

GateWatcher: Android Phone Based Video Surveillance System Using App Inventor

Sergio Sandoval-Reyes, Juan Carlos Castañeda-Martínez

Centro de Investigación en Computación, Instituto Politécnico Nacional, México
Sergio.Sandoval19@gmail.com; jc.castmart@gmail.com

Abstract. Video surveillance is an application that employs video cameras for the purpose of observing an area to monitor traffic, streets, facilities, and for home security. Video cameras can send video and audio via a wired or wireless network and the Internet, to watch live and recorded video from different devices, for example, cell phones. This paper describes GateWatcher an Android Phone based video surveillance system in which a video-doorbell is implemented using a cell phone and App Inventor. When a visitor rings the phone-based doorbell, you receive a sms message notification on your phone. You can accept this notification to watch the visitor returning another sms to the doorbell, which triggers its video-camera to start a Bambuser audio-video transmission to the Bambuser website via a Wi-Fi or a 3G network, which is accessed automatically from your phone to see and hear the visitor, but he/she cannot hear or see you. The application also allows communicating with the visitor through chat instant messages.

Keywords. Remote video surveillance, android, app inventor, bambuser.

1 Introduction

Video surveillance is an application that employs video cameras for the purpose of observing areas to monitor traffic, streets, facilities, and for home security. Video cameras can send audio and video via a wired or wireless network and the Internet, to watch remotely live and recorded video content from different devices. The traditional video surveillance systems like Closed Circuit Television (CCTV) or the PC based video systems require large and expensive equipment and do not provide mobility to monitor the premises remotely. Wireless networks such as Wi-Fi and 3G not only provide ubiquity but also large wireless bandwidth which make them possible to develop video applications for mobile phones such as an Android based video surveillance system.

Although there are many ways to use a smartphone to develop a surveillance system, most of them requires the knowledge of a classical computer programming language such as C, C++, Java, Python, etc. Lately however, visual programming languages such as Scratch [1], and App Inventor for Android [2], can also be used to interact with phones without the need of knowing the classical programming languages.

This paper describes GateWatcher an Android Phone based video surveillance system in which a video-doorbell is implemented using a cell phone and App Inventor. When a visitor rings the phone-based doorbell, you receive a sms message notification on your phone. You can accept this notification to watch the visitor returning another sms to the doorbell, which triggers its video-camera to start a Bambuser audio-video transmission to the Bambuser website via a Wi-Fi or 3G network, which is accessed automatically from your phone to see and hear the visitor, but he/she cannot hear or see you. The application also allows communicating with the visitor through chat instant messages.

The remainder of this paper is organized as follows: Section 2 presents a summary of works related to mobile phone based surveillance systems. Section 3 describes the design of GateWatcher using App Inventor. Section 4 shows the results of the implementation. Our conclusions are presented in Section 5. Finally, Section 6 outlines the future work.

2 Related work

Several research works on surveillance systems have been proposed. Some of the most important are described in the following.

2.1 Image based surveillance systems

In [3] a server with an embedded webcam capture images continuously of the desired location and stores them. Then a client login to the URL of the server can fetch the images from his phone using the HTTP protocol (Figure1). Although simple, the system only manages images but not video nor audio. A variant of this scheme [4] employs IP cameras, storing the images in the cloud and accessing it through a peer-to-peer network.

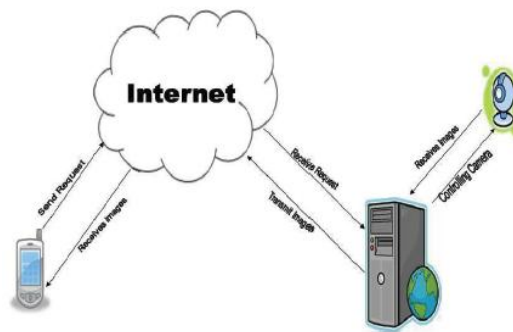


Fig. 1. An image based surveillance system.

In [5] the capture of the images is accompanied with an e-mail or sms text message alert to inform the presence of an intruder, while that in [6] the surveillance system adds the transmission of live or recorded video to the smartphone.

