IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 1, March, 2013
ISSN: 2320 - 8791
www.ijreat.org

Effective and Secure Scheme for IP-Camera Surveillance Based on Smartphone

Gowthamkumar Shanker[1], Bhuvanesh Krishnasamy[2] and Dr. Chitra [3]

[1,2&3] Department of Information Technology, Sona College of Technology, Salem, Tamil Nadu, India

Abstract

This paper presents system to improve real-time monitoring applications based on the usage of the service oriented system, with android Smartphone's as end user, allowing application to increasing the flexibility of the system. Video Surveillance systems have increase their needs of dynamism in order to allow the different users (operators and administrators) to monitor the system selecting different QoS depending on the system status and to access live and recorded video from different localizations, We present a QoS-aware service-based architecture for surveillance systems, and a prototype of this architecture, where a video surveillance application is developed over the Android platform.

Keywords: *IPcamera, Android, RTSP, live streaming.*

1. Introduction

Video Surveillance systems have increase their needs of dynamism in order to allow the different users (operators and administrators) to monitor the system selecting different QoS depending on the system status and to access live and recorded video from different localizations, for example, from their mobile devices. More concretely, in IP surveillance systems some resources involved are limited or expensive so dynamic reconfiguration could become competitive advantage for system integrator and designers able to offer flexible applications adaptable to users' needs.

Nowadays, it is very common for people to have mobile phones with an integrated digital camera. This gives the opportunity to develop a proposal for face Detection system on these phones. We have to feed the faces of people who we are authorized to our home in a database, which is used by some algorithm to detect those

faces and it is developed as application in android. The captured face is stored in cloud.

Advances in programming paradigms have allowed increasing the dynamism and flexibility of distributed environments. Service-Oriented approaches provide means of developing decoupled applications in heterogeneous networks by defining the concept of service. A service, in the SOA context, is an entity that receives and sends messages through well-defined interfaces, allowing building more complex applications that increase the value of the system. This concept can be applied to QoS-aware (Quality of Service) systems, in order to ease the configuration and reconfiguration of applications Besides, Android is a software stack for mobile devices that includes an operating system, middleware and applications that can be suitable for the development of the end-user surveillance application. In this paper, we present a QoS-aware service-based architecture for surveillance systems, and a prototype of this architecture, where a video surveillance application is developed over the Android platform.

2. Live Streaming

The monitoring system will have the advantage of different users to monitor the system, all users can access live and record video from different localizations, high resolution and it's a Dynamic system. The control system will have the additional advantage of controlling the direction of rotation of the IP camera, increasing brightness of the video and flip video image.

The server finds a video device and captures images. Then, it generates captured data and transforms the data format for RTP communication. After that, it generates the session manager and streams media data to the client. A client using the web or a Smartphone generates the session manager and connects to the server. Then, it receives media data and displays the video. In this way, the client can monitor the in-home or in-business location status in real-time. The architecture of the Smartphone is built upon the Android architecture. The Smartphone are able of sending requests to the server architecture to change the system configuration. The smart phones can also accept commands from the server.

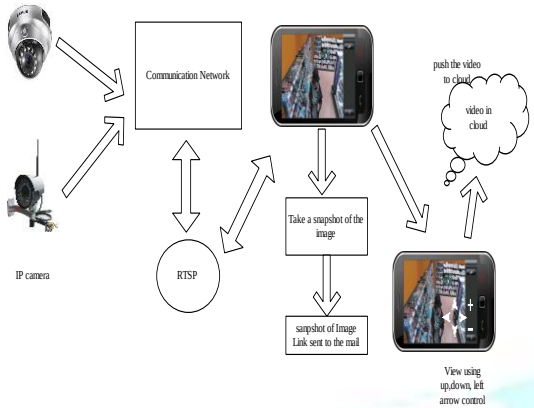

Fig. 1. Architecture Diagram

2.1 Secure Module

It provides security for video surveillance. Valid user-name and password should provided by the end user to access the application. User-name and password given by the end-user is comparing with Database which is hold user information on end-user Smartphone.

