Ubii: Towards Seamless Interaction between Digital and Physical Worlds

Zhanpeng Huang  Weikai Li  Pan Hui
HKUST-DT System and Media Laboratory
Hong Kong University of Science and Technology, Hong Kong, China
{soaroc, weikaili, panhui}@ust.hk

ABSTRACT
We present Ubii (Ubiquitous interface and interaction), an interface system that aims to expand people’s perception and interaction from the digital space to the physical world. The centralized user interface is broken into pieces woven in the domain environment. Augmented user interface is paired to the physical objects, where physical and digital presentations are displayed in the same context. The augmented interface and physical affordance respond as one control to provide seamless interaction. By connecting digital interface with physical objects, the system presents a nearby embodiment to afford users sense of awareness to interact with domain objects. Integrated on wearable devices as Google Glass, a less intrusive and more convenient interaction is afforded. Our research illustrates the great potential of direct mapping of interaction between digital interfaces and physical affordance by converging wearable devices and augmented reality (AR) technology.

Categories and Subject Descriptors
H.5.m. [Information Interfaces and Presentation (e.g. HCI)]: Augmented and Virtual Realities

General Terms
Design, Human Factor

Keywords
Physical affordance; augmented reality; wearable device; user interface; freehand interaction

1. INTRODUCTION
In the past thirty years, we have relied primarily on screen-based Graphical User Interface (GUI) to interact with digital world. No matter whether the screen is placed on a desk (e.g., PC screen), held on a hand (e.g., mobile phone screen), or displayed on a wall (e.g., projection screen), it has cultivated a predominantly visual paradigm of human-computer interaction [36]. However, traditional centralized GUI displays all information in the foreground. The interaction is separated from the physical world that we live in and interact with. It cannot leverage the rich senses and skills that we have acquired from the lifetime interaction with the physical environment [20].

Efforts have been focused on developing direct interaction between physical and digital worlds. As physical objects appeal to be more of our senses, they are redesigned as interaction metaphors. Architectural partition [20], physical props [15], mobile devices [6], and human body [13] are used to afford tangible interaction at a distance. However, physical objects as manipulation metaphor introduces intrusion to user interaction. Particularly, conventional user interface is designed to be irrelevant to the physical affordance of underlying objects. Users have to memorize manipulations like traditional GUI interaction.

As a new interface straddling the real and the digital, AR bridges the gaps between physical and virtual worlds [18]. It supports a sense of shared presence and communal social space. AR has been widely used to help users access additional information for a task without getting distracted...
from the task [16] [34]. Another baseline leverages AR to provide 3D menu interaction in current view of the real world [24] [30]. These works generate egocentric virtual user interface in front of current view, neither breaking of traditional centralized paradigm nor able to interact with physical objects.

Inspired by Weiser’s argument that computational services should be delivered through different devices whose design and location are tailored to support a particular task or set of tasks [38], we designed Ubii, a ubiquitous AR-based system to pair tailored interface to the physical affordance of objects. Ubii breaks traditional centralized user interface into individual pieces aligned with the physical objects. It is a directly mapped interface enabling users to perform in-situ interactions with the physical objects at a distance. As illustrated in Figure 1, users are able to manipulate physical objects including computers, projector screens, printers, and architecture partitions to complete tasks such as document copy, printing, sharing, and projection display. Individual AR menu overlays are aligned with physical objects to present their physical affordance. Developed on Google Glass, Ubii enables gesture-based freehand interaction with much less intrusion and more convenience.

2. RELATED WORKS

Ubii is built on several technologies including interaction at a distance, freehand interaction, and AR menu.

2.1 Interaction at a distance

Early systems such as PointRight [21] and Perspective Cursor [27] allow distant pointer motion across independently-driven display surfaces that are not organized in a single plane. The systems requires indirection of local input and complicated setups. Other works [39] [25] overcome the problems using absolute pointing techniques, but require additional pointing devices. In these methods motion errors are amplified as distance increases.