2.2 Video surveillance systems based on Wi-Fi and 3G wireless networks

Wi-Fi and 3G wireless networks feature faster data rates and higher stability of data links, allowing to develop video surveillance systems for mobile phones. In [7] a surveillance system is developed using a Wi-Fi network for transmitting data from each of the video data nodes to a video control server (Figure 2). This video server which has 3G Internet access, transmits the video data to a web site, from which the video data is stored in a video database and published to be accessed from 3G smartphones. A similar system [8] employs a WIMAX mobile network and SIP phones, but the system transmits live video.

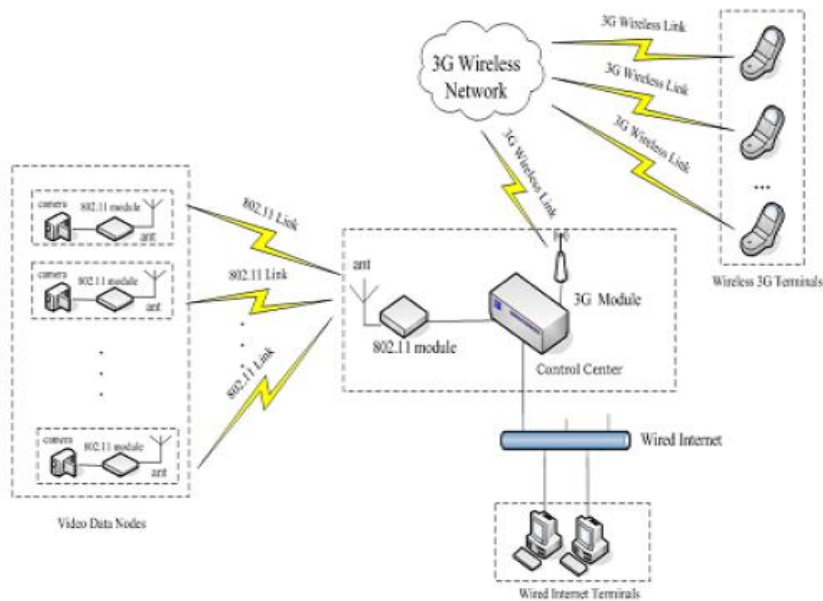


Fig. 2. A Wi-Fi and 3G based surveillance system.

2.3 Doorbell video surveillance systems

Currently there are several commercial video surveillance systems in the market focused specially to monitor visitors at the door of a home or premise. One of them is the MOBOTIX T24 [9], which is an IP door station based on the video telephony standard VoIP/SIP (Figure 3). When a visitor rings the doorbell, a connection is established with an IP video phone or a standard computer via the network to conduct a

video conversation from anywhere with the visitor at the door. DoorBot a similar product [10], is a Wi-Fi based doorbell that streams live video and audio of the front door directly to a smartphone or tablet (iOs and Android).



Fig. 3. The MOTOBOX VoIP/SIP door station.

CellNock is a Wi-Fi based video doorbell [11]. When a visitor pushes the doorbell a call alert is sent to a smartphone (Figure 4). The called party can accept or reject the call. Accepting the call allows to communicate with the visitor like a video conference call. If the call is rejected, the visitor has the option to record a video message which can be retrieved at convenience.



Fig. 4. CellNock doorbell.

2.4 App inventor for android (AIA)

AIA is a visual blocks language provided originally by Google and from January 2012, by the MIT Center for Mobile Learning. AIA graphical interface is very similar to the Scratch programming language, allowing users to drag-and-drop visual objects to create applications that run on many mobile phones with the Android OS. [12]. AIA has two main windows: a component designer for building the user interface and a blocks editor for defining the application behavior. Applications can be tested directly on the phone or an emulator (Figure 5).



Fig. 5. AIA Component Designer, Blocks Editor, Emulator and Phone.

After reviewing the above research and commercial works and taking advantage of the video camera and communication capabilities of Android phones, in the following sections we describe the design of GateWatcher an Android Phone based video surveillance system in which a video-doorbell is implemented using a cell phone and App Inventor. When a visitor rings the phone-based doorbell, a sms message notification is sent to a second phone. The receiver can accept this notification to watch the visitor returning another sms to the doorbell, which triggers its video-camera to start a Bambuser audio-video transmission to the Bambuser web site via a Wi-Fi or a 3G network. This web site is accessed automatically from the second phone to see and hear the visitor. The application also allows communicating with the visitor through chat instant messages.

