BlueTone: A Framework for Interacting with Public Displays Using Dual-Tone Multi-Frequency through Bluetooth

David Dearman and Khai N. Truong
University of Toronto
Department of Computer Science
Toronto, Ontario M5S 3G4, Canada
{dearman, khai}@cs.toronto.edu

ABSTRACT
Large information displays are common in public and semi-public spaces but still require rapid and lightweight ways for users to interact with them. We present BlueTone, a framework for developing large display applications which will interpret and react to dual-tone multi-frequency sounds transmitted from mobile phones paired with the display using the Bluetooth headset profile. BlueTone enables text entry, cursor manipulation and menu selection without requiring the installation of any special software on a user's mobile phone.

Author Keywords
BlueTone, public display, at a distance interaction, Bluetooth

ACM Classification Keywords
H.5.2.User Interfaces: Interaction Styles; I.3.6 Methodology and Techniques: Interaction techniques.

General Term: Design

INTRODUCTION

Large information displays are common in public and semi-public spaces such as malls, schools and airports. These displays provide a ubiquitous channel for disseminating information and a medium to support opportunistic collaboration between people who frequent a space [4, 6]. A large public display affords the opportunity for many people to view and interact with the information space simultaneously. We argue that the method of interacting with the large displays must be lightweight and should support the spontaneity with which a person may discover the space.

The majority of techniques that facilitate interaction with public displays involve direct manipulation [6], instrumenting the display or space with cameras to track users gestures [3, 8-11, 13, 14], or installing software on the user's mobile device to capture and communicate the user's intentions [7, 8]. In contrast, techniques used by e-Campus [5] and Blinkenlights [2] allow users to opportunistically annex [12] their mobile phone to facilitate spontaneous interaction with a public display; without requiring the installation of additional software on the mobile device. E-Campus [5] users interact with a public display and control applications by manipulating the Bluetooth 'Device Name' of their mobile phone (or similar Bluetooth enabled device) to specify a service and action. For example, *"ec flickr sunshine"* directs the display to search Flickr using "sunshine" and show the resulting photos. Users of Blinkenlights [2] control the display by calling a phone number and pressing the numeric 1-9 keypad to perform simple predefined actions.

In this paper, we present BlueTone, a framework that supports opportunistic interaction with public displays using any Bluetooth enabled mobile phone. BlueTone builds upon and synergizes the techniques of e-Campus and Blinkenlights to enable a lightweight interaction using all Bluetooth enabled mobile phones' ability to act as an audio gateway and transmit dual-tone multi-frequency (DTMF) sounds.

Related Work

Direct manipulation of interface widgets using a finger or pen currently remains the central interaction paradigm for large displays. Although, direct manipulation using pointing and clicking is simple and seemingly intuitive, it limits the number of simultaneous collaborators with respect to how many people can fit around the display and requires that the display is within reach. To address these issues, numerous techniques for interacting with large and public displays *at a distance* have been developed [1, 2, 4-11, 13, 14].

A keyboard and mouse is a simple solution for public kiosks,

Figure 1. A user controlling the playback of a YouTube video on a public display from his mobile phone using BlueTone.

UbiComp 2009, Sep 30 – Oct 3, 2009, Orlando, Florida, USA.

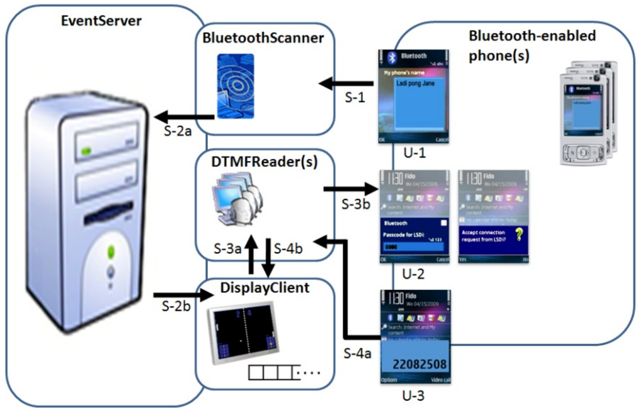

Figure 2. Overview of BlueTone's system architecture.

but limits the use and the relevance of the information display to a single user. Dynamo, a public interactive surface, used multiple keyboards and mice to support collaboration among small groups of users [6]. However, the use of a keyboard and mouse is most appropriate when the display is one that can be used while seated.

