

**PATIENT ROOM SECURITY AND  
SURVEILLANCE SYSTEM**

**A THESIS SUBMITTED TO THE GRADUATE  
SCHOOL OF APPLIED SCIENCES  
OF  
NEAR EAST UNIVERSITY**

**By  
NEŞE İPİN YAVUZ**

**In Partial Fulfillment of the Requirements for  
The Degree of Master of Science  
in  
Biomedical Engineering**

**NICOSIA, 2014**

**PATIENT ROOM SECURITY AND  
SURVEILLANCE SYSTEM**

**A THESIS SUBMITTED TO THE GRADUATE  
SCHOOL OF APPLIED SCIENCES  
OF  
NEAR EAST UNIVERSITY**

**By  
NEŞE İPİN YAVUZ**

**In Partial Fulfillment of the Requirements for  
The Degree of Master of Science  
in  
Biomedical Engineering**

**NICOSIA, 2014**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name :

Signature :

Date:

## **ABSTRACT**

This thesis describes a vision based simple and secure way of surveillance in the patient room. The surveillance of any sections in the patient room using the camera should be easy with background subtraction, thresholding and dilation methods which described in the thesis. Video should be acquired from the camera extracted to frames then the video frames processed where the action has detected in the region of interest. Background subtraction technique used to detect an action in the coordinates given in the video sequence. If any action is detects in the given coordinates, software will examine the situation and alert will be sent via SMS over GSM network because of it's highly availability, security and just in time process. In this, thesis application, one USB Camera, Siemens TC35 GSM terminal and an Intel based PC with a RS232 and USB port used to assemble hardware. Then, software developed to make image processing process and interfaced to hardware which control all devices and to fire appropriate alerts.

**Keywords:** Patient room, surveillance system, image processing, background subtraction, Biomedical Engineering

## ÖZET

Bu tez hasta odasını en basit ve güvenli şekilde gözetim altında nasıl tutabileceğimiz göstermektedir. Hasta (veya yoğun bakım) unitelerindeki belli alanların değişimlerini herhangi bir kamera yardımıyla gözetlemek bu uygulama ile oldukça kolay olur. Kameradan alınacak video görüntüsü karelere parçalanır, daha sonra sırasıyla işlenerek ve ilgilendiğimiz kordinatlar üzerinde değişim olup olmadığını tespit edilir. Değişimlerin tespiti imaj işleme tekniklerinden olan "Background subtraction technique" kullanılarak yapılır. İlgilendiğimiz koordinat bloğundan herhangi birinde hareket veya değişim tespit edilirse, yine yazmış olduğumuz program sayesinde durum değerlendirilir ve ilgili alarm ilgili kişilere SMS aracılığıyla GSM üzerinden iletilir. GSM in tercih edilmiş olmasının sebebi ise yüksek oranda kullanılabilir olması, güvenlik probleminin bulunmaması, ve zamanında yerine ulaştırılabilmesi başlıca sebeplerindendir. Tüm bu işlemlerin kontrolü yazdığımız Windows uygulaması ile yapılmaktadır. Ayrıca donanımsal olarak bir adet USB kamera, Siemens TC35 GSM terminal ve RS232 ve USB porta sahip Intel tabanlı PC kullanılarak işlemler gerçekleştirilmiştir.

**Anahtar Kelimeler:** Hasta odası, kamera, görüntü işleme, arka plan çıkarımı, Biyomedikal Mühendisliği

## **ACKNOWLEDGMENTS**

First of all, I would like to thank Assist.Prof.Dr.Boran Şekeroğlu to helping me on Image processing techniques at the early development stages of the thesis. Also I would like to thank him to guiding me and for all his support.

I would like to thank the management of the “Özel Medi-ce İnegöl Hastanesi, Bursa, Turkey” for giving me a room to make a successful demonstration of my thesis.

I would like to thank Op.Dr.Tunç Polat Edizsoy and Op.Dr. Özgür Oktay Nar for helping me on final development stage.

Also a special thank to my husband, Hakan Yavuz, for all his love and support.

And I would like to express my gratitude to all those who gave me the possibility to complete this thesis.

## TABLE OF CONTENTS

<b>ABSTRACT</b> .....	iv
<b>ÖZET</b> .....	v
<b>ACKNOWLEDGMENTS</b> .....	vi
<b>TABLE OF CONTENTS</b> .....	vii
<b>LIST OF FIGURES</b> .....	ix
<b>LIST OF SYMBOLS AND ABBREVIATIONS</b> .....	xi
<b>CHAPTER 1: INTRODUCTION</b>	
1.1 Literature Review.....	2
<b>CHAPTER 2: DETECTION USING BACKGROUND SUBTRACTION</b>	
2.1 An Image.....	5
2.2 Image Processing.....	6
2.3 Image Processing Applications.....	7
2.4 Surveillance Model Componential.....	13
2.4.1 Median Filtering .....	14
2.4.2 Background Subtraction.....	15
2.4.3 True Color to Gray Scale Conversion.....	17
2.4.4 Dilation.....	19
2.5 Application of Surveillance Model.....	19
<b>CHAPTER 3: GSM NETWORK, GSM MODEM, TERMINAL COMMUNICATION AND HARDWARE DESIGN</b>	
3.1 GSM Network Architecture.....	21
3.2 Application Basics.....	22
3.3 Gsm Modem.....	23
3.4 At Commands.....	23