3 Gatewatcher design

The app uses touch screen buttons, microphone and speaker to emulate the doorbell, thus as the video camera and sms notification capability of an Android phone (Figure 6). The phone uses also Bambuser a free live video broadcast service app [13], for streaming live video from mobile phones or PC-webcams to Internet via a Wi-Fi, or a 3G wireless network to the Bambuser web site, which can be accessed from a second mobile phone or PC with internet connection. The next section describes the design of the video doorbell using App Inventor.

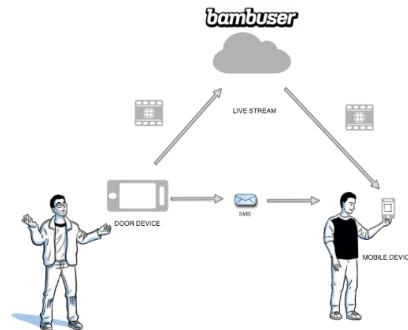


Fig. 6. GateWatcher architecture.

3.1 Building the gatewatcher GUI with the component designer, part I

The Component Designer shown on Figure 7 is the tool for designing the app interface. The left side palette has two types of components: 1) Visible like button, image, label, etc., and 2) Non-Visible like sound, texting, and ActivityStarter. We drag components from this palette to the viewer to specify the way the phone’s screen will appear when the application run, for saving data persistently, and for talking to the web. As a component is dragged into an app, its name and type appears in the list of the Components window. Components can be renamed, deleted, and another media added. When a component is selected, its properties that appear in the Properties window, can be modified. The visible and non visible components used are shown in Table 1.

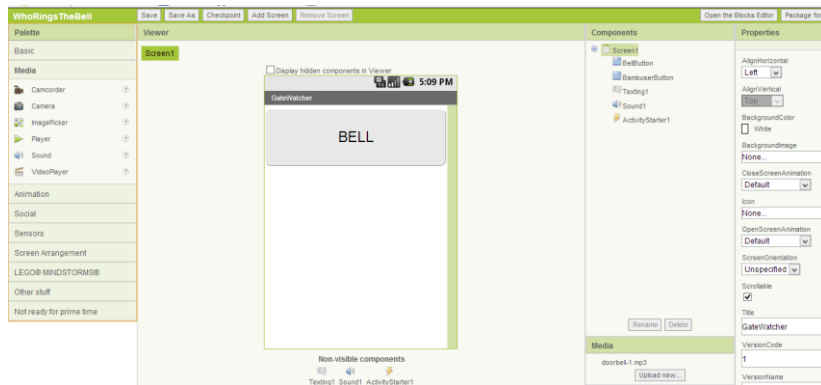


Fig. 7. AIA Component Designer for the GateWatcher application.

BellButton starts the application, produces a doorbell sound and sends a sms alert notification to the called party using the *Texting1* component. The *BambuserButton*, when touched, displays a message and launches the *Bambuser* app using the *ActivityStarter1* component (whose values were obtained using the Log Viewer app [14]).

Table 1. Visible and non visible components for the GateWatcher application.

Component type	Palette group	Component named as:	Purpose
HorizontalArrangement	Screen Arrangement	HorizontalArrangement1	A visual container
Button	Basic	BellButton	To ring the bell
Button	Basic	BambuserButton	To launch the Bambusser app
Sound	Media	BellSound	To sound the bell
Texting	Social	Texting1	To send and receive sms alerts
ActivityStarter	Other stuff	ActivityStarter1	To launch the Bambuser app

3.2 Programming the behavior of gateway with the blocks editor

The behavior of the application is defined in the Blocks Editor. This editor is launched clicking the “Open The Blocks Editor” button in the Component Designer. This Editor has two palettes from which blocks are dragged, the Built-in palette and the My Blocks palette (Figure 8).