Predominantly, research in this area has focused on instrumenting the display or the space around the display with cameras to track a cursor [8], markers [7], the laser spot from a laser pointer [9, 10, 11], and the gestures a person performs with their body [13, 14] or physical artifact [3]. Using only a handheld camera, Direct Cursor tracked the position of the cursor on a large display and repositioned the cursor with respect to the centre of the camera frame [8]. Similarly, Jeon *et al.* [7] used a camera phone to support cursor movement and object selection using optical flow and object markers. Oh and Stuerzlinger [10] allowed for selection and manipulation of interface widgets by adding a button to a laser pointer. Olsen and Nielsen [11] implemented scrolling, selection and graffiti input by tracking the dwell time and positioning of the laser spot using a webcam. Semantic snarfing used a laser pointer in conjunction with a handheld computer [9]. The laser pointer indicated the approximate area to display on the handheld which the user could then interact with on the display. However, calibrating the camera to track the laser spot can be difficult [9, 10, 11] and the scalability to allow for multiple simultaneous users can be costly [11].

Vogel and Balakrishnan tracked the gesture of a user's hand to differentiate between numerous interaction modalities [14]. Additionally, they used the physical proximity of the user with respect to the display to differentiate between implicit and explicit modes of interaction [13]. VisionWand [3], developed by Cao and Balakrishnan, afforded command input by tracking the posture and gestures performed using a simple plastic wand using two webcams.

THE BLUETONE SYSTEM

In this section, we describe the BlueTone architecture and how it enables interaction with the content of a public display using a mobile phone as an audio gateway to transmits predefined dual-tone multi-frequency (DTMF) sounds. Similar to e-Campus [5] and Blinkenlights [2], our system does not require a user to install any software on her phone. However, whereas e-Campus only allows users to specify the content to display, BlueTone allows users to specify and control content/applications, and actively collaborate with other users. Similarly, Blinkenlights requires a device that can place phone calls. BlueTone works with any Bluetooth capable mobile device that can transmit DTMF, not just mobile phones.

System Architecture

BlueTone is comprised of a large display and *at least one* computer with a Bluetooth adapter (in our case, a Rocketfish USB adapter using the WIDCOMM Bluetooth stack) acting as an event server *(EventServer)*. The *EventServer* reads and interprets the user's control actions (DTMF sounds generated by pressing buttons on the phone's numeric keypad) into interactions with the application shown on the large display.

To interact with a large display, users are first required to rename their phone in a manner similar to the e-Campus (Figure 2: U-1). The BlueTone command string (*e.g., "btone pong jane"*) follows the format:

```
<display_name> <application> <params>
```

The command string is discovered by a server thread *(BluetoothScanner)* when it performs a Bluetooth scan of nearby devices (Figure 2: S-1). The *EventServer* relays both the *friendly name* along with the phone's Bluetooth MAC address to the client application *(DisplayClient)* which controls the display (Figure 2: S-2a, 2b). The *DisplayClient* launches the application specified by the user[1] and shows a context menu for interacting with the content using the phone's keypad (*i.e.,* the context-menu is dynamic and application specific). The *DisplayClient* then informs the *DTMFReader* to pair with the user's phone as an audio gateway using a Bluetooth headset profile (Figure 2: S-3a, 3b, U-2). The *DTMFReader* captures and processes DTMF sounds generated when the user presses the keypad (Figure 2: S-4a, U-3) using a discrete Fourier transform to determine the strongest two audio frequencies. The two audio frequencies are then used to identify which keypad button was pressed. The *DTMFReader* relays the key press to the *DisplayClient* which executes the corresponding action (Figure 2: S-4b).

The *BluetoothScanner* thread and the *DTMFReader* thread can run on either the same machine or different machines. When both threads share a Bluetooth adapter, the *BluetoothScanner* thread only scans after a user has quit an application, because a Bluetooth device scan prevents other services to be performed on the same adapter at the same time. When the two threads use their own adapter, the

[1] The DisplayClient can be customized to immediately display the requested content or queue the content and show it for a predefined amount of time.

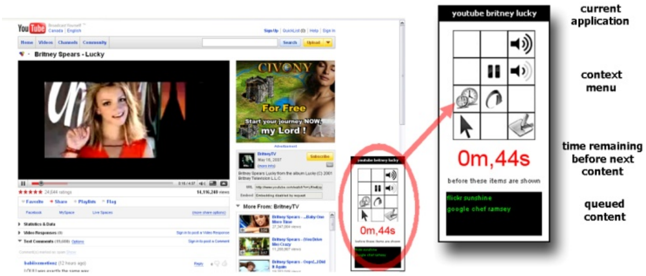

Figure 3. BlueTone context menu for interacting with YouTube. For example, the user will press '3' to increase the volume and '5' to pause playback.

