Context-Aware Cell-Phone based Video Surveillance System using Proximity Algorithm

Basavaraj M. Hunshal
Asst. Prof., K.L.E. College of Engineering, Chikodi.
Dist: Belgaum (Karnataka, India)

Madhurani B. Shiddibhavi
Asst. Prof., K.L.E. College of Engineering, Chikodi.
Dist: Belgaum (Karnataka, India)

Abstract - Traditional Surveillance Systems (SS) are hard to move from one place to another place and also their development cost is high. Due to their high cost they are not reachable to the common people. So, it is required to implement least cost and a portable Protective system which can move from one place to another place easily.On other hand Surveillance Systems are becoming more popular to protect the valuable things. Now aday's cellphones (Mobile phones) are becoming very common among the peopleand their capabilities are also pretty high in concern to their Processing speed and storage capacity. By considering all these things, instead of using separate Processing and storage unit to implement surveillance systems it is better to use Mobile Phones.By this Surveillance system works with Wireless communication.This paper proposes a Surveillance Systemusing Mobile Phones and Proximity Algorithm.Proximity Algorithm is more effective and efficient algorithm which is used here to detect the threat in the context area and after detecting it sends one alert message to the owner as well as it streams video to the server. Entire SS is developed by using J2ME.

Key words: Surveillance systems, Context Awareness, Proximity Algorithm, Background subtraction

I. INTRODUCTION

The increasing need for intelligent video surveillance in public, commercial and family applications makes automated video surveillance systems one of the main current application domains in computer vision. Now a day, Mobile phones are becoming very common among the people and accordinglytheir processing and storage capacities are also becoming pretty good. Present day mobile phones are having speed of 500 MHz – 1.3 GHz and storage capacity of phone memory ranges from 16 MB to 128 MB and Secondarymemory card capacity is up-to 32 GB. So, by considering these all features it is very effective if technology uses the proficiencies of Mobile Phones. It becomes very helpful for the people and corporate people to manage and also protect their valuable things by least cost Surveillance System. [2]

Since proposed Surveillance system use Mobile phone (Cell Phone) and Internet, it becomes easy foruse and easy for maintain as it provides roaming facility. Now a day Internet is allowing us to send and receive types of data. Here in this GPRS technology is used to send contextual information in the form of video as well as it sends one text message to the owner's mobile phone. Video streaming and message sending is done from Mobile phone to the Server by using GPRS Technology. It is still effective and helpful if 3G is available for all the people. [3]

Low cost intelligent wireless video surveillance system is proposed in this paper. It uses the method of proximity algorithm to find out moving objects in the contextual area. It can also performindependent surveillance mission and can be extendedto a distributed surveillance system. [1]

II. PROPOSED ALGORITHM

1.1 Background Subtraction

It is an essential thing to detect object moving in the video. Video can be viewed as stream or sequence of frames. Therefore, creation of a background template is very necessary to detect the moving object from the video.[5][6] Here simple background subtraction method applied is subtracting the timely updated background template from the observed image and then Calculating the threshold value as result to generate the objects of interest. Background subtraction algorithm is as shown in the below figure – 1.

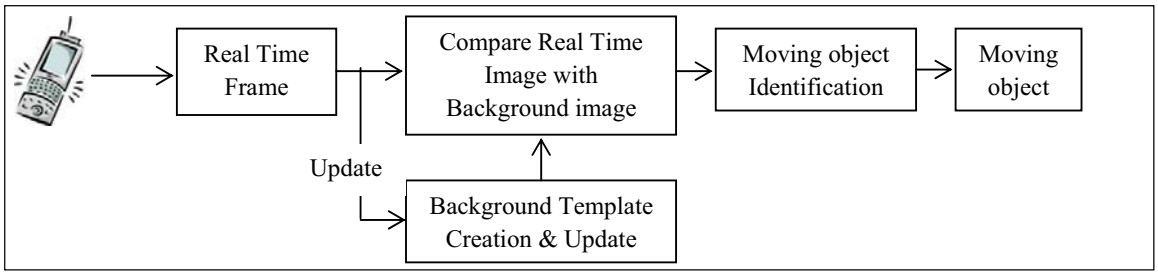

Figure1.Background Subtraction Algorithm

1.2 Background Template Creation

Before going to identify moving object from the scenario, first job is to create the background template,where expected object is not present. Background of the image means, background things left in the respective image except our needed object present in the image. Generally in an image both Background and as well as Foreground (expected object) items are present.