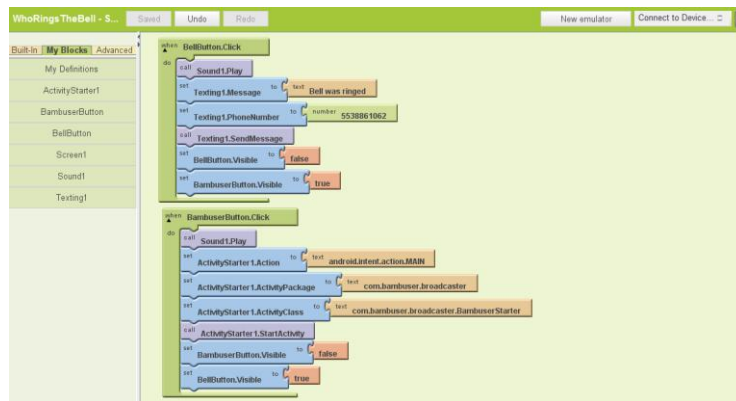


Fig. 8. The Blocks Editor with the My Blocks palette of GateWatcher.

The Built-in palette contains built-in blocks for standard programming control and functionality, for text and list manipulation, and mathematical, logical and control operators (Figure 8). The My Blocks palette contains blocks representing the components of the applications that were added in the Component Designer. In a very similar way like Scratch, the App Inventor blocks language provide visual cues to ease the

development tasks, and only some blocks lock in place, reducing the possibility of errors (Figure 8). The application behavior is directly defined through a set of event-handlers (e.g., “when event *BellButton.Click* occurs, do *callTexting1.SendMessage*”). Live testing can be performed with a Wi-Fi connected Android phone (using the MIT AICompanion application), clicking the *Connect to Device* button located in the upper right side. Once the application is tested, it can be deployed by packing it into an Android apk application by clicking the upper right *Package for Phone* button in the Component Designer (Figure 7).

3.3 Building the gatewatcher GUI with the component designer, part II

The second part of the GateWatcher application corresponds to the receiver. Here we want to receive an alert in our mobile phone when somebody knocks at the door of lets say, our home, and to decide if we want to do something about it, like acknowledging the alert notification sending an sms to the sender and accessing automatically the Bambuser web site to watch and hear the visitor and optionally to interact with he/she. In order to do this we used an application reported in [15], but modifying it with the ActivityStarter component used before in section 3.2 to access the Bambuser web site. The user interface and the blocks editor of this second part is shown in figure 9.

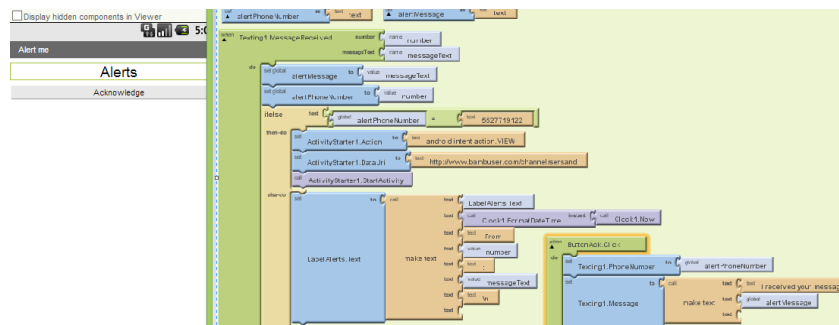


Fig. 9. GateWatcher receiver alert notification and access to the Bambuser web site.

4 Experiments and results

For the experiments we use as the video doorbell a Lanix Ilium S400 Android 4.0.2 phone with Internet connection via Wi-Fi, and the *Bambuser* live video broadcast application installed, thus as the *Log Viewer* app. This logger app was very useful to set the properties of the Activity Starter component to launch the Bambuser app. These properties were: the action name, the activity package name, the activity class name, thus as the Data URI of the Bambuser app as is shown in [16]. For the reception part, a LG P500 Android 2.2 phone with Internet connection via a 3G wireless network was used. The sms text messages used were standard. Google has a free sms service called *Google Voice*. Unfortunately, this service only works at the USA and

Canada. Figure 10 shows the results of the GateWatcher implementation. The left screen captures show the user interface with buttons to send the sms alerts and to launch the Bambuser live video streaming via Wi-Fi. The screen captures toward the right show the reception user interface, the sms alert notifications, and the access to the Bambuser web site to watch and hear the visitor at the door. The Bambuser web site allows the called party to communicate with the visitor through instant messages, but the visitor does not need to type anything, just talk (Figure 11). In this way the visitor cannot hear or see the recipient party.