BluetoothScanner thread scans continuously, with each scan taking between 8-15 seconds to complete.

In our architecture, the *DTMFReader* can be decoupled from the rest of the system, thus multiple *DTMFReader* threads can run at the same time. This allows multiple users to control the display at the same time and enables collaboration (*e.g.,* two or more people can play pong or another collaborative game together).

BlueTone-Enabled Services

BlueTone is inspired by e-Campus [5], as such BlueTone supports many of the applications found in the e-Campus system (*e.g.,* Google, YouTube, and flickr). For example, as shown in Figure 3, a user is able to watch a particular YouTube video, but also has the added ability of controlling audio/video playback. The user presses '5' on their mobile phone to pause the video and '3' to increase the volume.

BlueTone enables developers to create and deploy a broad spectrum of applications that the user can interact with through her current mobile phone. For any application registered with the system, the designer can specify the actions to be performed when different context menu items are invoked (through a custom AutoIt[2] script that the *DisplayClient* will execute).

By default, the * key allows the user to toggle between using the keypad as a directional keypad (centered on 5) for manipulating the cursor or using it to invoke the application specific context menu items. One use of this feature is when a user needs manipulate controls on the screen for which an application designer has not added as a context-menu item. Similarly, the # key allows the user to use the keypad as a multi-tap text-entry technique. This is useful for entering short text into forms on Web pages.

Finally, our system allows multiple users to collaborate when supported by the application. For example, when there are multiple users who have specified that they want to play pong, the application would allow them to compete against one another. Figure 4 shows a 2 player pong game which allows each user to press '5' to move the paddle up and '8' to move it down (or '2' and '0' to move the paddle up and down faster).

[2] *http://www.autoitscript.com*

DESIGN CHALLENGES & LIMITATIONS

BlueTone allows application developers to create and deploy public and large display software that user's can interact with using a mobile phone or Bluetooth enabled mobile device. Our framework addresses limitations in Davies *et al.*'s e-Campus by giving users the ability to interact with the content of the display, and Blinkenlights by not requiring an active phone connection. In this section, we discuss the limitations and challenges involved in supporting user interactions with large screen displays, and consider other light-weight mechanisms for enabling usage of large screen displays from a user's mobile phone. We will compare our approach against that of a Web and texting service.

Device & Service Requirements

BlueTone does not require the user to install any software on her phone, but does require that the user have a Bluetooth enabled phone that they are willing to pair with the large display. Alternatively, a user can visit a unique web site to interact with a specific large display. The Web site could also provide context-menus to allow interaction with the display in a very similar manner to how our system works. Such a technique requires that the phone has mobile Internet service enabled or Wi-Fi support. In contrast, a texting service would impose the fewest device and service requirements. To interact with the display, the user would simply texts commands to a pre-defined phone number. This technique has the lowest hurdle to adoption, however, has latency and robustness issues, which we discuss next.

Latency & Robustness

BlueTone has a ~10-15 seconds delay after the user has renamed her device with when the *DisplayClient* adds the command string to the display queue. This latency is caused by the Bluetooth discovery process. Additionally, if the device pairs with the large display for the first time, the user must also go through the process of specifying a Bluetooth pass code and authorizing the connection. In subsequent interactions with the large display, the user will only face the ~10-15 second time delay. While the phone is paired with the large display and no data is being transferred, the Bluetooth

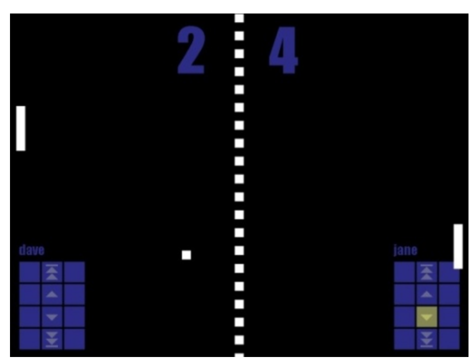

Figure 4. BlueTone context menu for pong. For example, pressing '5' on the phone's numeric keypad will move the paddle up and '8' will move it down.

connection will enter a power saving mode. As a result, the DTMF for the first few subsequent key presses are not transmitted. To address this problem, the context menu changes in shade to let the user know that the phone has been silent for a period of time. Furthermore, when the system recognizes valid DTMF sounds, the context menu highlights the associated button on the keypad.