Mobile devices have been widely used as physical interface to provide foundation of new interaction paradigm [1]. The relative pointing on mobile devices screen is transferred to distant screens to reduce interaction distance. The Point & Shoot [2] technique employs visual codes to set up an absolute coordinate system for object selection with mobile phones. To create a self-contained system without fiducial markers, Boring et al. leveraged preformed content arrangement to determine the spatial relationship between the mobile device and display screen [5]. Chang and Li [10] allowed arbitrary content arrangement by matching the captured images with content on display screens. The method does not support continuous interaction as content snapshot are required in advance for further analysis.

Some works support fluent interaction through video stream captured by built-in cameras on mobile devices. Device screens act as portholes of physical world, through which users can interact with the physical objects. The world in miniature metaphor facilitates interaction at a distance. Touch Projector [6] affords bimanual interaction with one hand holding a mobile phone and the other hand interacting with remote screens through live video on the mobile phone. Virtual Projection [3] uses mobile phone as a controllable projection metaphor to interact with content on stationary displays. More works can be referred to survey [26]. As mobile devices have to occupy users’ hands when they are used as the manipulation metaphors, these methods can not provide friendlier freehand interaction. Additionally, interaction context is scaled down on smaller display screens, impeding the precision of fine manipulations.

2.2 Freehand interaction

Freehand interaction has been explored to deliver natural and intuitive yet effective interaction [30], especially for non-desktop scenarios (e.g., mid-air interaction) when keyboard and mouse devices are not appropriate or available. Previous works [19] [29] require fiducial markers or digital gloves to track hand gestures. Other works leverage Microsoft Kinect to detect hand poses and movements for freehand menu selection [12] and object manipulation [35]. However, these methods suffer from drawbacks. Instrumented gloves are encumbered and prone to induce fatigue. Fiducial markers and ambient sensors require delicate setups and calibrations.

Image-based methods [40] [4] have also been proposed to detect and recognize hand gestures using image processing technology, which are suitable for both closed and public environments [33]. As mobile devices become powerful, the image-based methods are promising for mobile devices as built-in cameras can be used without resorting to additional devices or sensors.

2.3 AR menu

Menus are widely used in traditional GUI to increase communication bandwidth with computers. However, using menus in the domain of AR is not straightforward, especially when the context is extended to three-dimensional environment. In order to support mid-air operations, non-pull-down menus such as ring menu [23] and tile menu [32] have been designed for manipulation in three-dimensional space.

As traditional mouse and keyboard devices have been discarded to reduce intrusion in AR applications, other inputs are developed for menu manipulation. In Studierstube [41], a tablet-and-pen device is used for menu selection. However, users have to take the tablet and pen with their hands. Some physical props such as fiducial markers [31] and hand fingers are also used to select menu items. The physical props are recognized and tracked using image processing techniques and then mapped for menu manipulation. To capture diverse motions such as wrist tilt and multiple pinch gestures, digital gloves shipped with motion tracking sensors are used [30].

2.4 Rethinking

Early systems provide seamless interaction in order to share the rare computing resources. The problem is partially alleviated as computing devices (e.g. smartphones) become miniaturized and ubiquitous. Mobile devices are promising to provide friendly user interaction when direct touch is unavailable or cross-display interaction is required. Although embedded cameras on mobile devices have been addressed as a suitable input [7], device intrusion disturbs user experiences. Freehand gesture-based method does not introduce much intrusion, but it has the deficiency of low precision [4]. Gesture-based menu system is promising to provide freehand interaction with higher communication bandwidth, but it is difficult to select menu items even with advanced-low pass filters [30].

Our lifetime interaction with the physical world tells that a large part of our interaction is daily operations, which
requires minimal visual attention as the functionality of objects is fully understood and works as expected [17]. An ideal user interface should embrace the richness of human sense and full skills acquired from our daily activities. It appears when and where it is required, otherwise it disappears to minimize visual attention and intrusion. Interaction should be also directly channeled to the physical objects rather than through a single workstation (as traditional GUI), so that it is not separated from the context where we reside and act.

To address the issues, we propose Ubii, a system to be invisible in the ecology of work places. Herein invisibility means it does not intrude user activities. Ubii is similar to the work [17] with a series of improvements. Rather than reprogramming functions of physical objects, we anchor our views on seamless interaction with individual physical objects without intrusion. Wearable devices such as Google Glass replace mobile devices to enable freehand interaction. 3D ring menu is used to support mid-air operations while pinch gesture is adopted to meet required precision of menu manipulation. By converging the technologies together, Ubii establishes a direct type of affordance, which is similar to the affordance of direct touch. The aim of our research is to illustrate a concrete interaction approach to move beyond current GUI paradigm which is bounded to computers with flat windows, mice, and keyboards.