Therefore, during Background template creation time an average of Background of around 'n' images are been considered. Background scenarios may contain altering objects like leaf movement, and curtain movements etc., they are not considered as moving objects because moving object not stays in the same place. So, if you are constructing Background as an average of around 'n' images then it can solve the problem of altering objects.

An Average value of pixels means total number of pixels which are present in the same place of each frame.Below formula is used to calculate average pixels of 'n' frames,

$$\text{Background Template} = \frac{(\text{Frame}_1 + \text{Frame}_2 + \text{Frame}_3 + \ldots\ldots\ldots + \text{Frame}_n)}{n}$$

In our work, n=10, means first 10 frames average pixels are been considered for the above calculation.

1.3 Background template updating

Due to natural changes in the climate like darkness, more light background extraction may become problem. Updating the background is much necessary. To update background, instead of using separate background updating algorithm like "Moving Average algorithm" it is better to use simple method like as given in below algorithm,

For each pixel in Frame j:

If pixel_of_frame$_k$ is >pixel_of_background$_k$
Then,
{
Increase pixel_of_background$_k$by one
}
Else if *pixel_of_frame$_k$ Is* <pixel_of_background$_k$
Then,
{
Decrement pixel_of_background$_k$by one
}

Here, 'k' is kth-pixel of Current frame and Background template. By this method system automatically adjusts background template according to environmental changes.

1.4 Moving object recognition using Proximity Algorithm: [8]

1.4.1 Algorithm Explanation:

Step-1) The algorithm takes the reference snapshot (Background Template)
Step-2) Get the total pixels of the image,
(Width of the image * height of the image)

Step-3) Get the RGB value of the image.
Step-4) Get the average value of RGB for each pixel
Step-5) i) Create two parallel arrays;
1) First array is to store the color of each pixel;
2) Next array is to start the color count.
ii) Increment the color array count as and when you find a new color---------- (1)
Step-6) Increments the color count array as and when it finds the color which is already there in the color array.
Step-7) Repeat the same for the consecutive snaps/images you will get another color array count -------- (2) andcolor count array.

Subtraction Operation: (2) – (1) = result

If the result is greater than the threshold value (the value which we set) then it is considered to be motion detected.

Result > threshold then we can say MOTION DETECTED.

Explanation of Proximity Algorithm in detail:

intpixmityAlg (int image []) : Algorithm gets the Average of pixel's color in an Integer array as Input.
int LENGTH = image.length; : Get the length of the Array.

Create two Array of integer type with the length,

intcolor_array [] = new int [LENGTH]; : First one is to hold the color.
intcolor_count [] = new int [LENGTH]; : Second is to hold the color count.

Both are parallel array.

intarray_count=1; : A array_count variable is initialized with "1".

booleancolor_found=false;: intially color found is false

color_array [0] =image [0]; :"Color Array" will be loaded with a color from "Average Color Array" and its corresponding "Color Count Array" will be initialized as "1"

```
if (color_array[j] ==image[i])
{
        color_count[j]=color_count[j]+1;
        color_found=true;
        break;
}
```

"Average Color Array" will be iterated to find next colors,If same color is found again in the iteration, value of particular color index will be increment with "1" in "Color Count Array".[4] [8]

```
if(!color_found)
{
        array_count=array_count+1;
        color_array [array_count] =image[i];
        color_count [array_count] =1;
}
```

if the color_found is false(i.e., new color is found),
array_count=array_count+1;→color_count[array_count]=1; : Whenever new color is added to "Color Array", array_count will be increment with "1".

Hence count will vary according to the image variation.Thus the count is used for motion detection.

Initially the application stores the count of reference image.Then for all the consecutive images, application finds the count and calculates the difference.

If difference is greater than threshold value, motion is detected

1.4.2The engine has two steps of working-

- First the input image and the reference image (Background Template) is compared.

- Whenever a pixel has changed considerably (determined by the *threshold* variable) an internal black-white-red image is marked (at the same exact location where the change occurred). Therefore in
- The first step, the internal black-white-red image is has lit up clusters in the space where a change has occurred.
- The next step is to eliminate these clusters that are too small, but still appeared in our black-white-red image. Only the big clusters are left (and are colored red). During this process we keep a track of count of the big clusters.
- If the count is greater than *blob_threshold* then the input frame is determined to haveconsiderably motion as to the previous frame.
- Calculate the reference intensity of input-image.
- Normalize the intensity values of the input-image.
- Mark intensity values (pixels) that changed based on the reference image.
- Count the blob count - the number of pixels that are composite con-joined clusters (3 _ 3 matrix) of marked pixels.

