# Android Interface Based GCM Home Security System Using Object Motion Detection

[1]Shalmika. R, [2]Surya. R, [3]Amirthavalli. R
[1,2] B.E CSE Final Year Student,[3]Assistant Professor,
Velammal Engineering College, Chennai, Tamil Nadu, India

***Abstract* - Nowadays public space is monitored by cameras which make up the Video surveillance system to reduce the crime and increase the security. Wide and complex areas are monitored by these cameras. To efficiently observe such a wide area at lower cost, mobile applications can be deployed. By analyzing the motion capture data from these cameras human action recognition can be performed. Mobile applications can remotely control home appliances. Likewise mobile applications can also also used to alert the user about any intrusion. This project proposes the Object Motion Detection Model that uses two algorithms viz. Seven Gray Scale Algorithm and Image Cauchy Distribution Model Algorithm. Motion is detected when there is any change in the pixel values. The detected motion is captured and saved on the server and the server will notify a GCM alert. The Android application will receive the GCM notification based on an id generated for every device that is registered. The detected image can be viewed using the URL received through the GCM notification. Many automated systems have been developed which informs the user in a remote location about any intrusion or an attempt to intrude in the house.**

***Keywords* - surveillance; mobile robots; android; sensor; intruder; motion capture**

## I. INTRODUCTION

Surveillance is the monitoring of the behavior, activities, or other changing information, usually of people for the purpose of influencing, managing, directing, or protecting. Surveillance is therefore an ambiguous practice, sometimes creating positive effects, at other times negative. It is sometimes done in a surreptitious manner. It most usually refers to observation of individuals or groups by government organizations, but disease surveillance, for example, is monitoring the progress of a disease in a community.

The word surveillance is the French word for "watching over"; "sur" means "from above" and "veiller" means "to watch". The inverse or reciprocal of surveillance is to watch from below.

The word surveillance may be applied to observation from a distance by means of electronic equipment, or interception of electronically transmitted information. It may also refer to simple, relatively no- or low-technology methods such as human intelligence agents and postal interception.

## II. SYSTEM ANALYSIS AND DESIGN

The existing methodology is a switch that is attached to the door. It detects any intrusion attempted by intruders. The image of the intruder that is detected is then stored in the server and can be retrieved after some time. This interrupts the GSM modem which sends a pre-configured warning SMS to the mobile phone in the remote location. Moreover there is no alert system to inform the admin when an unknown object is detected. If the user acknowledges the pop-up, immediately a message is sent back to the remote modem.

The existing system has many disadvantages adding to the inconvenience and inefficiency. The main problem is that there is no accuracy in the captured image. Moreover the moving object cannot be detected correctly. There is no SMS alert about the motion detection to the user and the image cannot be retrieved at the time of motion detection.

## III. PROPOSED MODEL

In the proposed system, the moving object is identified using the Image Cauchy Distribution Model method. The previous frame is compared with the current frame. From that the moving object is identified. Hence we can detect the exact image of the moving object. Controlling home appliances remotely with mobile applications have started becoming quite popular due to the exponential rise in the use of mobile devices. Another advantage of this system is when the threshold value reaches the limit then the server has detected a motion. Then the system will then alert the user automatically by sending a GCM alert to user's mobile application. The architecture diagram of the proposed system is as shown in Fig.1 below.

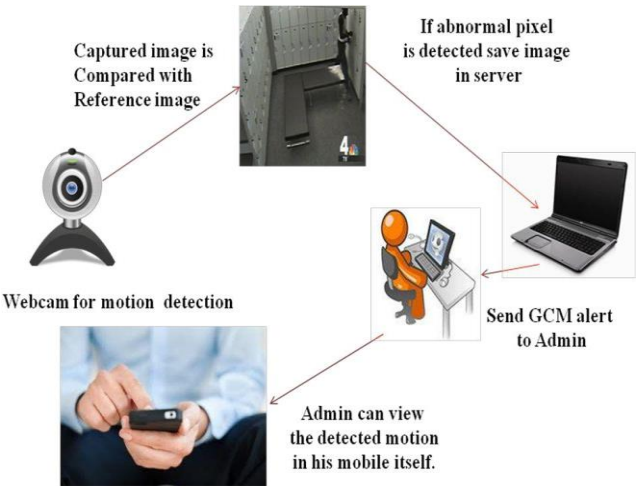


Fig.1 Architecture Diagram

## IV. GOOGLE CLOUD MESSAGING

Google Cloud Messaging (GCM) is a service that helps developers to send data from servers to their Android applications on Android devices, or from servers to their Chrome apps and extensions. The Android service was first unveiled on June 27, 2012, at Google I/O 2012 held at the Moscone Center in San Francisco. The Chrome service was announced before Google I/O 2013 as a blog post titled 'Building efficient apps and extensions with push messaging.

GCM replaces the beta version of Android Cloud to Device Messaging Service. This free service has the ability to send a lightweight message informing the Android application of new data to be fetched from the server. Larger messages can be sent with up to 4 KB of payload data.