3.5 Hardware Design.....	25
<b>CHAPTER 4: SOFTWARE DESIGN AND IMPLEMENTATION</b>	
4.1 Flow Chart Of Vision Based Patient Room Surveillance System.....	27
4.2 Gsm Modem Setup.....	30
4.3 Delphi Integrated Development Environment.....	31
4.4 Delphi Video Grabber.....	32
4.5 Delphi Gsm Modem Communication.....	33
4.6 Software Flow .....	33
<b>CHAPTER 5: RESULTS AND DISCUSSION</b>	
5.1 Basic Scenario.....	36
5.2 Patient Room Scenarios.....	38
5.3 Patient Room Scenarios Step By Step Application.....	39
5.4 Alarm Sensitivity And Error Calculation.....	43
<b>CHAPTER 6: CONCLUSION AND SUGGESTIONS FOR FUTURE WORK.....</b>	<b>45</b>
<b>REFERENCES.....</b>	<b>46</b>
<b>APPENDICES</b>	
<b>APPENDIX A: TC35 GSM MODEM SPEC.....</b>	<b>49</b>
<b>APPENDIX B: SURVEILLANCE SOFTWARE MODULE.....</b>	<b>51</b>
<b>APPENDIX C: GSM INTERFACE SOFTWARE MODULE.....</b>	<b>69</b>



## LIST OF FIGURES

<b>Figure 1.1:</b>	BGS Application Task Example.....	3
<b>Figure 2.1:</b>	An Image — an array or a matrix of pixels arranged in columns and row.....	5
<b>Figure 2.2:</b>	Each pixel has a value from 0 (black) to 255 (white).....	6
<b>Figure 2.3:</b>	True color image assembled from red, green and blue.....	6
<b>Figure 2.4:</b>	Automatic number plate recognized gate.....	7
<b>Figure 2.5:</b>	Traffic sign and lane recognition.....	7
<b>Figure 2.6:</b>	Passive sensor example of remote sensing.....	8
<b>Figure 2.7:</b>	Moving object tracking and velocity determination.....	8
<b>Figure 2.8:</b>	Recognition based tracking.....	9
<b>Figure 2.9:</b>	Oil spill extent between July 21 and August 10, 2006.....	10
<b>Figure 2.10:</b>	X-Ray and CT example application of biomedical application.....	11
<b>Figure 2.11:</b>	Diagram illustrating the processing steps for generating personalized.....	11
<b>Figure 2.12:</b>	Lung area at inspiration and expiration in a healthy subject.....	12
<b>Figure 2.13:</b>	Processing of the digital mammogram and colored density map of breast parenchyma.....	12
<b>Figure 2.14:</b>	NKK company delta eye automatic surface inspection and marking system....	13
<b>Figure 2.15:</b>	Thermogram shows a fault with an industrial electrical fuse block.....	13
<b>Figure 2.16:</b>	Application of median filter to a simple one dimensional signal.....	14
<b>Figure 2.17:</b>	Application of median filter to a 3x3 pixel image.....	14
<b>Figure 2.18:</b>	Noise generation and median filtering application.....	15
<b>Figure 2.19:</b>	BGS application task example.....	16
<b>Figure 2.20:</b>	Gray scale conversion example RGB image, gray-scale conversion applied...	19
<b>Figure 2.21:</b>	An original image and 3x3 dilation applied image.....	19
<b>Figure 2.22:</b>	BGS real application on patient room.....	20
<b>Figure 3.1:</b>	Basic network structure of the SMS in GSM.....	21

<b>Figure 3.2:</b>	Basic schematic, MS equipment placed on the GSM network.....	22
<b>Figure 3.3:</b>	Basic hardware schematic.....	25
<b>Figure 4.1:</b>	Basic software schematic flowchart.....	29
<b>Figure 4.2:</b>	Hyperterminal terminal settings.....	30
<b>Figure 4.3:</b>	Hyperterminal GSM Modem Session and log view.....	31
<b>Figure 4.4:</b>	Delphi IDE initial view.....	32
<b>Figure 4.5:</b>	Initial delphi project view and components settings.....	33
<b>Figure 4.6:</b>	Software design view.....	34
<b>Figure.4.7:</b>	Step by step process block diagram of the flow.....	35
<b>Figure 5.1:</b>	Basic scenario.....	36
<b>Figure 5.2:</b>	Snapshots 1 basic scenario.....	37
<b>Figure 5.3:</b>	Snapshots 2 basic scenario.....	37
<b>Figure 5.4:</b>	Patient room scenario.....	38
<b>Figure 5.5:</b>	Snapshot 1for patient room background and gray scale conversion.....	40
<b>Figure 5.6:</b>	Snapshot 2for patient room background and gray scale conversion.....	40
<b>Figure 5.7:</b>	Snapshot patient room right hand moved and snapshot in gray scale conversion.....	41
<b>Figure 5.8:</b>	Patient room background subtracted.....	41
<b>Figure 5.9:</b>	Patient room background subtracted & thresholding applied.....	42
<b>Figure 5.10:</b>	Patient room BGS & thresholding & 8 pixel dilate applied.....	42
<b>Figure 5.11:</b>	Four Alerts Fired.....	41