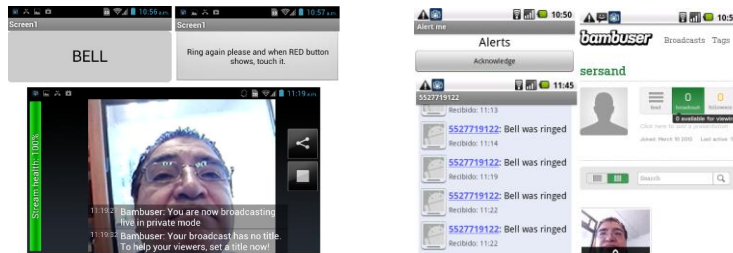


Fig. 10. Left: GateWatcher live video transmission; Right: Alerts & live video reception.



Fig. 11. GateWatcher: communicating with the visitor via Bambuser’s instant messages.

5 Conclusions

Surveillance systems are very important in our days for security reasons. In this work we design GateWatcher a video doorbell surveillance system using an Android phone with Wi-Fi internet connection, to monitor remotely the presence of a visitor when rings the bell of a home. The audio and video of the visitor was sent to the Bambuser web site from which can be accessed in real time using a second Android phone with Internet connection. The user interfaces of GateWatcher was eased using App Inventor with its intuitive drag-and-drop graphical interface, and its sms texting messages and Activity Starter components. This paper showed that it is not very difficult the implementation of a smart surveillance system with live video streaming using Android smartphones with sms alert message notifications.

6 Future work

This application may still be expanded beyond the actual state, making it more useful by including a log of doorbell ringing events, thus as the option that allow the visitor to leave a recorded video message that can be retrieved at convenience. The transmission of alert notifications by email or social networks would be convenient.

Acknowledgements

This work was supported by SIP-IPN project 20131854.

References

1. Scratch imagine program share, Scratch: a programming language for everyone. <http://scratch.mit.edu>, (2012).
2. Wikipedia, Google App Inventor, http://en.wikipedia.org/wiki/Google_App_Inventor.
3. Badrun Nahar and Md. Liakot Ali, Development of Mobile Phone Based Surveillance System, Proceedings of the 13th International Conference on Computer and Information Technology, pp 23-25, Dhaka, Bangladesh, (2010).
4. D. Shiny Irene and R. Dhanalakshmi, Video Surveillance System and Content Sharing Between PC And Mobile Using Android, <http://ieeexplore.ieee.org>, paper 6508392, (2013).
5. Rashmi, R., Latha, B.: Video Surveillance System And Facility To Access Pc From Remote Areas Using Smart Phone, <http://ieeexplore.ieee.org>, paper 06508393, (2012).
6. Estevez Ayres, I., Garcia Valle, M., Basanta Val P., Fernandez Pacheco, I.: Using Android Smartphones in a Service-Oriented Video Surveillance System, <http://ieeexplore.ieee.org>, paper 0572290, (2011).
7. Wang Liwei, Yan Shi and Xu Yiqiu, A Wireless Video Surveillance System based on 3G Network, 2nd Conf. Environmental Science & Information Application Technology, (2010).
8. LUO Yong-mei, WEI Bo, and JIN Zhi-gang, SIP-based WIMAX Wireless Video Surveillance System, 2012 International Conference on Computer Science and Service System.
9. Mobotix T24, “T24, Hemispheric VoIP/SIP video door station”, http://www.mobotix.com/eng_US/content/view/full/78917, (2013).
10. DoorBot, The Wi-Fi enabled Smart doorbell, <https://secure.christiestreet.com>, may 2013.
11. CellNock, Peerhole Cell Phone, <http://www.cellnock.com>, (2013).
12. MIT Center for Mobile Learning, App Inventor for Android. <http://appinventor.mit.edu>.
13. Bambuser Interactive live video broadcast service from mobile phones and web cameras to the Internet, <http://www.bambuser.com>, (2013).
14. Log Viewer, Real-time system log of Android, <https://play.google.com/store/apps/details?id=jp.itplus.android.log.viewer>, (2013).
15. MIT App Inventor Tutorials, AlertMe: Texting in the background when app is not running, <http://appinventor.mit.edu/explore/content/alertme.html>.
16. MIT App Inventor Explore, Using the Activity Starter, <http://appinventor.mit.edu/explore/content/Using-activity-starter.html>, (2013).