A user can log in to an Application to obtain access and can then log out or log off when the access is no longer needed. To log out is to close off one's access to a computer system after having previously logged in.

2.2 Location of Camera

The camera location is set by user who is going to use the application. The application deployment may be depends on their need i.e. in office, Home, college, schools. In this module, there are three location will be added. By using this module, user can choose the location which one the user has to see. For example, the user will select office location, it transmit to that appropriate locations. If the user will use wired medium to transmit the data location should be in 100 meter distance.

2.3 Surveillance Camera

Video surveillance cameras are becoming more popular and, most importantly, more accessible to the average person. Huge store chains or gated communities aren't the only places than can afford the benefits of surveillance cameras anymore. They are available for your home and office needs, no matter how simple or how elaborate a setup you require.

IP (Internet protocol) camera is used as surveillance camera in our project. IPCAM is an integrated wireless IP Camera solution. It combines a high quality digital video Camera with network connectivity and a powerful web server to bring clear to our mobile from anywhere on our local network or over the Internet. The basic function of IPCAM is transmitting remote video on the IP network. The high quality video image can be transmitted with 30fps (frame rate per second) speed on the LAN/WAN by using MJPEG hardware compression technology.

2.4 Video Streaming

Video streaming sends both live and recorded video to a user terminal. Videos stored in web server will be viewed by user on their android mobile. RTSP (Real Time Streaming Protocol) is used to view the videos in mobile which is stored in web server.

Table 1. RTP Payload Type

Payload Type	Name	Type	Description
14	MPA	Audio	MPEG-1 or MPEG-2 Audio Only
26	MJPEG	video	JPEG or MJPEG Video
31	H.261	Video	ITU-T H.261 Video
32	MPV	Video	MPEG-1 and MPEG-2 Video
33	MP2T	Audio/video	MPEG-2 transport stream Video
34	H.263	Video	H.263 video, first version

2.5 Face Recognition

A facial recognition system is a computer application for automatically identifying or verifying a person from digital image or a video frame from a video source. One of the ways to do this is by comparing selected facial features from the image and a facial database. It is typically used in security systems and can be compared to their biometrics. The detected face is stored in our database. The stored image is taken from cloud. Then it is processed with face recognition technique. The detected image is compared with our database images. If any person images are not matched with our database means, it will give an error message for that unauthorized person.

3. Moving Object Detection

Moving object detection is a process of confirming a change in position of an object relative to its surroundings or the change in the surroundings relative to an object.

This detection can be achieved by both mechanical and electronic methods. In addition to discrete, on or off motion detection, it can also consist of magnitude detection that can measure and quantify the strength or speed of this motion or the object that created it. Video tracking is the process of locating a moving object (or multiple objects) over time using a camera. It has a variety of uses, some of which are: human-computer interaction, security and surveillance.

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans). If the moving object is detected in camera, sms alert will sent to the user.

3.1 Basic Background Subtraction Algorithm

Surveillance and monitoring systems often require on line segmentation of all moving objects in a video sequence. Segmentation is a key step since it influences the performance of the other modules, e.g., object tracking, classification or recognition. Basic background subtraction (**BBS)** algorithm computes the absolute difference between the current image and a static background image and compares each pixel to a threshold. All the connected components are computed and they are considered as active regions if their area exceeds a given threshold. This is perhaps the simplest object detection algorithm one can imagine. The method in our implementation assumes that each pixel of the background is a realization of a random variable with Gaussian distribution.