Google Cloud Messaging for Android (GCM) is a free service that helps developers sends data from servers to their Android applications on Android devices, and upstream messages from the user's device back to the cloud. This could be a lightweight message telling the Android application that there is new data to be fetched from the server (for instance, a "new email" notification informing the application that it is out of sync with the back end), or it could be a message containing up to 4kb of payload data (so apps like instant messaging can consume the message directly).

The GCM service handles all aspects of queuing of messages and delivery to the target Android application running on the target device.

The primary characteristics of Google Cloud Messaging (GCM):

1) It allows 3rd-party application servers to send messages to their Android applications.
2) Using the GCM Cloud Connection Server, you can receive upstream messages from the user's device.
3) An Android application on an Android device doesn't need to be running to receive messages. The system will wake up the Android application via Intent broadcast when the message arrives, as long as the application is set up with the proper broadcast receiver and permissions.
4) It does not provide any built-in user interface message data. GCM simply passes raw message data received straight to the Android application, which has full control of how to handle it. For example, the application might post a notification, display a custom user interface, or silently sync data.
5) It requires devices running Android 2.2 or higher that also have the Google Play Store application installed, or an emulator running Android 2.2 with Google APIs. However, we are not limited to deploying our Android applications through Google Play Store.

It uses an existing connection for Google services. For pre-3.0 devices, this requires users to set up their Google account on their mobile devices. A Google account is not a requirement on devices running Android 4.0.4 or higher.

## V. SYSTEM IMPLEMENTATION

***Detecting Image Using Cauchy Distribution Model***

This module is to detect the motion in the particular area. The motion detection is done using Cauchy Distribution Model and Absolute Differential Estimation. Absolute Differential Estimation is used to compare the background frame and incoming video frame if any changes occur in incoming video frame. Cauchy distribution Model is used to detect the pixel of moving object in the detected incoming video frame.

A. Sending GCM Alert

Whenever motion is detected the image is saved on the server and the server will notify the Google server. The Google server will send a GCM alert to the users' android mobile who are all registered for that application. Google Cloud Messaging for Android (GCM) is a service that allows you to send data from our server to the user's Android-powered device. This could be a lightweight message telling the application that there is new data to be fetched from the server (for instance, a movie uploaded by a friend), or it could be a message containing up to 4kb of payload data (so applications like instant messaging can consume the message directly). Fig.2 shows the connection between the third party server and the GCM server which then sends the alert message to the android mobile.

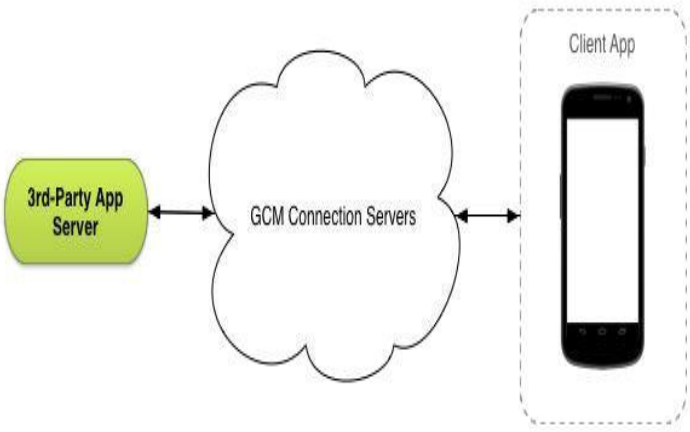


Fig.2 Sending GCM Alert
This is how these components interact:

1) Google-provided GCM Connection Servers take messages from a 3rd-party application server and send these messages to a GCM-enabled Android application (the "client app") running on a device. Currently Google provides connection servers for HTTP and XMPP.
2) The 3rd-Party Application Server is a component that you implement to work with your chosen GCM connection server(s). App servers send messages to a GCM connection server; the connection server queues and stores the message, and then sends it to the device when the device is online. For more information, see Implementing GCM Server.
3) The Client App is a GCM-enabled Android application running on a device. To receive GCM messages, this app must register with GCM and get a registration ID. If you are using the XMPP (CCS) connection server, the client app can send "upstream" messages back to the connection server. For more information on how to implement the client app, see Implementing GCM Client.

***User Authentication for Application***

User authentication is a means of identifying the user and verifying that the user is allowed to access some restricted services.

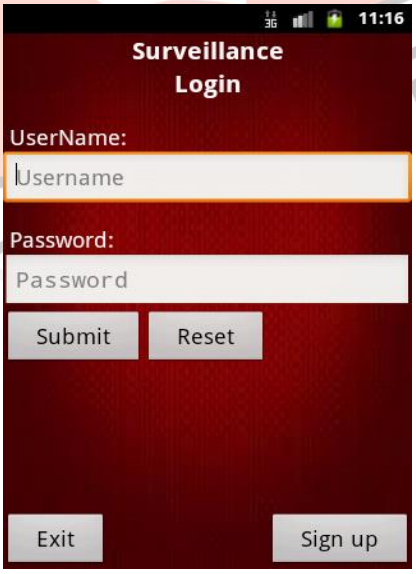


Fig.3 Login Screen

This module authenticates the user to view the motion detected image. This module includes username and password for authentication to the application. The validation is based on web service in server.