## LIST OF SYMBOLS AND ABBREVIATIONS

<b>AT</b>	AT tention
<b>BG</b>	Back Ground
<b>BGS</b>	Back Ground Subtraction
<b>Bpp</b>	Bit per pixel
<b>BSC</b>	Base Station Controllers
<b>BSS</b>	Base Station System
<b>BTS</b>	Base Transceiver Stations
<b>CPR</b>	Cardiopulmonary resuscitation
<b>CT</b>	Computer Aided Tomography
<b>DNR</b>	Do Not Resuscitate
<b>GIF</b>	Graphics Interchange Format
<b>GPRS</b>	General Packet Radio Service
<b>GSM</b>	Global System for Mobile Communication
<b>HDTV</b>	High Definition TV
<b>HLR</b>	Home Location Register
<b>i</b>	Interval
<b>IDE</b>	Integrated Development Environment
<b>JPEG</b>	Joint Photographic Experts Group
<b>LZW</b>	Lempel-Ziv-Welch
<b>MAP</b>	Mobile Application Part
<b>MO</b>	Mobile Originated
<b>MRI</b>	Magnetic Resonance Imaging
<b>MSC</b>	Mobile Switching Center
<b>MT</b>	Mobile Terminated
<b>PA</b>	Personal Announcement
<b>PAL</b>	Phase Alternating Line
<b>PC</b>	Personal Computer
<b>PDA</b>	Personal Digital Assistants
<b>PP</b>	Point to Point
<b>PS</b>	Post Script
<b>PSD</b>	Photoshop Document

<b>RGB</b>	Red-Green-Blue
<b>SME</b>	Short Message Entity
<b>SMS</b>	Short Message Service
<b>SMS</b>	Short Message Service
<b>SMSC</b>	Short Message Service Center
<b>t</b>	Time
<b>Th</b>	Threshold
<b>t<sub>i</sub></b>	Time in i <sup>th</sup> interval
<b>TIFF</b>	Tagged Image File Format
<b>USB</b>	Universal Serial Bus
<b>VLR</b>	Visitor Location Register
<b>YUV</b>	Y-Luminance, U-Chrominance1, V-Chrominance2

# **CHAPTER 1**

## **INTRODUCTION**

The purpose of this thesis is to articulate the techniques involved in this thesis for patient room security that how it will detect the action in the region of interest. This thesis will give the explanations to the image processing concepts in the field of surveillance and triggering specified alert by sending SMS communication via GSM Modem. Step by step methods of the techniques used and corresponding results are provided following chapters.

Today, Image processing is a rapidly growing area of computer science. Its growth has been supported by technological advances in computer processors, digital imaging, and mass storage devices. So, traditionally used analogue imaging is now switching to digital systems, because of their flexibility and affordability. Important examples can be seen in medicine, photography, film and video production, security monitoring, and remote sensing. These and other sources produce huge volumes of digital image data every day, more than could ever been examined manually (Seemann, 2002).

Image processing is a method that to get an enhanced image or extract some useful information from digitalized image. That is a signal manipulation that input is an image, like digital photograph or video frame and output may be an image enhanced on characteristics associated. While the signal processing methods applying to image, the Image Processing system includes treating images as two dimensional signals.

First of all, on digital image processing, useful information from images extracted. Ideally, that needs too much processing power, so it has done by computers almost no human intervention. Image processing algorithms placed at three levels. At the bottom level, those techniques deal with the raw, de-noising and edge detection being good examples. In the middle level, it utilizes low level results for later use such as edge linking and segmentation.

Background subtraction (BS) is a technique that to segmenting out an object of interest in a scene like in any surveillance area. This technique involves subtracting an observed image from a background image. The areas on the image where there is a significant difference between the observed and static images indicate the location of the moving objects. The technique of subtracting the observed image from a background image, which known as

background subtraction and thresholding applied result to find objects of interest involved in this thesis. Retrieving the coordinates after analysis and processing images, would acquaint us about changes in the area to trigger alert.

Today's technology brings the alert triggering system which is GSM communication. Today everybody has a mobile phone, and they carry on hand most of the time. With a mobile phone, one can send and read SMS (Short Message Service) messages at any time, no matter whether one is in an office, on a bus or at home. A lot of innovative applications are now developing on SMS mobile service, and more are being developed. Some common example use of SMS applications, are Person-to-Person text messaging, provision of information, downloading/uploading binary data, alerts, notifications and marketing. In this thesis, SMS used to give alarm to the responsible person if any unwanted action in the patient room. It preferred because of its highly availability.

