
No app needed: Enabling mobile phone communication with a tourist kiosk using cameras and screens

**Scott Carter, Jennifer Marlow,
Matthew Cooper**

FX Palo Alto Laboratory, Inc.
3174 Porter Drive, Palo Alto, CA
94304
carter,marlow,cooper@fxpal.com

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

Copyright held by the owner/author(s).
UbiComp/ISWC'17 Adjunct, September 11–15, 2017, Maui, HI, USA
ACM 978-1-4503-5190-4/17/09.
<https://doi.org/10.1145/3123024.3123188>

Abstract

For tourists, interactions with digital public displays often depend on specific technologies that users may not be familiar with (QR codes, NFC, Bluetooth); may not have access to because of networking issues (SMS), may lack a required app (QR codes), or device technology (NFC); may not want to use because of time constraints (WiFi, Bluetooth); or may not want to use because they are worried about sharing their data with a third-party service (text, WiFi). In this demonstration, we introduce ItineraryScanner, a system that allows users to seamlessly share content with a public travel kiosk system.

Author Keywords

Public displays; Mobile phones; Tourist itinerary planning

ACM Classification Keywords

H.5.m. [Information Interfaces and Presentation (e.g. HCI)]:
Miscellaneous

Introduction

Despite the global diffusion of mobile devices and networks, our field work and that of other researchers has found that tourists navigating new environments continue to rely on public displays such as large printed maps or interactive digital directories. These displays can provide a curated set of sites and events specific to a particular place and

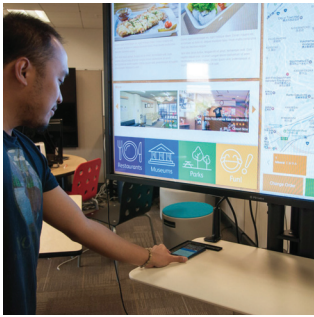


Figure 1: The ItineraryScanner system.

are sufficiently large to allow group discussion and collaboration. These displays can fill gaps in local knowledge when mobile device coverage is poor (for example, in subterranean subway systems).

While public displays are useful on their own, the ubiquity of personal devices means that designs of digital public display systems must permit users to transfer data back-and-forth between the display and their mobiles. Our observations of public tourist kiosks in the wild have revealed that the most common means of enabling such transfers is via QR code (which requires that users have a separate QR code reader app already installed on their phones).

However, our field work has also revealed that users (particularly from North America, or other countries where the use of QR codes is not widespread) rarely install these required applications. First, users are often not aware of the need to install apps for public display interaction while planning their trip. Even if users are aware of code reading apps, it may be difficult to anticipate their importance. Additionally, when tourists seeking information actually face a public display during their trip, installing new apps can be inconvenient if they are heading to another destination, have poor internet connectivity, or are anxious about the privacy implications of downloading third-party apps in the wild.

To address these concerns, we introduce ItineraryScanner, a public kiosk system that allows people to create and edit itineraries using seamless interactions between a mobile phone and a larger public display (see Figure 1). Users can build an itinerary using recommendations shown on the public display, or send recommendations to the display via a camera-based scanner integrated into the kiosk. Users can share recommended sites with the display using either barcodes generated with any third-party barcode application, or simply having the kiosk's camera scan their mobile

device's screen. The kiosk system will automatically determine the site, add it to the current itinerary, and update a list of other recommended sites that complement the current itinerary. Users can also scan content from paper flyers or other documents (see Figure 2).

When users are satisfied with their selections, they can take a photo of the itinerary on the public display screen, or use a barcode scanning application on their phone to scan a QR code on the public display and download the itinerary to their device. In this way, on-the-go tourists can leverage their personal devices and interactive public displays without the hassle or worry of downloading a third-party app.

Related work

Most commercial applications attempt to facilitate data migration between public displays and personal devices with technologies that require a good internet connection or the installation of additional applications. Example prototype systems that enable display-mobile information transfer, such as PresiShare [2] or the WrayDisplay [3] require additional software to be installed on a user's mobile device.

Additionally, information transfer is frequently done in one direction, where users can transfer information from a large display onto their mobile devices. There may be cases where a tourist wishes to share information from their own mobile phone to a larger public display, for example to engage in collaborative trip planning and discussion. CozyMaps [1] presents one example of collaborative sharing of map-based content from tablets to a larger screen, but once again, it requires a special app that depends on an internet connection.