3. SYSTEM DESIGN

Designing a system as Ubii requires consideration of physical affordance, which determines the physical objects perceived by the users, the virtual interface presented to the users, and the interaction for users to bridge the virtual interface and the physical objects.

3.1 Physical affordance

All objects present a sort of physical affordance, while we only focus on those that have relations with the digital world in our system. For instance, a computer provides an affordance for entering the cyberspace; a printer exhibits an obvious affordance for printing digital documents. Physical surfaces such as tabletop and architecture partitions (e.g., walls and ceilings) can also act as tangible interface. To name an example, a blank wall can be treated as a virtual bulletin board for sharing information among group members. Authorized members are able view notes and documents on the virtual bulletin board while walk-in visitors only see the blank wall.

Users require interface to interact with the objects. With Ubii users perform the affordance through the interface at a distance without physical touch. Remote operations reduce manipulation distance and facilitate cross-device interaction, which is especially preferable for different groups of objects that have connection with each other. Table 1 lists the physical objects and their applicable interaction which Ubii is interested in. Although the physical objects may be connected through network communication, interaction between the objects is not mutual from the user’s point of view. For instance, a user can drag and drop a document from a computer to a printer for printing service, but it is unable or meaningless to transfer anything from a printer to a computer. The applicable interaction between the objects is explained in detail as below:

**Computer:** Computers are able to interact with all objects listed in Table 1. Ubii enables a computer to interact with itself for documents and image manipulations (e.g., moving and zooming) at a distance. It can interact with other computers for document copy without portable mass storage devices (e.g., USB devices). A computer can ask a printer for printing documents without users’ physical touching. With a single gesture it sends documents for display on a large projector screen. In Ubii system it even can “upload” digital documents to physical surfaces (e.g., walls) for document sharing among multiple users. Figure 2 shows an embodiment of transmitting documents between two computers. A user pinches a document item from a computer, drags and drops towards another computer to finish transmitting the document between the computers, without physically operating the computers.

**Printer:** Printers are designed to work as passive receivers and do not have much interaction in general. In Ubii printers do not actively interact with other objects. However, as an important output device, it can passively interact with other objects including computers and physical surfaces. As illustrated in Figure 3, a user is dragging a document item from a computer and towards a printer. When the document item is moved near the virtual capture zone of printer affordance, the printer captures the document and starts to print it.

Figure 2: Left panel: a user picks a document item from a computer and drags it towards another computer. Right Panel: the user drops the document item to the other computer to finish document copy. 

Figure 3: Left panel: a user is dragging a document item towards a printer. The capture zone of the printer is indicated by a virtual blue overlay. Right Panel: The printer captures the document item and prints out the document.
3.2 Menu design

A menu system in the domain of AR applications is not necessarily to be two-dimensional. It is able to feature depth, rotation, and position in 3D space. We follows the criteria in [8] to design an interactive menu for our system:

**Placement:** Menu placement is classified into four categories [11]. We adopted the object-referenced placement to attach the menu to the physical objects. It is consistent with the design goal of breaking centralized interface into tailored individual interface associated with physical objects. Visual markers tagged on the physical objects are used to anchor the individual menus. The menus move along with the physical objects. Menus’ size vary according to relative spatial relations between user and physical objects to indicate depth and distance.

**Orientation:** Aligning the menu with users’ view makes it easy to read but may occlude the environment [8]; Rather, we align the menu with physical object surface which is tagged with the marker to improve readability. In addition, it achieves a better 3D spatial presence of the menu.

**Trigger mechanism:** A menu may be visible or hidden based on active state of the corresponding object. The menu will be visible if the physical object and its visual marker appear in user’s current view. A menu item is triggered whenever a user pinches the item with fingers. A capture zone is activated if a menu item is dropped to its capturing region. All triggers are translated as sockets to perform underlying network communication.