III. IMPLEMENTATION OF PROPOSED WORK

To implement applications for Hand held devices like Mobile phones (Cell phones) first thing to consider is their capacity of working and their storage capacity. So, to implement this proposed work Java 2 Platform, Micro Edition API's are been used. By this it creates ease of design & understanding for the people. This is proposed inconsideration with CDC and CLDC device configurations.

Depending on how it is implemented, the API's allow application to play and record sound and video and to capture still images. MMAPI was developed under the Java Community Process has JSR 135API.

Video Controller API (JSR 15): It has mainly 4 types of Classes with Package name *javax.microedition.media* that has *package manager, Player*, *PlayerListener* and various types of *controllers*. Through these functionalities of API Video Capturing, Displaying video, capturing images.

Wireless Message API(JSR 120): J2ME wirless toolkit supports the Wireless Messaging API(WMA) with sophisticated simulation environment.WMA1.1 enables MIDLets to send and receive Short Message Service(SMS) or Cell Broadcast Service (CBS) messages.

WMA 2.0 (JSR 205) provides the facilities like MMS services. [4][7]

These are the some of JSR API used here to implement the proposed system. The detailed Block diagram is as shown below,

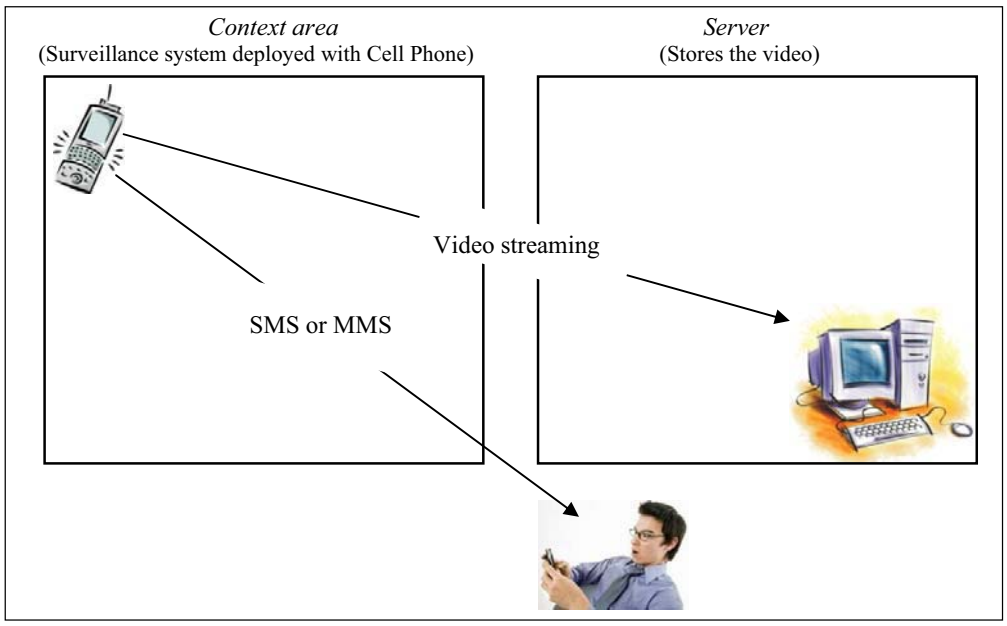

Figure 2. Detailed working Block diagram

Proposed work has been implemented under two modules. 1st module is designed and installed over Cell Phone (That cell phone should be a Java enabled). 2nd module is designed and installed into server that is in remote place from the context area.

Working of the system: Initially Client application present in the Cell phone captures 'n' number of snaps of context area where expected threat is not present. Then that creates back ground template and stores that into the memory for future use.