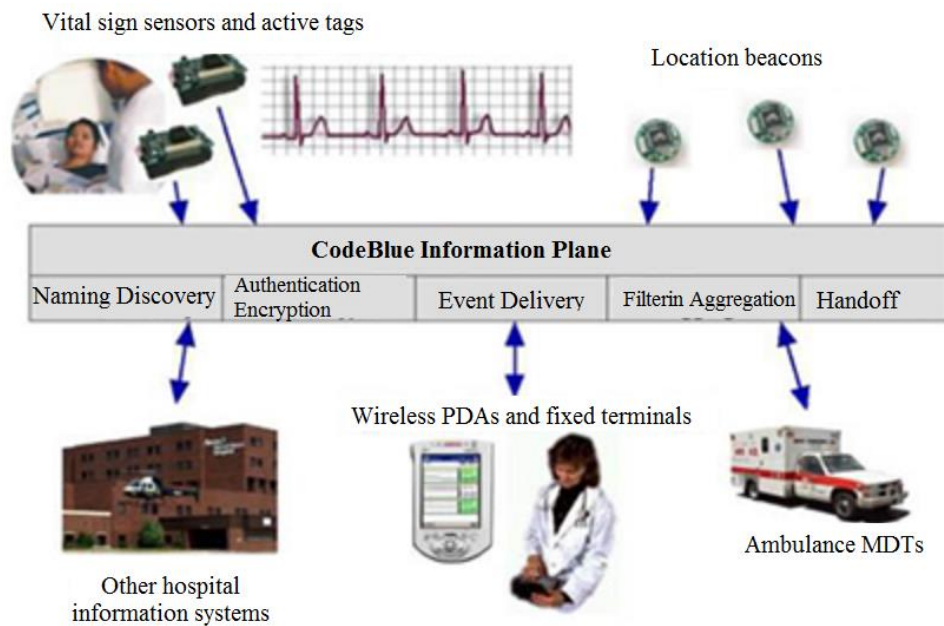
This thesis will cover interfacing DELPHI with default camera and GSM Modem, the methodology, experiments and results. The interfacing of DELPHI part will talk briefly about the DELPHI, the camera which used in the thesis and its specifications. What is the approach in the thesis and the assumptions we have made, described in this section. The detection technique part will handle the techniques used in the thesis and have a visual explanation shown for the readers. The results section will show the images taken from real application, after performing each image processing technique and necessary SMS communication.

### **1.1. Literature Review**

In the literature search although there is not a patient room surveillance technique which developed in the thesis, there is some sensor based network and coding systems exists and mainly used to overcome the problems that which are noted. In the most of the countries, hospitals developed a coding system to give safe, quality health care for their communities. The aim of coding system is to communicate an emergency quickly as possible and to mobilize expert assistance. Especially that gains importance when then physicians and staff often work in multiple hospitals and clinics.

The Code Blue is a sensor-based network that proposed for emergency medical care consisting of vital sign sensors, personal digital assistants (PDAs) and personal computer(PC) systems (Malan et al., 2004).That enables medical teams to assess the condition of patients at

the emergency site and ensures seamless transfer of data among medical personnel. This architecture provides reliable data transfer and a decentralized security system as shown in Figure 1.1.



**Figure 1.1:** BGS application task example (Channa & Ahmed, 2010)

It provides handoff of credentials across rescue personnel and seamless access control across patient's transfer. The Code Blue uses publish/subscribe model for data delivery. The sensor nodes detect vital signs, location information and identities. The physicians subscribe to this information through PDAs and PC systems to monitor the patient status. The major security services offered by Code Blue include authentication and encryption. It desired to develop light-weight cryptographic techniques for resource constrained mobile nodes in Code Blue.

The encryption service also provides confidentiality of critical information, such as vital signs of patients (www.spacetelescope.org, 2012).

Code blue system working principles can be listed:

- It initiated whenever a person appears to be unresponsive or could become unresponsive regardless of reason like cardiac arrest, injury, illness, medication reaction, etc...

- The staff finding the unresponsive person will take the role of Incident Manager and will call the Code Blue response. The manager will make the announcement of CB, and the location of the Code and the announcement will page overhead three times.
- All available staffs like medical, physicians, nursing will access to the location of the Code Blue.

Finally, in any Emergency cases the most important things are time, good communication and good coordination between staffs. Although there are some coordination techniques are available like Code Blue System, they are not preventing the emergency cases before that have happened. With the technique in this thesis some add-ons to patient room can be seen to overcome most of the emergency cases, and well organize the staff before they face with the emergency situations.