Attributed to the capability of short target-seeking time and low error in selection [9], ring menu is adopted to collaborate with hand-based interaction for freehand menu selection in Ubii. Menus’ items are distributed on an invisible ring around an anchor determined by the visual marker. Traditional ring menu enlarges the ring size or uses hierarchical structure when it is too crowded to display all menu items in a single ring. Neither method is feasible as items increase to squeeze the space. To solve the problem, we enable menu items to be rotatable along the ring. As illustrated in Figure 6, only a few menu items are active and others are folded. Active items disappear and hidden items appear as the ring menu rotates clockwise.
Rather than providing operating items (e.g., “translate” and “print”), operated items (e.g., document and image) are defined as menu items as physical affordance of the object items is already understood by users. Users are able to directly perform actions on object items without calling operating commands. For instance, dragging a document item implies copying, printing, or displaying the document according to the target that the document is dropped to.

Capture zone is designed to decrease required precision of interaction. Any menu item dropped to the range of a capture zone is captured by the receiver to trigger underlying commands. For instance, when a user selects a menu item (e.g., a document) of a computer, all printers, projector screens, and predefined physical surfaces in user’s current view are overlaid with their capture zones. If the item is dropped to the capture zone of a printer, the printer captures the document and starts to print it out. When the item is dropped to the capture zone of a projector screen, the computer connected with the projector will receive the document, and then asks the projector screen to display it.

3.3 Interaction design

Ubiib extends previous video-based interaction to hand gesture interaction without any digital gloves or fiducial marker tracking. However, pointing gesture is difficult for direct positioning operations (e.g., menu selection) because it cannot achieve the same high precision as physical input devices (e.g., mouse and stylus) in free spaces [30]. Directional gesture is ambiguous and error-prone due to lack of distinctive gesture delimiters [37].

Our essential design principle is to leverage multiple discrete gesture inputs to reduce the required precision of continuous hand gesture. The pinch gesture is stable and precise attributed to its discrete and unambiguous states of fingers together and apart [40]. In addition, it appears natural to users as it is evocative of grabbing a physical object [30]. Pinch gesture is suitable for interaction which requires high precision, such as menu picking and dropping. Combined with directional gesture, it also can be used for coarse interaction, such as dragging. Ubiib integrates multiple hand gestures to support several types of freehand interaction:

- Pick: a pinch gesture (as illustrated in Figure 7a) is interpreted as a pick operation. The pick operation should be performed on a menu item, otherwise it is invalid.
- Drop: an un-pinch gesture (as illustrated in Figure 7b) acts as dropping the selected item. A pick operation should always finish with a drop operation.
- Drag: triggered when a directional gesture is performed with a pick operation (as illustrated in Figure 7c). The selected item is moved along with directional gesture if only a single drag operation is acted. The drag interaction terminates whenever a drop operation happens.
- Rotate: a combo operation of two pinch gestures that move reversely along the Y direction on screen space (as illustrated in Figure 7d). The interaction is used to rotate a ring menu which is located between the two pinch gestures. Relative distance is mapped as rotation angle. Rotation operation stops whenever either or both pinch gestures end.
- Zoom: a combination of two drag operations performed on the same item (as illustrated in Figure 7e). Relative movement is transferred to scale up and down the item size. As one or both of the drag operations end, the zoom operation is terminated.

The transition of different interactions is illustrated in Figure 8. Pick and drop are instantaneous interactions. All interactions start with the pick and end with the drop operation. The other three start from the pick operation and end to the drop operation. They also enclose self-transitions for continuous interaction.
4. IMPLEMENTATION

Implementing our system requires technologies from fields of object tracking and hand gesture recognition. We employed a QR-based method for both object tracking and pose evaluation. A double-threshold algorithm was proposed to accurately detect pinch gestures. Our work explored the synergetic benefits that go beyond the sum of multiple technologies to achieve our design goals.

Figure 9 shows the system workflow. The built-in camera on Google Glass takes live video of the scene around users, which is streamed for image analysis of object tracking and hand recognition. The system extracts object information of IP address and object type from QR code, which is used for physical affordance recognition and network communication. Camera pose is also evaluated by calculating the homography of QR code images. Virtual camera is updated according to the real camera pose to render 3D menus from corresponding viewpoint, making virtual menus align with surfaces of physical objects. Menus are rendered as AR overlays to blend with video stream, and then displayed on Google Glass display prism.

