**Cloud and Smartphone-based Home Anti-Theft System**

**M.Eng Degree Design Project Report**

**School of Electrical and Computer Engineering**

**Cornell University**

**Xi He(xh243)**

**Yihuang Chen(yc993)**

**Project Advisor: \_\_\_\_\_\_\_\_\_David Albonesi\_\_\_\_\_\_\_\_\_\_\_**

**Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**1 Introduction**

In order to protect home security and guard people or house owner’s property, this anti-theft system is proposed and implemented to achieve the goal. While people is away from home, if the user can watch what’s happening at home through a phone app or be alerted automatically when crime is detected, it would be very handy. If the user can guard his home by himself, it would be better than letting other people guard his property. There are some companies such as Alarm.com, Inc doing similar projects and products to implement the features to deal with the home security issues.

This system prototype is based on the client-server architecture and involves three major components: a smartphone, a computer server, and a security camera. Client/Server communication is based on Google Cloud Messaging, wifi, TCP/IP, UDP and real time stream protocol RTSP.

*Smartphone:*

The smartphone app has three functionalities: 1. It controls the system on/off. When the app user leaves the house, he can turn on the system. He will view the real-time video of her house at any time and receive the alert image at anywhere at anytime. 2. It receives the captured image every time a moving object goes into the camera scope. This image is tailored to fit the Google Cloud Messaging format. The smartphone is able to reorganize the image data chunks and displays it on the screen. 3. The user can watch the real time stream video.

*Computer Server:*

The computer in the server side controls security cameras and detects moving objects in the camera scope by the computer vision algorithm. It is also responsible for sending real time video stream to the client device(smart phone) and alert the user through Google Cloud Messaging service if suspected object is detected.

*Cameras:*

In this project prototype, the secure camera is the webcam on the computer. It is used to capture the image inside the house.

# 

# 

# **2 System Design**

The system consists of a server, a webcam and a client. The server is a computer sitting at home while the client is an android phone app that the user holds to accept notification and control the server remotely.

## 2.1 Server

The home server is running all the time. Its job includes 1) Running computer vision algorithm to detect moving objects, 2) Sending the captured image to the phone app through Google Cloud Messaging. 3) Take various remote control requests from the user to perform corresponding tasks such as turning the system on or off, clearing the current warning, sending the real-time monitoring video stream and recording the live video.

## 2.1.1 Server FSM

The server is running the code all the time, but it has a three-stage finite state machine:

1. **Off**: In Off mode, the server is in sleep and is not doing anything but waiting for signal from user to start the system (start all the features on the server side).
2. **On**: When the system is on, all the feature including computer vision detection, image sending and real-time video monitoring are turned on so the home is under protection.
3. **Wait**: Right now, the system is already on but has detected a suspected scene and has sent an alert/image to notify the user. In Wait mode, the user needs to take a look at the notification and manually clear the warning or dial 911. If clearing the warning by clicking ok, the system will then go back to On mode. Before the user clears the warning, the system won’t send suspected images anymore.

## 2.1.2 Motion Detection Algorithm

The computer vision algorithm is an object detection algorithm which detects moving objects in the image. It checks the picture difference every frame and every 4 seconds to ensure that moving objects can always be detected. If the picture difference is beyond a certain threshold, it should be recognized as a moving object.

## 2.1.3 Multi-Thread and Synchronization

The server is a standby system and runs a few threads. No manual operation at the home server needs to be performed once the program is running. The system is robust enough from system crashes. It is able to be remotely controlled and can handle all the requests from the android app. There are several threads dealing with the server tasks and the client requests. These threads run in a synchronized manner and perform interrelated tasks without interference with each other. These threads are:

1. System Control Thread

This thread is the central control thread which decides if the system needs to shut down or turn up. Besides that, the thread takes request for querying system state(on/off) and it can clear the system warning. It listens request from the client and performs corresponding tasks. When it receives turning on/off signal from the client, it can notify other threads to stop to achieve safe stop, start or clearing warning. This thread runs all the time as the whole program is running and it communicates with the client through TCP/IP.

1. Computer Vision Detection Thread

This thread is running when system is started and under protection. It detects moving object and records entire video clip. It also saves the crime image on local disk if suspected scene is detected when there is no pre-exist warning. (Warning must be cleared before saving the next image for sending to client. The network traffic is reduced in this way).

1. Scene Check Thread

This thread is to take the saved crime image and send to the user through Google Cloud Messaging when there is an available image.