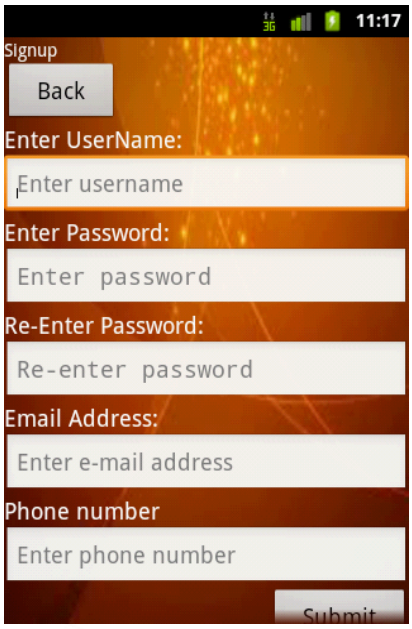


Fig.4 Sign Up Screen

***Viewing the Detected Image***

The Android application will receive the notification (GCM) based on project id which is

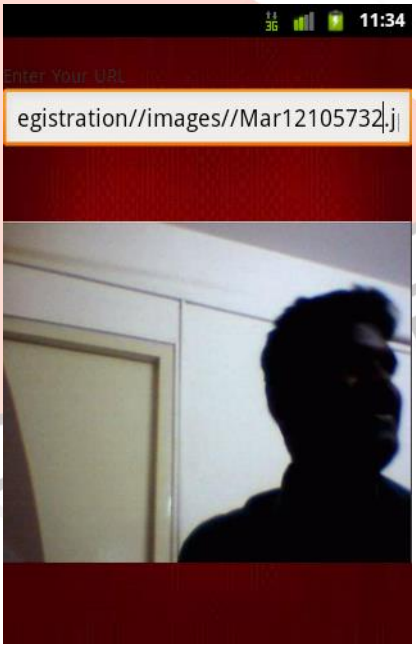


Fig.5 viewing detected image

## VI. CONCLUSION

This project introduced an approach for an effective video surveillance in the current system; this overcomes the traditional Surveying where Human intervention is needed and has to watch keenly for keeping track of the entire system. But now with this project we have introduced a unique technique which is a Major advantage to the old system.

This project also has a unique feature in which it sends GCM alert at once there is any sort of variation in the captured pixel. Also we are in intent to dedicate this project to many important Surveillance Areas so that Many Unwanted things can be prevented.

## VII. FUTURE SCOPE

Though this project has many added advantage, in future we like to upgrade this into the next level that is not only by just viewing the captured image, we can also view the entire clip of what happened and what has been captured. All this will be done just at the spontaneous moment, within seconds of the action been happened at the site.

**VIII. REFERENCES**

[1] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. CVPR*, 2008.
[2] J.Nielbes, H. Wang and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," in *IJCV*, vol. 79, pp. 299–318, 2008.
[3] P. Matikainen, M. Hebert, and R. Sukthankar, "Representing pairwise spatial and temporal relations for action recognition," in *Proc. ECCV*, 2010, vol. 6311, pp. 508–521.
[4] P. Natarajan and R. Nevatia, "View and scale invariant action recognition using multiview shape-flow models," in *Proc. CVPR*, 2008.
[5] P. Yan, S. M. Khan, and M. Shah, "Learning 4d action feature models for arbitary view action recognition," in *Proc. CVPR*, 2008.
[6] H. Ning, W. Xu, Y. Gong, and T. Huang, "Latent pose estimator for continuous action recognition," in *Proc. ECCV*, 2008, vol. 5303, pp. 419–433.
[7] J. C. Niebles, C.W. Chen and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *Proceedings of the 12th European Conference of Computer Vision (ECCV)*, Crete, Greece, September 2010.
[8] T. Hofmann, B. Scholkopf, and A. J. Smola, "Kernel methods in machine learning," *Annals of Statistics*, vol. 36, pp. 1171–1220, 2008.
[9] Android Programming and Book! Release 002 by **Nicolas Gramlich** Pg 5 to 60.
[10] Professional Android Application Development by Perth, Western Australia and Reto Meier in London.
[11] Android Open Source Project. Open Handset Alliance: http://source.android.com/license
[12] Dalvik Debug Bridge Monitor Service: http://developer.android.com/guide/developing/tools/ddms.html
[13] Official Android website: http://www.android.com/ Open Handset Alliance, website: http://www.openhandsetalliance.com/
[14] Java Virtual Machine Tool Interface Spec: http://java.sun.com/javase/6/docs/platform/jvmti/jvmti.html