Algorithm

```
protected static boolean isDifferent(int[] first, int width, int
height) {
if (first==null) throw new NullPointerException();
if (mPrevious==null) return false;
if (first.length != mPrevious.length) return true;
if (mPreviousWidth != width || mPreviousHeight != height)
return true;
int totDifferentPixels = 0;
for (int i = 0, ij=0; i < height; i++) {
for (int j = 0; j < width; j++, ij++) {
int pix = (0xff & ((int)first[ij]));
int otherPix = (0xff & ((int)mPrevious[ij]));
//Catch any pixels that are out of range
if (pix < 0) pix = 0;
if (pix > 255) pix = 255;
if (otherPix < 0) otherPix = 0;
if (otherPix > 255) otherPix = 255;
if (Math.abs(pix - otherPix) >= mPixelThreshold) {
totDifferentPixels++;
//Paint different pixel red
first[ij] = Color.RED;
                                        }
                                }
                        }
```

3.2 Instance Based Learning Algorithm

To count the number the number of people in video. The first one is based on two thresholds which are related to the average width and to the average area of a blob top zone, which represents a person head. By matching the width and the head region area of a current blob against these thresholds it is possible to estimate if the blob encloses one, two or three persons. The second strategy is based on a zoning scheme and extracts low level features from the top region of the blob, which is also related to a person head. Such feature vectors are used together with an **instance-based classier** algorithm to estimate the number of persons enclosed by the blob.

3.3 Generate Alert MSG

In this Module used to send the alert message to the corresponding user, While Surveillance the remote area when motion is occurred it automatically send alert Message to the user. And store photo in local mobile SD card.

Algorithm

```
public void run()
 {
if (!processing.compareAndSet(false, true)
return;
 Log.i("JO", "BEGIN PROCESSING...");
try {
if (Preferences.USE_RGB) {
img = ImageProcessing.decodeYUV420SPtoRGB(data,
width,height);
} else {
img = ImageProcessing.decodeYUV420SPtoLuma(data,
width,height);
}
 long aConversion = System.currentTimeMillis();
 Log.i("JO", "Converstion=");
int[] org = null;
if (Preferences.SAVE_ORIGINAL && img != null)
org = img.clone();
if (bmp != null ) {
long now = System.currentTimeMillis();
```

```
if (now > (mReferenceTime +
Preferences.PICTURE_DELAY)) {
mReferenceTime = now;
Log.i("JO", "REFER:");
Bitmap previous = null;
if (Preferences.SAVE_PREVIOUS && pre != null) {
if (Preferences.USE_RGB)
previous = ImageProcessing.rgbToBitmap(pre,width,
height);
else
previous = ImageProcessing.lumaToGreyscale(pre,width,
height);
}
Bitmap original = null;
if (Preferences.SAVE_ORIGINAL && org != null) {
if (Preferences.USE_RGB)
original = ImageProcessing.rgbToBitmap(org,width,
height);
else
original = ImageProcessing.lumaToGreyscale(org,width,
height);
                    }
                }
            }
```

4. System Implementation

- The application will be targeted on a handheld device with android OS V3.5 or less.
- The application will be built using Eclipse with an Android Development Tools (ADT) plug in.

Emulators provided by the Android software development toolkit will be used to develop and test the application. Proper implementation is essential to provide a reliable system to meet the organization requirements. Successful implementation may not guarantee improvement in the organization using the new system, but improper installation will improve it. The implementation needs for easy and quick using and time saving programs. After the system is implemented and conversion is complete, a review should be conducted to determine whether the system is meeting expectations and where improvements are needed.