1. Real-Time Video Stream Monitoring Server Thread

This thread is a sub-server in the system which listens to incoming requests from the app user. It opens the real-time video streaming when receiving the video request. The thread has two socket connections: one is the RTP connection, which is the channel for the live video stream over UDP; the other one is RTSP connection, which deals with the control message over TCP. It sends RTP video packets over UDP and communicates with the client through RTSP. When the video stream is open, it will wait for further RTSP request from the client and follow the command accordingly. A simple protocol here is implemented to ensure such scheme from system crash. The RTP packet format is based on programming assignment 5 in CMSC 691C, University of Maryland Baltimore County.

## 

## 2.2 Client

## The smartphone plays the role of a remote controller, an alert image receiver and a live video player. 1) The user is able to query the system status and turn the system on/off at any time. 2) The smartphone has a background thread to receive the image data chunks. The image is showed on the screen after combining all the data chunks together. 3) The app is able to receive the live video and display it with high quality.

### 2.2.1 System Status Control

When the user wants to leave home, in order to get the alert image and live video, the remote server system has to be turned on beforehand. After the user clicks the “turn on system” button, a status query message is sent out via a socket connection on TCP port 6789. The server will respond the client with the system status. 1) If the system is off right now, the app will send out the turn on message to the server to let the system get started. 2) If the system is already on right now, the app will make no further action.

When the user arrives at home, she can turn off the system service. Similarly, after the user clicks the “turn off system” button, a status query message is sent out via the socket on TCP port 6789. The server will respond the client with the system status. 1) If the system is on right now, the app will send out the turn off message to the server to let the system go sleep. 2) If the system is already off right now, the app will make no further action.

Every time the client wants to send out the control message, the a socket is created on TCP port 6789. After sending out the message and receiving the response, the socket is closed.

### 2.2.2 Alert Image via GCM

Every time the computer vision algorithm detects a moving object, the server will send out the captured image to the client via Google Cloud Messaging(GCM). GCM service has a 4KB data restriction on every message. So, the image has to be splitted into several 4KB data chunks. The client application will receive all of the data chunks and notify the user(i.e., shaking the phone and making the sound). It then combines them into a whole picture. The user can see what is happening at home.

### 2.2.3 Live Video via RTSP

In the video display activity, a RTSP messaging socket on TCP port 8554 and a RTP video transmission socket on UDP port 25000 are created. These two socket connections will not be destroyed until the user exits this activity.

When the user wants to watch the video, a setup message is sent to the server. Then, the server will respond with an ACK. Next the user will send out a begin command to the server. The server will respond with the video stream. Actually, the video is a flow of images. The app receives the images every 1/50 seconds and draws them on the screen. We have tried the experiment on the relation between the image data rate and the video quality. Basically, there is a tradeoff: If you want higher video quality, you need to send the image as frequently as possible if the network bandwidth is unlimited. However, in reality the network doesn’t have enough bandwidth to sustain heavy traffic, the video quality will be affected dramatically by the image lost. 50 images per second is a good image rate to relieve the bandwidth and offer a fluent video flow. Now the user is able to view the live video and knows what is happening at home.

# **3 System Workflow**

## 3.1 Server

The followings are displayed when the system is started by remotely control from the android phone. Once system is started. It initializes all the functional threads and starts detecting algorithm.

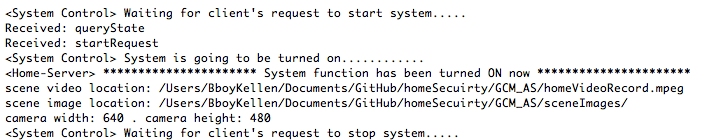


Figure 1: System is turned on

……

Then the followings are the displayed when the system is stopped by remote control from the android phone. It safely stops all the threads first before telling android phone that the system is shut off now

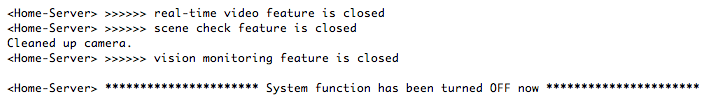


Figure 2: System is turned off

## 3.2 Client

# 

# Figure 3 shows the main page of the android application. When the user leaves home, he can click the “Turn System On” button to let the system work. The remote server will receive the signal and start to work. If the system is already on, he can click “View Video”.



Figure 3: Android application’s main page

After clicking the “View Video” button, the server starts to send the video. The user is able to see what is happening at home. Figure 4 is the video displayed on the screen.



Figure 4: View Video

When the remote server turns on, the computer vision algorithm starts running. It continuously detects the moving object. When there is a stranger coming into the house, the server will send a picture to the client through GCM(Google Cloud Messaging). The android phone receives the image and makes the notification as Figure 5 shows.