Hand contours are extracted from video stream using a modified inexpensive method based on [40]. The contours are then filtered using double thresholds to determine pinch gestures. Users are able to interact with physical objects through virtual menus using pinch gestures, which are interpreted as socket to perform communication between physical objects through underlying network.

4.1 Object tracking

Ubii heavily relies on the built-in camera on Google Glass to understand the domain environment. The object tracking involves in recognizing object and evaluating spatial relationship between objects and users. Rather than non-intrusive markerless methods such as [22], we adopted the marker-based baseline, which is generally more accurate and robust to image distortion and illumination variation [28]. In our system QR codes are used as fiducial markers to encode additional information for understanding object functions.

As illustrated in Figure 10, object information including IP address and object type is encoded into QR code in advance using the qrcode tool\(^1\). For non-wired physical objects such as physical surfaces (e.g., wall and tabletop) and projector screens, IP addresses of their counterparts, namely computers storing documents on physical surfaces and computers connecting to projectors for projection display, are encoded. Camera calibration is also performed beforehand to calculate the intrinsic matrix \(K\) with the classic chess board method [42]. At runtime whenever one or more QR codes appear in the live video stream captured by embedded camera, the system resorts to the integrated Zbar library\(^2\) to decode object information from QR codes. It is then able to determine menu item layout according to the extracted information. The large finders in QR codes are also extracted to get the homography matrix \(H\) by solving the coordinate vector-valued function, which is then used to evaluate the pose matrix \(R|T = K – I \times H\). In practise, we adopted a tricky implementation by using the more efficient and accurate solvePnP method from the library OpenCV\(^3\). The intrinsic matrix \(K\) is mapped to the Normalized Device Coordinates (NDC) in OpenGL ES rendering system for perspective projection, while the pose matrix \(R|T\) should be flipped along X axis for consistently locating virtual menus in OpenGL coordinate frame.

In Ubii QR markers are added to the objects, which may be not consistent with our vision of designing a self-contained system. We view QR markers as a temporary and minor violation of our premise, and believe that advances in ubiquitous sensors and smart devices will help us to understand environment without additional annotation in future.

4.2 Hand gesture recognition

Pinch gesture provides a simple and reliable way to detect when interactions start and end without additional gesture delimiters. It can be determined by the thumb and index fingers touching together to make a hole as illustrated in the Figure 11a. We basically follow Wilson’s vision-based method [40] to detect one and two-hand pinching gestures, with several improvements in our system.

The first step is to distinguish hands and background using the well-studied segmentation and connected components techniques in computer vision community. In Wilson’s method the background of interaction context is constrained to a black computer keyboard. As the background is constantly darker than the foreground hands, the method chooses to separate the background from the image. In Ubii the background varies according to user’s view angle. Alternatively, we extract the foreground hands from the image by using hand skin color as reference color. With the same reason in Wilson’s method, it is reasonable to take the largest pixel components as hand sections because hands generally occupy the largest region from user-centered perspective.

Contours of hand section are then extracted. The largest one is the outer contour of the hand section as illustrated in Figure 11b. Any inner contour indicates a hole in the hand

---

\(^1\)https://pypi.python.org/pypi/qrcode

\(^2\)https://github.com/ZBar/ZBar

\(^3\)http://docs.opencv.org/
Figure 11: The pinch gesture detection. a) a typical pinch gesture; b) outer contour of hand is extracted; c) and d): red closed regions are recognized as pinch holes; e) and f): green closed regions are eliminated as non-pinch holes.

section. Ambient illumination and other non-pinch gestures may also produce inner contours. Our method adopts contour size to filter out the wrong contours. Any inner contour with size out of given bounds is discarded. The bounds are determined by the acceptance and elimination curves as illustrated in Figure 12. The elimination curve represents accuracy of excluding non-pinch gestures that have inner contours in terms of a lower threshold of the inner contour’s size. The inner contours of non-pinch gestures are generally small. The gestures will be more accurately expelled if we set the lower threshold to be larger. Conversely, holes in pinch gestures are normally large. Most pinch gestures detection will pass if the threshold is small, as shown by the acceptance curve that recognition accuracy declines when the threshold gets larger. We chose a lower bound 0.6 and upper bound 0.8 to achieve over 85% accuracy of both pinch gesture detection and non-pinch gesture elimination.