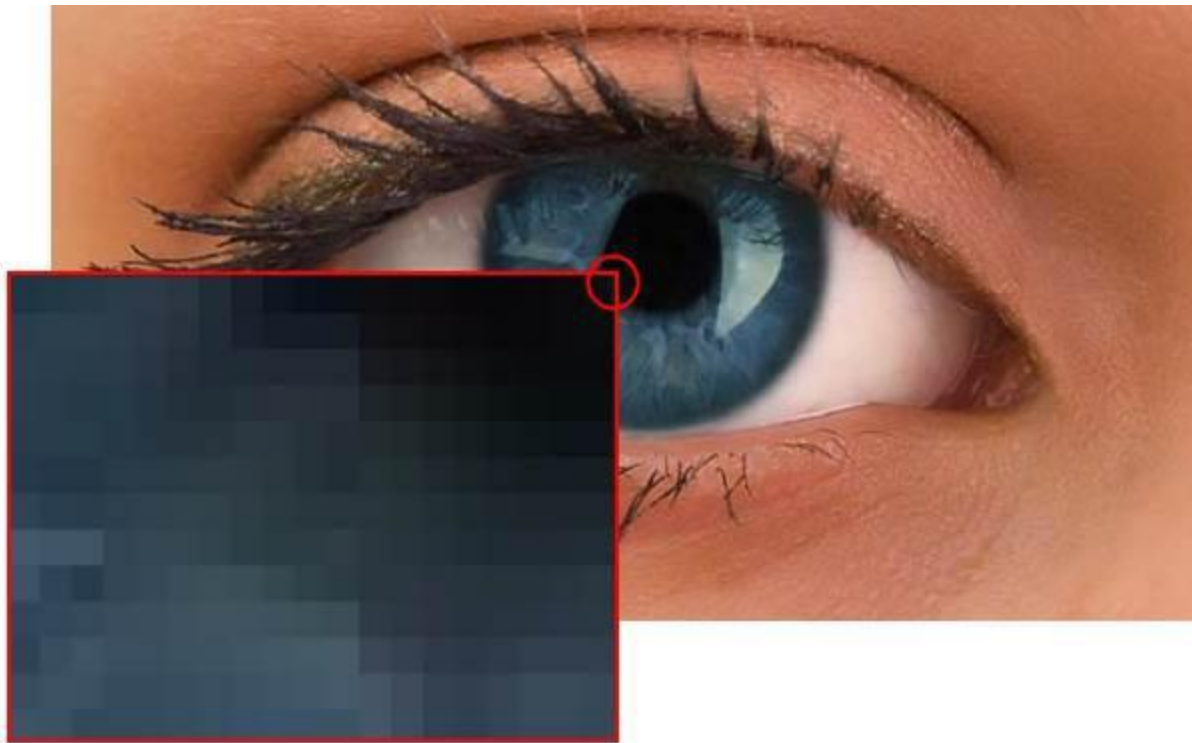


## CHAPTER 2

### DETECTION USING BACKGROUND SUBTRACTION

#### 2.1. An Image

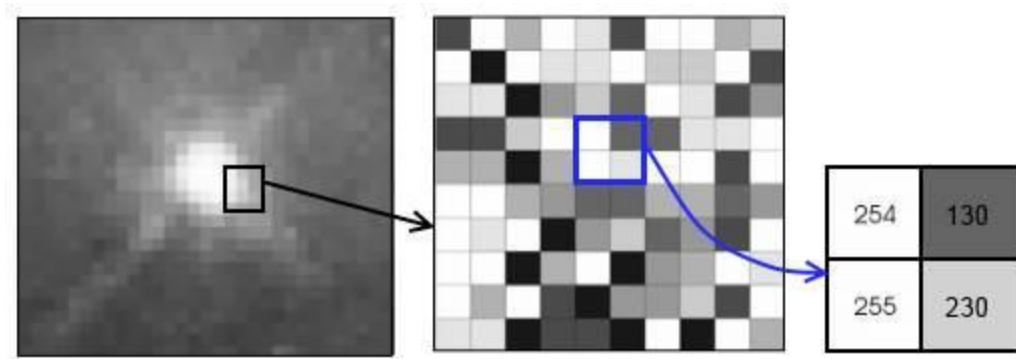
An image is a matrix of square pixels (picture elements) arranged in columns and rows, like in a two dimensional array. Figure 2.1 shows that a focused part of true color image how it is looking like.



**Figure 2.1:** An image — an array or a matrix of pixels arranged in columns and rows

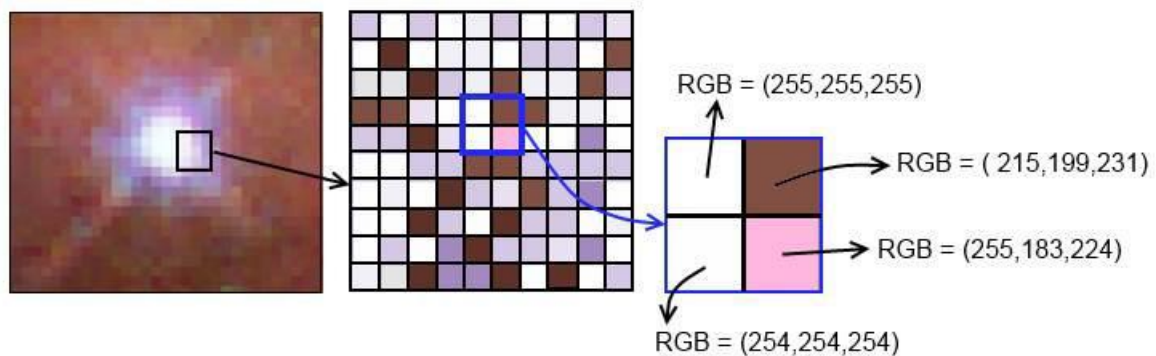
True Color image contains 24 bits color depth and that are combination 8-bits of Red, Green and Blue Color intensity. It has 16777216 colors, and shortly it is called as 16million colors. (RGB colors represents = 8 bit of Red \* 8 bit of Green \* 8 bit of Blue)

In the grey scale image, each picture element has an intensity that a range from 0 to 255. That is normally called a black and white image, but such an image will also include many shades of grey. An example of a grey scale image is shown in the Figure 2.2.



**Figure 2.2:** Each pixel has a value from 0 (black) to 255 (white) (www.spacetelescope.org, 2012)

On Figure 2.2, the color depth of the image shows the possible range of the pixel values which depends on, so here 8 bit = 256 tones or grayscales.



**Figure 2.3:** A true-color image assembled from red, green and blue (www.spacetelescope.org, 2012)

## 2.2. Image Processing

Image processing called as a method that converts a real image into digital form in order to get an enhanced image, to extract useful information. Rapidly growing technologies today made its applications in various aspects of a business. Image Processing now the core research area within engineering and computer science disciplines.

Image processing includes the following steps:

- Acquiring an image from an optical scanner or by digital cam.
- Manipulating and analyzing an image.
- On the final stage output can be altered or reporting based on image analysis.