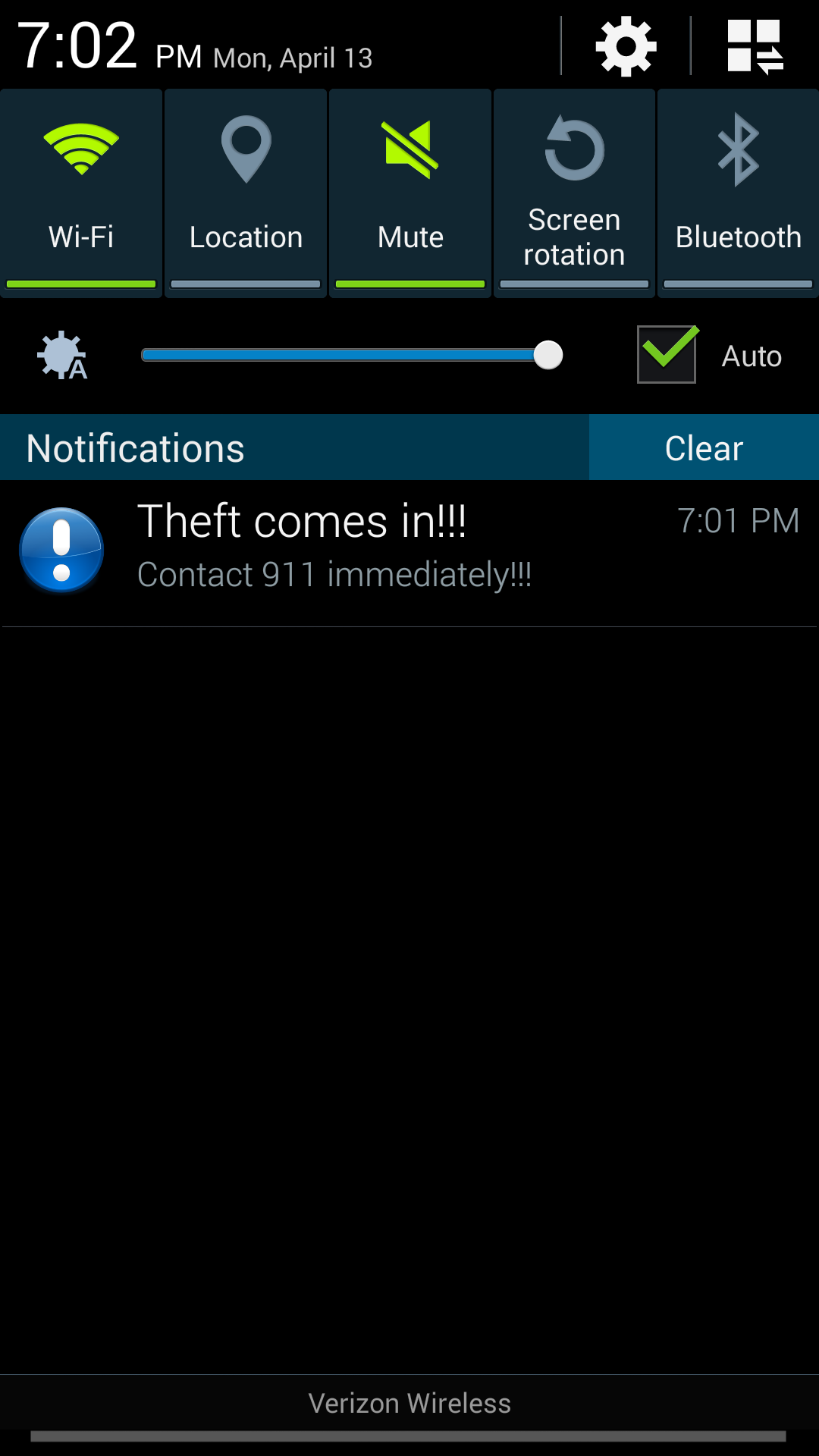


Figure 5: Notification when receiving the detected image

After clicking on the notification flag, the detected image is displayed on the screen. So, the user can dial 911 immediately.