The method works in both one and two-hand pinch gesture detection as shown in Figure 11c and Figure 11d. It is also robust to eliminate other non-pinch gestures. Figure 11e and Figure 11f show two gestures with holes in hand sections, but they are eliminated in our method as the pinch holes are smaller than the given lower bound value. In practice, it is applicable to uncheck the upper bound as result set of a small upper bound always includes result of a large upper bound. However, size of pinch hole is constrained to hand anatomy. A reasonable upper bound guarantees excluding outliers.

The pinch gesture detection relies solely on the hole between thumb and index fingers. The method fails if the hole background is occluded by other curled fingers [40]. It neither works when the hole is invisible to the camera, such as the thumb and index fingers being horizontally coplanar. We see it to be a reasonable compromise for freehand interaction without device intrusion.

5. EVALUATION

We designed several experiments to evaluate the system from different aspects. Participants were invited to perform the experiments. Their task completion time and failed trials were measured and compared. The participants were asked to answer paper questionnaires after the experiments. A technology acceptance model (TAM) [14] was adopted to evaluate user experience.

5.1 Participants

We invited 10 participants (6 males and 4 females) from other departments in our campus, with ages between 19 and 31. They are frequently computer users and familiar with common computer operations including document copy, print, and projector display. Several (4) have some experience of gesture interaction, including Microsoft Kinect and Wii for computer games. Three of them have experience or hear about AR techniques.

5.2 Experiment design

We designed four tasks in association with selected physical affordance in Section 3.1. The tasks were conducted in a closed work space with dimensions of about 10m*5m. The work space is equipped with several interconnected computers, printers, and a set of projector and display screen. Several USB devices were available for document copy in our experiments. The computers had installed instant messaging software Skype\(^4\) and share tool Dropbox\(^5\) for transmitting and sharing document. We registered several accounts in advance to use the software. Computers, printers, and projector were running so that participants did not need to login or switch on devices. The computer connected with the projector was also configured beforehand for projection display. Computers’ names, and printers’ name, document names and paths on computers, were listed on task cards for reference.

**Ex1: Copying documents between computers.** Participants were required to copy specified documents between assigned computers with USB flash copy, Skype, and Ubii. By using USB flash copy participants had to copy documents between different computers and USB devices. They also needed to find other Skype accounts on target computers in order to transmit documents through Skype. In Ubii, participants needed to perform pick, drag, and drop operations with their

---

\(^4\)http://www.skype.com/

\(^5\)https://www.dropbox.com/
hands to finish document copy. Rotate operation may be required when the specified document items are folded in menus.

Ex2: Printing documents. Participants were asked to print specified documents on assigned printers by using functions provided by computers and Ubii. In traditional way participants had to select “print” menu item and fill the configuration panel such as assigning printer name. In Ubii participants had to pick, drag, and then drop the specified document item to the capture zone of assigned printer.

Ex3: Displaying documents on projector screens. Participants finished the task of displaying specified documents on the projector screen with traditional USB flash copy and alternative way of Ubii. In traditional way participants copied documents to the computer connected with projector using USB devices, and then opened the document for projection display. Using Ubii system, participants directly interacted with the projector screen rather than the computer connected with the projector.

Ex4: Sharing documents. Participants had to share documents with other participants through Dropbox and Ubii. By using Dropbox participants had to copy specified documents to Dropbox share folders or share Dropbox link, and then wait for documents uploading or receiving in Ubii. Participants used an architecture wall as virtual share space for uploading and downloading document with drag and drop interaction.

Each participant was required to perform the four tasks, and each task was performed 4 times. The tasks were presented in random order among the participants. We got a total of 10 (participant) × 4 (task) × 4 (repetition) = 160 times of operations, with each task repeatedly performed 40 times. Each participant completed the whole study within 60 minutes or less.