The main purpose of Image processing can be listed in five groups:

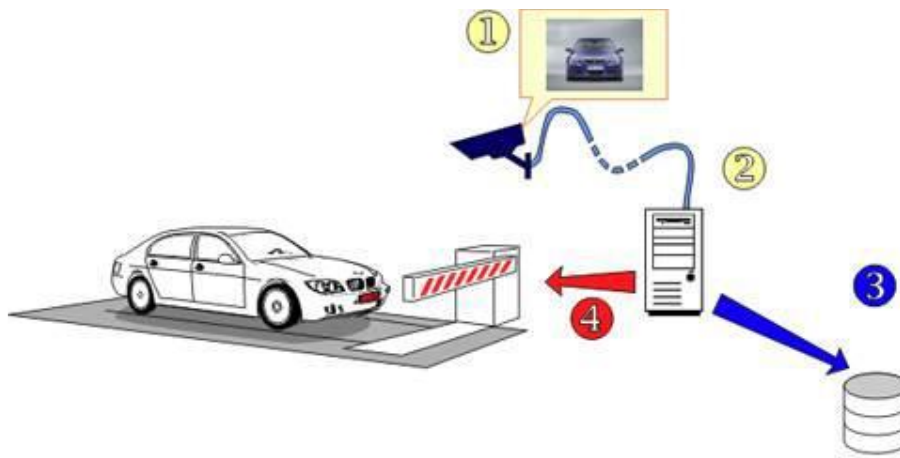
- 1- Visualization: Observe the objects that are not visible.
- 2- Image sharpening and restoration: To create a better image.

- 3- Image retrieval: Seek for the image of interest.
- 4- Measurement of pattern: Measures various objects in an image.
- 5- Image Recognition: Distinguish the objects in an image.

### 2.3. Image Processing Applications

Image processing applications can be listed like in the following:

1. Intelligent transportation systems: Automatic number plate recognition and traffic sign recognition system can be the best example of it and figure out on Figure 2.4 and Figure 2.5.



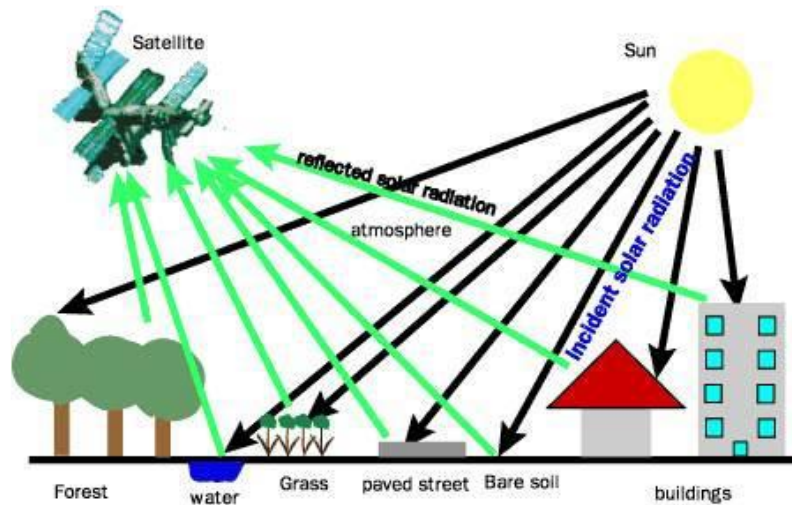
**Figure 2.4:** Automatic number plate recognized gate



**Figure 2.5:** Traffic sign and lane recognition (Patented by Bosch)

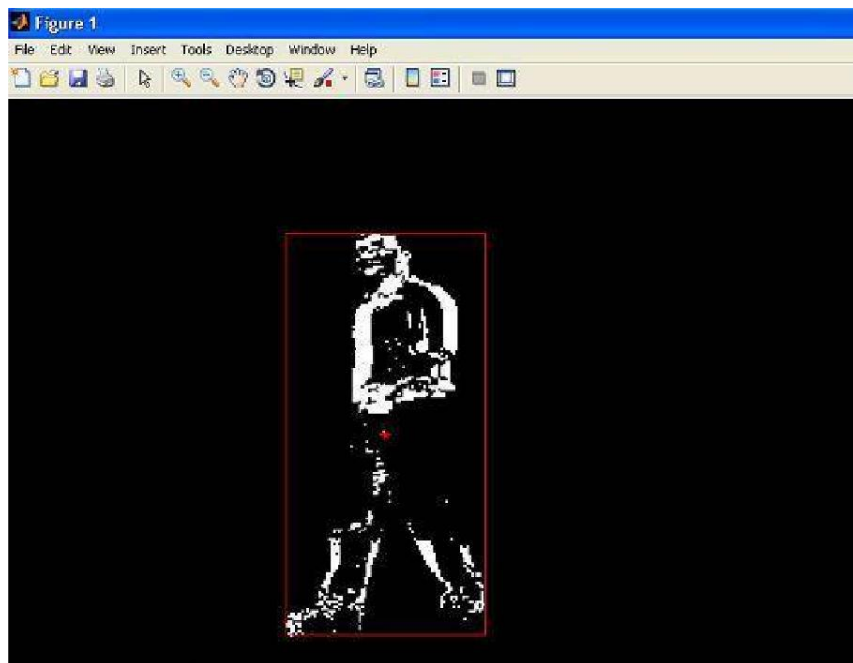
2. Remote sensing: Like shown in the Figure 2.6, in this application, A sensors capture a picture of the earth's surface in remote sensing satellites or multispectral scanner, which is mounted on an aircraft. The picture captured is processed by transmitting it to the station on Earth. Techniques used to interpret an object and the regions are used in

city planning, flood control, resource mobilization, agricultural production monitoring, etc.