Now cell phone is ready to keep it at context area. It is left to the user to maintain secrecy during keeping time of cell phone. Before keeping cell phone start the Surveillance application load into the cell phone. Cell phone starts observing the context area to find threats by capturing snaps at every fixed regular intervals of time. And it tries to compare that new snap with created back ground template. If any motion of object is found in new snap then, the client application sends one SMS or MMS message over internet for Authorized person and at next instant starts streaming context video to the server via GPRS that is present and specified by the authorized person.[4]

Later, if the authorized person wants to see the scenario of the contextual area he can find it from the server. It is an effective, efficient, easily maintainable, movable and low cost surveillance system.

For calculating template 'n' numbers of frames are required. If the sizes of frames are high then it becomes very difficult to store those into the memory of cell phone. So, it is necessary to reduce the sizes of the frames. If sizes of the frames are been reduced then information from the snap will be lost. But if reduction is done over unimportant space of the snap it is acceptable. [2][4]

IV. EXPERIMENT AND RESULT

The prototype has been implemented on N70 GSM phone and Motorola IC902 CDMA1X phone. The Phone Configuration is sheet is as given below,

Table 1. Phone descriptions

	N70	IC902
CPU	TI OMAP 1710 ARM-926 220MHz	Qualcomm MSM6550 22f MHz ARM9™
Screen	176 X 208 pixels, 35 X 41 mm	240 X 320 pixels
Camera (Back)	2 MP	2 MP
Network	GPRS, EDGE, WCDMA 400-700kbps	
OS	Symbian OS v8.1a, Feature pack 3	CDMA Brew

Table 2. Result Analysis: Threshold value of pixel variation is given with the application at initial time only. In below Examples Threshold value is 10 %, if pixel variation is above 10 % then you can say Motion of an object is detected.

Sl.No.	Background Template	Compared real time snap	Description
1.			Here in this scenario real time snap is compared with Template and Pixel variation found is 40 % so, *application detects threat found.*
2.			In this scenario real time snap is compared with Template and Pixel variation found is 8 % so, application announces *Motion of object not found.*
3.			Real time snap varies with template by 10 % of pixel variation there *application says Motion of an object found.*

V. CONCLUSION

The Context-aware cell-phone based surveillance system led the development towards autonomous systems. In the traditional surveillance systems there is no independency to freely move the surveillance system and they need separate images capturing unit and images processing unit by that they leads to more cost. Whereas in this, image capturing unit and image processing units are present with the Cell Phone only.

With the help of J2ME technology, proposed work requires minimum hardware. It is very easy to deploy in the emergency situations and also easy to maintain.

Overall it is a low-cost, wide area context-aware cell phone based video surveillance system.

REFERENCES

[1] "Intelligent Distributed Surveillance Systems: a review", M.Valera and S. A. Velastin.

[2] "A practical Home Security System via Mobile Phones", BING-FEI WU, HSIN YUAN PENG, CHAO-JUNG CHEN Department of Electrical and control Engineering National Caio-Tung University 1001 Ta-Hseih Road, HsinChu, Taiwan R. O. C hhtp://cssp.cn.nctu.edu.tw.

[3] "Evaluating Video Streaming over GPRS/UMTS networks: A practical case", Almudena Diaz, Pedro Merino, Laura Panizo, Alvaro M. RecioDpto, Lenguages y Ciencias de la cantutaci'on, University of Malaga, Span. Email: almudiaz@lcc.uma.es,pedro@lcc.uma.ec, laurapanizo@lcc.uma.ec, amrecio@lcc.uma.ec

[4] "A J2ME-Based Wireless Intelligent Surveillance System using Moving object recognition", LizhongXu, Zhong Wang , Huibin Wang , Aiye Shi , Chenming LiCollege of Computer and Information Engineering Hohai UniversityNanjing, P. R. China 210098lzhxu@hhu.edu.cn / wz1980@gmail.com

[5] "Background Subtraction techniques: a review" Massimo PiccardiComputer vision group, Faculty of Information Technology University of Technology, Sydney (UTS), Austrailiamasi; E-mail: imo@it..uts.edu.au.

[6] M. Piccardi, "Background subtraction techniques: areview", IEEE International Conference on Systems, Man andCybernetics, Oct. 2004, vol. 4, pp. 3099–3104.

[7] "Developing Wireless Applications using the J2ME", Bill Day Technology Evangelist Sun Micro Systems, Inc. www.billday.com

[8] "A Proximity Algorithm Accelerated by Gauss-Seidel Iterations for L1=TV Denoising Models", Qia Li † Charles A. Micchelli‡ Lixin Shen§† Yuesheng Xu