right is internal back view and bottom-right is the back view with a cover

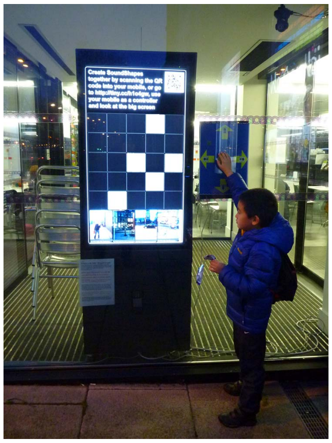
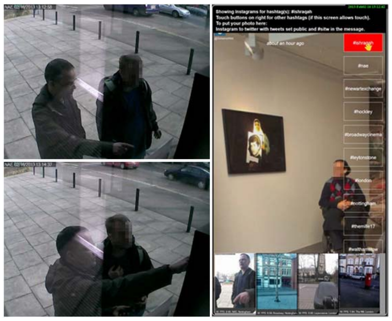


Figure 8: Experience SoundShape. Light-Keypad enables users to move the interface cursor to any of the 25 pads and select it. Interactions work well in the evening.

SoundShape (please refer to Figure 8) allows people across multiple sites to create visual and sound patterns collaboratively. Light-Keypad enables end-users to move the interface cursor to any of the 25 pads and select it.



Showing Instagrams for hashtag(s): #ishraqah
Touch buttons on right for other hashtags (if this screen allows touch).
To put your photo here:
Instagram to twitter with tweets set public and #sitw in the message.

Figure 9: Experience ScreenGram. Light-Keypad enables users to select onscreen tag buttons.

ScreenGram (please refer to Figure 9) allows people posting pictures to the screen or viewing pictures by using Light-Keypad to select different onscreen tag buttons.

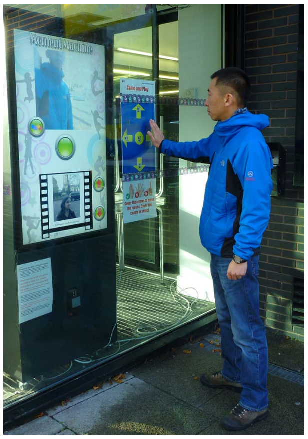


Figure 10: Experience MomentMachine. Light-Keypad enables users to take photos in front of the screen.

MomentMachine (please refer to Figure 10) allows people taking photos in front of the screen and sharing them onscreen with people at other venues. Based on a three months period statistics (from March to May), by using Light-Keypad, 198 photos were taken by users during 203 MomentMachine running hours, providing a basic indication that the interface can successfully be used by people on the street.

IV. CONCLUSION

The Light-Keypad is a lightweight response to through-glass interaction. It is relatively easy-to-implement, inexpensive, and flexible in configuration.

By monitoring a group of light sensors' instant brightness and average ambient brightness, intelligent software may implement the functions of a keypad. This solution offers an interaction method which works through any transparent physical medium, e.g. a thick coated double glazing, where some other technologies do not work. And it works properly under direct sunlight and dim illumination. It is a non-intrusive solution that can be used on existing objects and structures without modifying them.

This solution has been applied in our research project successfully.

ACKNOWLEDGMENT

The authors would like to thank Ava Fatah gen Schieck, Moritz Behrens, Wallis Motta and Efstathia Kostopoulou, who are our research partners at the Bartlett, University College London, UK. The authors would also like to thank Nemanja Memarovic, researcher and teaching assistant at the Faculty of Informatcis, University of Lugano, Switzerland, who is the main developer of MomentMachine.

REFERENCES

[1] Brignull, H., Izadi, S., Fitzpatrick, G., Rogers, Y., & Rodden, T. (2004, November). The introduction of a shared interactive surface into a communal space. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work* (pp. 49-58). ACM.

[2] Struppek, Mirjam. "The social potential of Urban Screens." *Visual Communication* 5.2 (2006): 173.

[3] Han, Jefferson Y. "Low-cost multi-touch sensing through frustrated total internal reflection." *Proceedings of the 18th annual ACM symposium on User interface software and technology*. ACM, 2005.

[4] Matsushita, Nobuyuki, and Jun Rekimoto. "HoloWall: designing a finger, hand, body, and object sensitive wall." *Proceedings of the 10th annual ACM symposium on User interface software and technology*. ACM, 1997.

[5] Moeller, Jonathan, and Andruid Kerne. "ZeroTouch: An optical multi-touch and free-air interaction architecture." *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*. ACM, 2012.

[6] Paradiso, Joseph A., et al. "The interactive window." *ACM SIGGRAPH 2002 conference abstracts and applications*. ACM, 2002.

[7] Crevoisier, Alain, and Cédric Bornand. "Transforming daily life objects into tactile interfaces." *Smart Sensing and Context*. Springer Berlin Heidelberg, 2008. 1-13.

[8] Wilson, Andrew D. "TouchLight: an imaging touch screen and display for gesture-based interaction." *Proceedings of the 6th international conference on Multimodal interfaces*. ACM, 2004.

[9] Hodges, Steve, et al. "ThinSight: versatile multi-touch sensing for thin form-factor displays." *Proceedings of the 20th annual ACM symposium on User interface software and technology.* ACM, 2007.

[10] Touch Foil used in current research project, http://www.visualplanet.biz/

[11] Microsoft Kinect, motion sensing input device, http://www.xbox.com/en-GB/kinect

[12] OpenCV, http://opencv.org/

[13] Phidgets Inc., http://www.phidgets.com/