Our system addresses the aforementioned barriers to phone-display sharing with a solution that does not require the installation of additional apps or even a mobile phone with

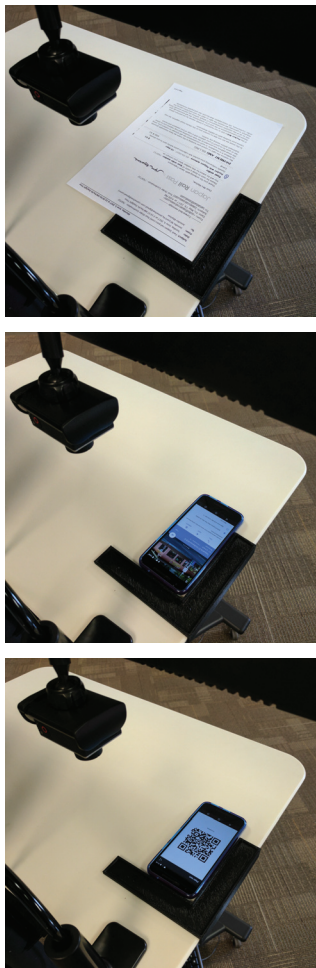


Figure 2: ItineraryScanner can scan text from paper flyers or phone screens as well as QR codes.

internet connection to share information between a phone and a larger public display. It also allows users to contribute content to the display and take display content with them.

System

The ItineraryScanner system is a public display kiosk that includes three core components: an itinerary creation web app running on a public display; a camera and processing unit to scan mobile phone screens and documents; and a simple, network-based multimedia messaging system to pass content between the two. The scanning component processes video preview frames continuously in two separate threads. One thread detects and can process barcodes representing content from a set of well known travel and recommendation mobile apps (e.g., Yelp, Google Places, etc.). When it detects supported content it extracts the embedded content (such as the title of a restaurant, museum, or other place of interest) and sends it to the public display via the messaging system.

The other processing thread attempts to detect text in the image using well-known OCR tools (Tesseract¹). To reduce errors in the OCR analysis, the thread first normalizes image frames and determines the pixel-level difference between subsequent frames. Once the difference between frames falls below a predetermined threshold and is stable for a predetermined interval (usually a few seconds), the system saves a frame with values averaged over a smaller window as the baseline frame. The system then compares all further frames to the baseline. At this point the system is trying to determine when new content (e.g., a mobile phone screen or paper flyer) appears below the sensor. To do so, the system continues comparing incoming frames both to recent frames and the baseline frame until new frames are

both significantly different from the baseline and are relatively stable compared to recent frames. Having detected new content in the scanning bed area, the camera then snaps a higher resolution photo and processes that with an onboard OCR engine. The engine returns text regions, including bounding boxes and the text content within. The system combines proximate text regions using a connected components algorithm. It then uses heuristics to detect the most likely title text region in the image (the largest text block closest to the top left of the image). The text within this region is sent on to the messaging system.

Once the messaging system passes title text along to the itinerary app, the app performs a search against the Google Places API for the site, geofencing the search within a few dozen miles of the pre-configured location of the kiosk. If the application cannot find the requested site it shows a brief notification. If it does find the location, the application adds it to the current itinerary. The app furthermore queries other third-party services connected to the system to determine recommended sites nearby.

Conclusion

Our field work revealed that tourists can potentially take advantage of public displays to help them navigate unfamiliar environments and find out-of-the-way, local sites. However, despite the proliferation of new communication technologies, sending information between mobile devices and public kiosks remains a barrier. ItineraryScanner ameliorates this issue, allowing tourists to send content to a kiosk from mobile screens as well as paper flyers.

Acknowledgements

We thank John Doherty, Anthony Dunnigan, and Joseph Andrew de la Peña for their help.

¹<https://github.com/tesseract-ocr/tesseract>

REFERENCES

1. Kelvin Cheng, Liang He, Xiaojun Meng, David A. Shamma, Dung Nguyen, and Anbarasan Thangapalam. 2015. CozyMaps: Real-time Collaboration on a Shared Map with Multiple Displays. In *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '15)*. ACM, 46–51.
2. Matthias Geel, Daniel Huguenin, and Moira C. Norrie. 2013. PresiShare: Opportunistic Sharing and Presentation of Content Using Public Displays and QR Codes. In *Proceedings of the International Symposium on Pervasive Displays (PerDis '13)*. ACM, 103–108.
3. Nick Taylor and Keith Cheverst. 2009. Social interaction around a rural community photo display. *International Journal of Human-Computer Studies* 67, 12 (2009), 1037–1047.