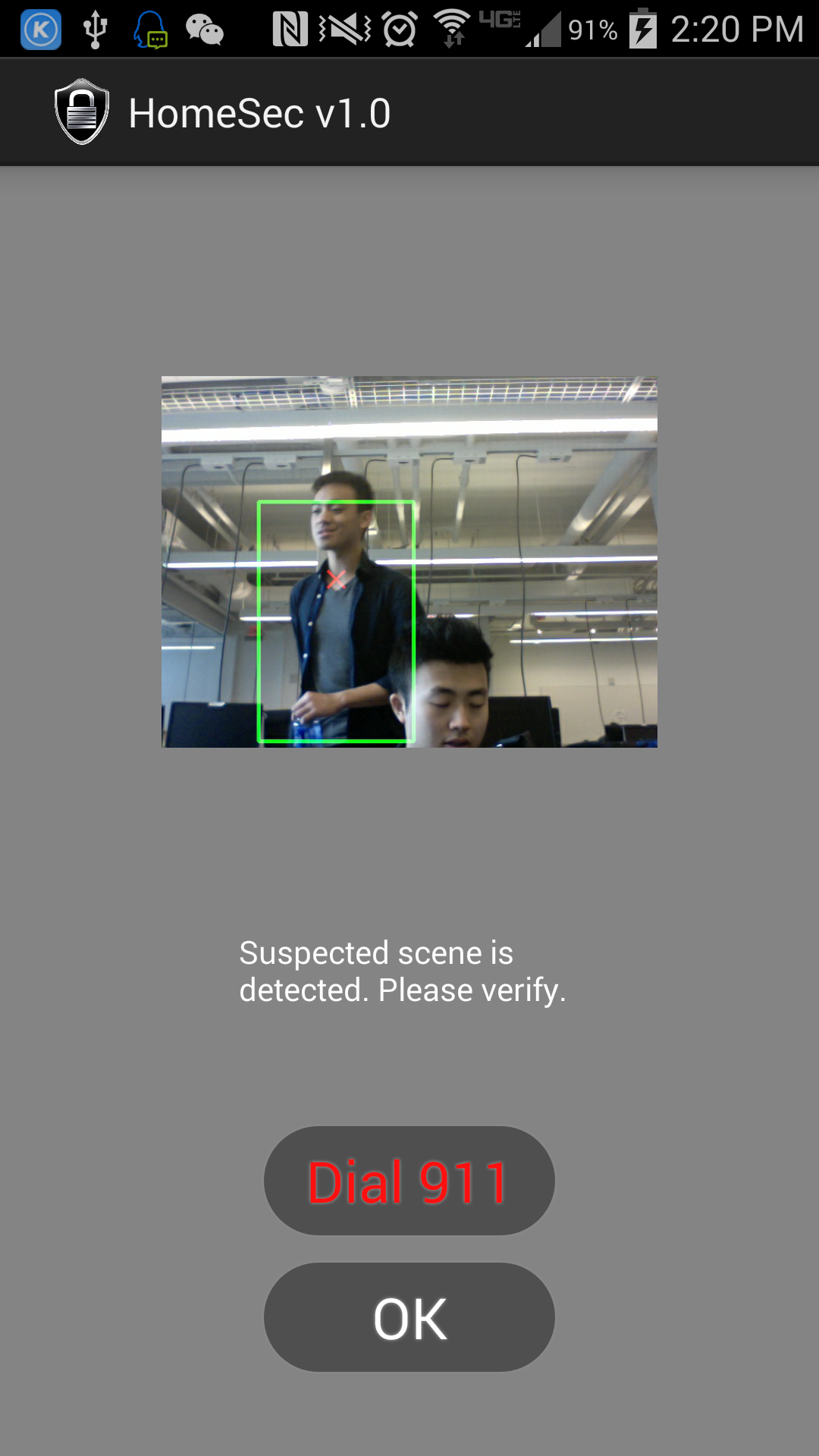


Figure 6: The received the detected image of a moving body

# **4 Conclusion**

## 4.1 System Functionality

This project offers a handy way for the user to know what is happening at home. It embraces many technologies to achieve the theft detection goal. 1) The computer vision algorithm is utilized to capture the appearance of the moving body. 2) The network communication is introduced to let the user see what is going on at home. Google Cloud Messaging is used for sending the alert image and the real time stream video is sent to the client over UDP. The socket timeout issues are carefully addressed in this project to make the program robust. 3) The synchronization issues are dealt at the server side to let the system run in order. 4) The smartphone development makes the application convenient to use.

## 4.2 Future Work

The current work done is a prototype of the system architecture. In order to achieve WAN communication so that user can watch video anywhere, the home server code will be put up on a private cloud on the public network so the smartphone and server on the cloud can communicate all the time without any boundary. A wireless camera will need to be purchased or designed to replace the webcam and it will be put at home and keep sending the frames in real-time to the server (in the cloud). The server in the cloud then does all the computation and proceeds the actions when interacting with the smartphone.

# **5 Reference**

[1] http://en.wikipedia.org/wiki/Real-time\_Transport\_Protocol

[2] http://www.csee.umbc.edu/~pmundur/courses/CMSC691C/lab5-kurose-ross.html

[3] http://en.wikipedia.org/wiki/Real\_Time\_Streaming\_Protocol

[4] http://hmkcode.com/android-google-cloud-messaging-tutorial/

[5] https://www.alarm.com/