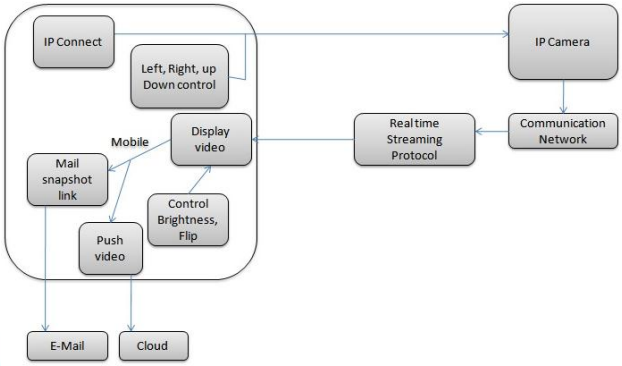

Fig. 2. Flow Diagram

4.1 Streaming and IP-Camera Controlling Algorithm

```
try {
db = SQLiteDatabase.openDatabase(
"/data/data/com.ipcam.stream/databases/ipcam.db", null,
SQLiteDatabase.OPEN_READONLY);
c = db.rawQuery("select * from tblCamera where
camera_desc = '"+ camera_desc + "'", null);
if (c.moveToFirst()) {
ip = c.getString(c.getColumnIndex("host"));
port = c.getString(c.getColumnIndex("port"));
user = c.getString
(c.getColumnIndex("username"));
pass = c.getString(c.getColumnIndex("password"));
Display display =
getWindowManager().getDefaultDisplay();
if (display.getWidth() < 600) {
// Low resolution image stream using the cgi link
cameraLink = ip + ":" + port + "/snapshot.cgi?user=" +
user+ "&pwd=" + pass + "&resolution=8";
}
Else
 {
// High resolution image stream using the cgi link
cameraLink = ip + ":" + port + "/snapshot.cgi?user=" +
user+ "&pwd=" + pass + "&resolution=32";
}
          }
c.close();
db.close();
}
 catch (SQLiteException e) {
System.out.println("SQLException: " + e);
}
```

IP-Camera Direction Control

```
left.setOnClickListener(new OnClickListener() {
public void onClick(View v) {
```

```
httpgetCall(ip + ":" + port +
"/moveptz.xml?dir=left&user=" + user+ "&password=" +
pass.trim());
httpgetCall(ip + ":" + port +
"/moveptz.xml?dir=stop&user=" + user+ "&password=" +
pass.trim());
                                    }                       });
right.setOnClickListener(new OnClickListener() {
public void onClick(View v) {
httpgetCall(ip + ":" + port +
"/moveptz.xml?dir=right&user=" + user+ "&password=" +
pass.trim());
httpgetCall(ip + ":" + port +
"/moveptz.xml?dir=stop&user=" + user+ "&password=" +
pass.trim());
                                    }                       });
top.setOnClickListener(new OnClickListener() {
public void onClick(View v) {
httpgetCall(ip + ":" + port +
"/moveptz.xml?dir=up&user=" + user+ "&password=" +
pass.trim());
httpgetCall(ip + ":" + port +
"/moveptz.xml?dir=stop&user=" + user+ "&password=" +
pass.trim());
                                    }           });
bottom.setOnClickListener(new OnClickListener() {
public void onClick(View v) {
httpgetCall(ip + ":" + port +
"/moveptz.xml?dir=down&user=" + user+ "&password="
+ pass.trim());
httpgetCall(ip + ":" + port +
"/moveptz.xml?dir=stop&user=" + user+ "&password=" +
pass.trim());
                                    }
            });
protected Integer doInBackground(Bitmap... data) {
for (int i = 0; i < data.length; i++) {
Bitmap bitmap = data[i];
String name =
String.valueOf(System.currentTimeMillis());
if (bitmap != null)
save(name, bitmap);}
return 1;
}
});
```

The above code is used to streaming and controlling the directions of the ip-camera.