In comparison, a Web-based solution does not face similar robustness issue. Any latency is a result of the phone's network connection to the Website. However, a texting-based approach could potentially face significant latency because the text message could take between only a few seconds to days to reach the large display (in some cases, when texting across different providers, a message can also be lost). Additionally, the variability in timeliness of text messages could result in commands arriving in the wrong order. In some application usage scenario (such as video game play), the specific sequence of the user's interaction is important. As a result, this approach is the least robust.

Spoofing

Of the three techniques, only BlueTone is exposed to spoofs. To minimize spoofing, the display can be designed to show the unique URL to a Web site for controlling it or a number to text commands. However, during the initial interaction stage with BlueTone, the user must allow the display to pair with her phone. At any time during this process another device can spoof the name of the display and attempt to connect with the user's phone. Currently, our system is designed to show a queue of the discovered devices and pairing only happens immediately before the user's content gets shown. As this occurs, the display tells the user a random pass code to enter (this step needs to happen only once). Additionally, this issue can be minimized by having the phone display the type of Bluetooth connection that is being requested (*e.g.,* to prevent a culprit from requesting an OBEX connection).

CONCLUSION

BlueTone is a framework for developing interactive large display applications without requiring the installation of additional software on a user's mobile device to support input. The phone and display are paired using the Bluetooth headset profile, allowing the phone to communicate dual-tone multi-frequency sounds that the display's application can interpret and react to. BlueTone builds upon the work of e-Campus [5] and Blinkenlights [2], to facilitate opportunistic interaction with public displays using DTMF sounds communicated over a mobile phone's Bluetooth connection to enable rapid and lightweight interaction that was not previously possible.

ACKNOWLEDGEMENTS

We thank Larry Rudolph for showing us that it is possible to stream music from a mobile phone to a large display using the Bluetooth headset profile and inspiring this work.

REFERENCES

1. Ballagas, R., Borchers, J., Rohs, M., and Sheridan, J.G.: The smart phone: a ubiquitous input device. IEEE Pervasive Computing 2006, 5(1), 70-77.
2. Blinkenlights. http://www.blinkenlights.net
3. Cao, X. and Balakrishnan, R.: VisionWand: interaction techniques for large displays using a passive wand tracked in 3D. In *Proc. UIST 2003*, ACM Press (2003), 173-182.
4. Cao, X., Massimi, M. and Balakrishnan, R.: Flashlight Jigsaw: an exploratory study of an ad-hoc multi-player game on public displays. In *Proc. CSCW 2008,* ACM Press (2008), 77-86.
5. Davies, N., Friday, A., Newman, P., Rutlidge, S. and Storz, O.: Using Bluetooth device names to support interaction in smart environments. In *Proc. MobiSys 2009,* ACM Press (2009), 151-164.
6. Izadi, S., Brignull, H., Rodden, T., Rogers, Y. and Underwood, M.: Dynamo: a public interactive surface supporting the cooperative sharing and exchange of media. In *Proc. UIST 2003*, ACM Press (2003), 159-168.
7. Jeon, S, Hwang, J., Kim, G.J. and Billinghurst, M.: Interaction techniques in large display environments using hand-held devices. In *Proc. VRST 2006,* ACM Press (2006), 100-103.
8. Jiang, H., Ofek, E., Moraveji, N. and Shi, Y.: Direct pointer: direct manipulation for large-display interaction using handheld cameras. In *Proc. CHI 2006*. ACM Press (2006), 1107-1110.
9. Myers, B.A., Peck, C.H., Nichols, J., Kong, D. and Miller, R.: Interacting at a distance using semantic snarfing. In *Proc. Ubicomp 2001*. Springer-Verlag (2001), 305-314.
10. Oh, J-Y. and Stuerzlinger, W.: Laser pointers as collaborative pointing devices. In *Proc. GI 2002*, 141-149.
11. Olsen, D.R. and Nielsen, T. Laser pointer interaction. In *Proc. CHI 2001*, ACM Press (2001), 17-22.
12. Pierce, J.S. and Mahaney, H.E.: Opportunistic annexing for handheld devices: opportunities and challenges. Human-Computer Interaction Consortium (HCIC) 2004, Fraser, CO.
13. Vogel, D. and Balakrishnan, R.: Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In *Proc. UIST 2004,* ACM Press (2004), 137-146.
14. Vogel, D. and Balakrishnan, R.: Distant freehand pointing and clicking on very large, high resolution displays. In *Proc. UIST 2005,* ACM Press (2005), 33-42.