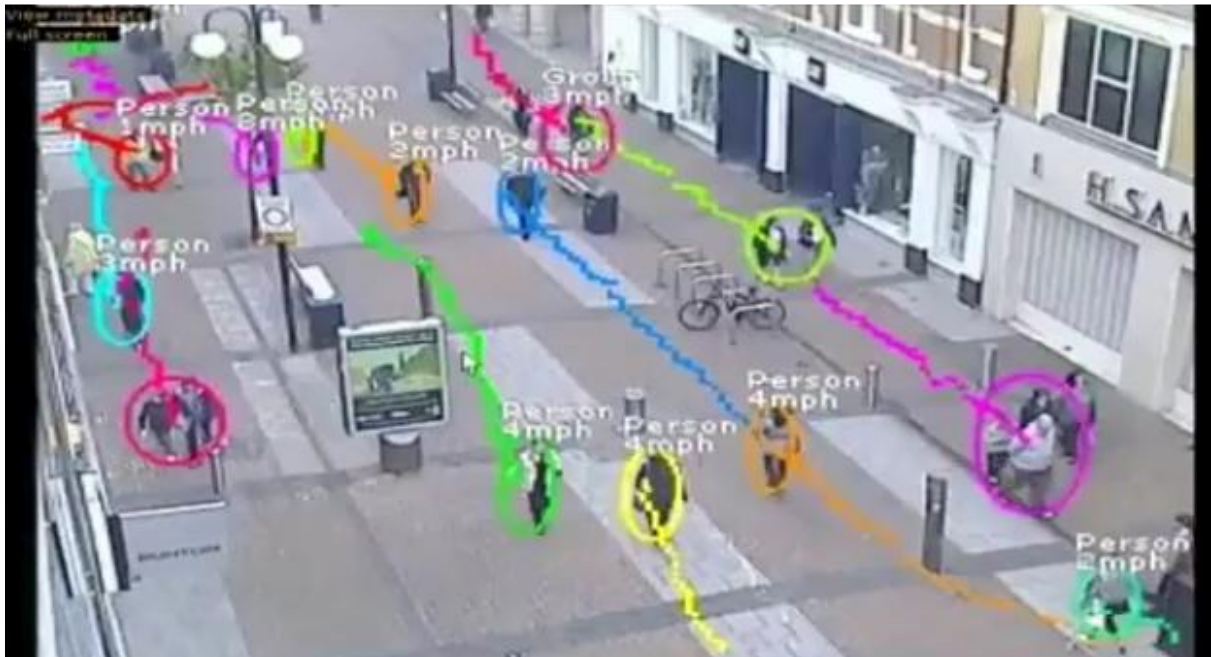
**Figure 2.6:** Passive sensor example of remote sensing

3. Moving object tracking: To acquire visual record of the moving object and measure motion parameters, motion based tracking or recognition based tracking approach are mainly used. The example of moving object tracking, recognition based tracing and velocity determination example figure out on Figure 2.7 and Figure 2.8.