4.1 Result

Fig. 3. Login

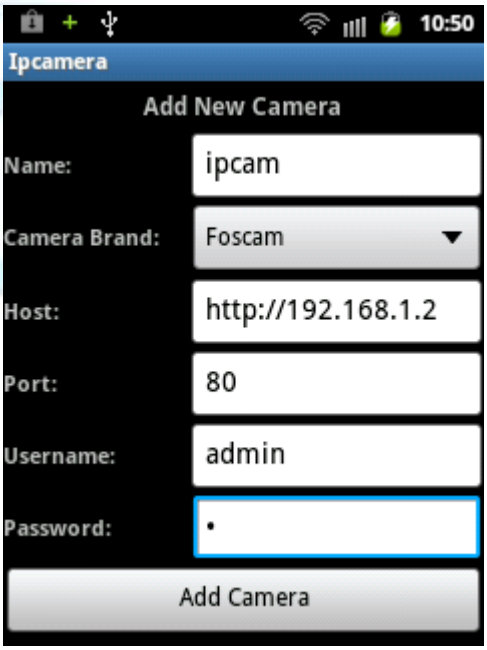

Fig. 4. Adding New IP-Camera

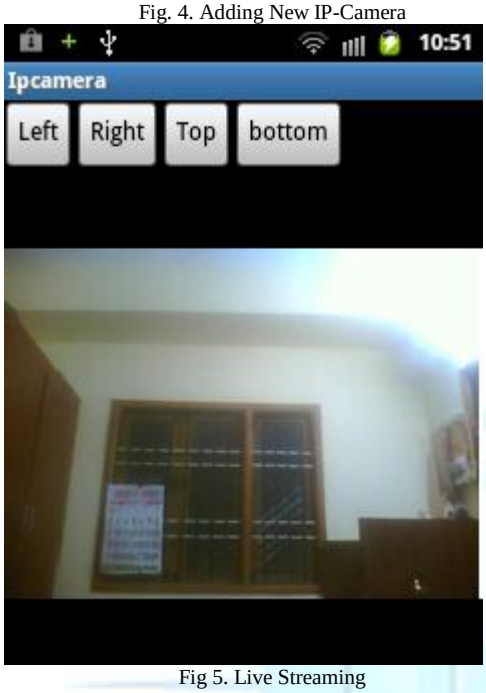

Fig 5. Live Streaming

4. Conclusion

Service-oriented paradigm can be successfully applied to surveillance systems, increasing their flexibility and dynamism, allowing the creating of applications of added value, such as the usage of Smartphone's as user terminals to control and watch over different areas. In this paper, a service oriented architecture for surveillance systems was proposed and a prototype of the system using an android terminal was described. The face is captured using mobile camera. The captured image is stored in cloud.

References

[1] I. Estévez-Ayres, P. Basanta-Val, M. García-Valls, J. A. Fisteus and L. Almeida, "QoS-aware Real-Time Composition Algorithms for Service-Based Applications", IEEE Trans. on Industrial Informatics, vol 5 (3) pp. 278-288, Aug. 2009.

[2] I. Estévez-Ayres, L. Almeida, M. García-Valls and P. Basanta-Val, "An Architecture to Support Dynamic Service Composition in Distributed Real-Time Systems", Proc of the 10th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC), May 2007. Santorini Island, Greece.

[3] OMG, "Data Distribution Service for Real-time systems". Object Management Group, 1.2 formal/07-01-01 edition, January 2007.

[4] Jianyu Dong "Efficient and Effective Streaming Technologies for 3-D Wavelet Compressed Video",the Ohio State University 2002.

[5] M D Walker, M Nilsson, T Jebb and R Turnbull "Mobile video-streaming", BT Technology Journal-Vol 21 No 3- July 2003.

[6] Jian Zhou, "New Techniques for Streaming MPEG Video over the Internet", UMI Microform 3111144, Copyright 2004 by ProQuest Information and Learning Company, pp. 11-26.

[7] Hatem BETTAHAR, "Tutorial on Multicast Video Streaming Techniques", SETIT 2005, 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications March 27- 31, 2005 –TUNISIA.

[8] Mamoona Asghar, Saima Sadaf , Kamran Eidi, Asia Naseem, Shahid Naweed "SVS - A Secure Scheme for Video Streaming Using SRTP AES and DH", European Journal of Scientific Research ISSN 1450-216X Vol.40 No.2 (2010), pp.177-188 ©EuroJournals Publishing, Inc. 2010.