5.3 Results & discussion

Figure 13 compares participants’ task completion time using Ubii system (green bars) and other traditional ways such as document copy through USB devices and Skype, and document share through Dropbox. The time cost of Ubii is much less than other traditional methods in all four tasks. Particularly, it is (M=7440ms, SD=360ms) almost 4 to 7 times faster respectively than transferring on Skype (maroon bar, M=33780ms, SD=3320ms) and USB flash copy (brown bar, M=33475ms, SD=3530ms) in Ex1. It (M=7690ms, SD=300ms) also achieves high performance gain compared with traditional projection display using USB devices (brown bar, M=56370ms, SD=6510ms). Using USB flash copy requires plugging in, plugging out, and copying documents between USB devices and computers. Ubii simplifies the manipulations without directly physical touch. For document printing Ubii (M=10250ms, SD=380ms) eliminates operations of clicking “print...” menu item and selecting specified printer from printer list required by traditional method (blue bar, M=20250ms, SD=574ms). We found that some participants referenced to task cards when they selected a printer from configuration panel. Ubii (M=8740ms, SD=310ms) is also more efficient in document sharing compared with Dropbox sharing (yellow bar, M=13510ms, SD=1130ms), as it does not require confirming target user’s Dropbox account, which is normally not the same as target computer’s account.

We compared completion time (green bars) of different tasks using Ubii. The task completion time includes both operation time and document transfer time. The most time-consuming task is printing operation, followed by document copy, projection display, and share. As document printing requires most time due to document transfer between computers and printers, namely almost three times longer (M=6100ms) than that between computers (M=2100ms), it is in fact the fastest operation (M=4200ms, SD=200ms) comparing to document copy (M=4840ms, SD=310ms), projection display (M=5200ms, S=330ms), and document sharing (M=6650ms, SD=300ms).

We measured failed trials and ratios of each task as illustrated in Table 2. All failed trials are induced by failures of Ubii. Document sharing has the highest failed ratio of 10%, and document copy with the lowest of 5%. No participant encountered more than twice failures in single task. The failures mainly lie in the wrong detection of hand gestures, which are caused by fast hand movements and background color.

In addition to profile the system from the aspect of runtime performance, we evaluated user experience of Ubii using the TAM model, which includes four criteria of perceived enjoyment (PE), perceived usefulness (PU), perceived ease of use (PEOU), and intention to use (IOU). We used the Likert scale\(^6\) to quantitatively measure the four criteria, which scale from 1 to 5 scores, with strong disagree (1), disagree (2), neutral (3), agree (4), and strongly agree (5). As illustrated by Figure 14, the four terms in each task are positive in general. In Ex2 (document printing) and Ex3 (projection display) the first quartiles (Q1) pass over 3, and medians (Q2) are not lower than 4. The bottom whiskers also end

<table>
<thead>
<tr>
<th>Task</th>
<th>Failed Trial</th>
<th>Failed Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ex1</td>
<td>2</td>
<td>5%</td>
</tr>
<tr>
<td>Ex2</td>
<td>3</td>
<td>7.5%</td>
</tr>
<tr>
<td>Ex3</td>
<td>3</td>
<td>7.5%</td>
</tr>
<tr>
<td>Ex4</td>
<td>4</td>
<td>10%</td>
</tr>
</tbody>
</table>

\(^6\)http://en.wikipedia.org/wiki/Likert_scale
6. CONCLUSION

Uubi provides a decentralized interface system for manipulation between the physical and digital worlds. User interface and interaction are delicately designed to afford free-hand user experience. The system is evaluated from both aspects of technical and user experience. By interacting with objects at a high-level of physical affordance rather than underlying digital counterparts, the system helps users perform operations in much convenient and natural way. It has been proved to be useful for simple and frequently-performed manipulations with connected digital devices in closed environments.

In our experiments, we found that most failures were caused by hand gesture detection. The color-based method is chosen to relieve computational cost, but it is not accurate enough in backgrounds with similar color of hand skin. In future we plan to combine depth and visual cameras to improve the accuracy. Another problem is the limitation of poor computational capability and limited power capacity of Google Glass. We will try to leverage device offloading strategies to outsource computationally intensive tasks to companion mobile devices. The method has been proved to improve both real-time performance and runtime sustainability.

7. ACKNOWLEDGMENTS

This work has been partially supported by the European Commission under the Horizon 2020 Program through the RAPID (H2020-ICT-644312) project.

8. REFERENCES