**Moving object tracking and velocity detection**

**Figure 2.7:** Moving object tracking and velocity determination (Shanthan & Reddy, 2012)

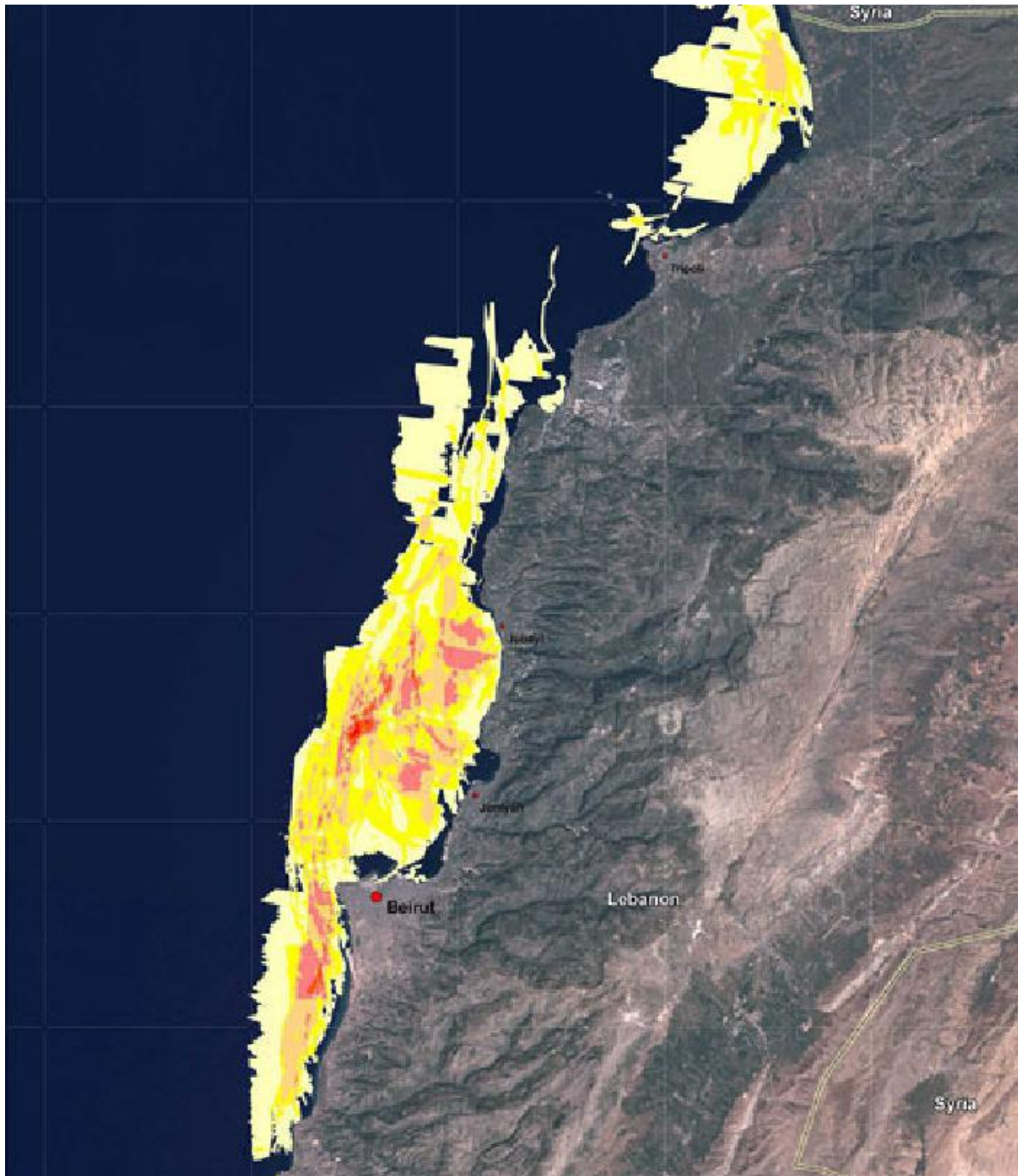


**Figure 2.8:** Recognition based tracking (UK Surveillance Commissioner, 2011)

4. Defence surveillance: It is used to keep an eye on the air, land and oceans continuously.

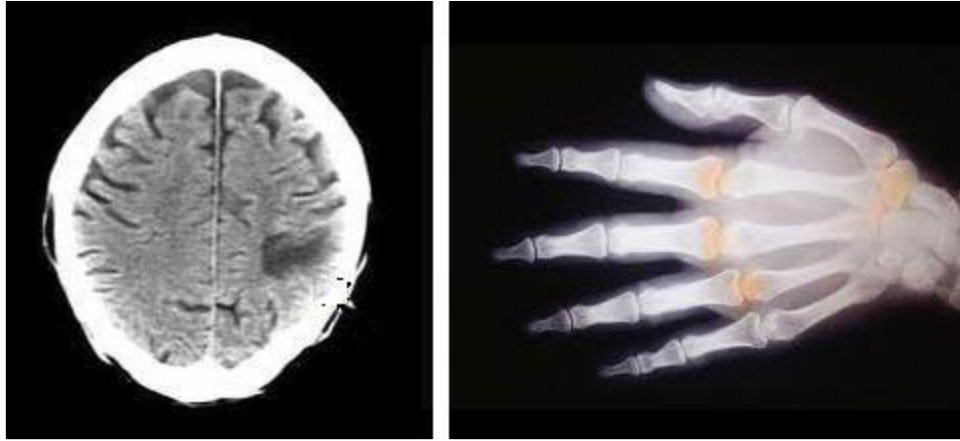
As an example Figure 2.9, shows that oil spewed into the sea after Israeli attack to power utility on the south of Beirut between 13 and 15 July. Estimated oil spewed calculated as 10,000 to 15,000 tons using an image processing technique.





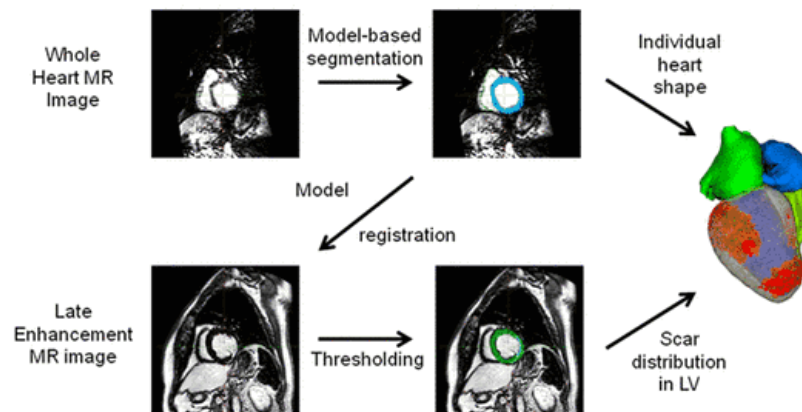
**Figure 2.9:** Oil spill extent between July 21 and August 10, 2006 (UN News Service, 2006)

5. Biomedical Imaging techniques: Mainly use on medical diagnosis such as X- ray, Ultrasound, computer aided tomography (CT). The example of X- ray, MRI, and computer-aided tomography (CT) application is like given below figure.



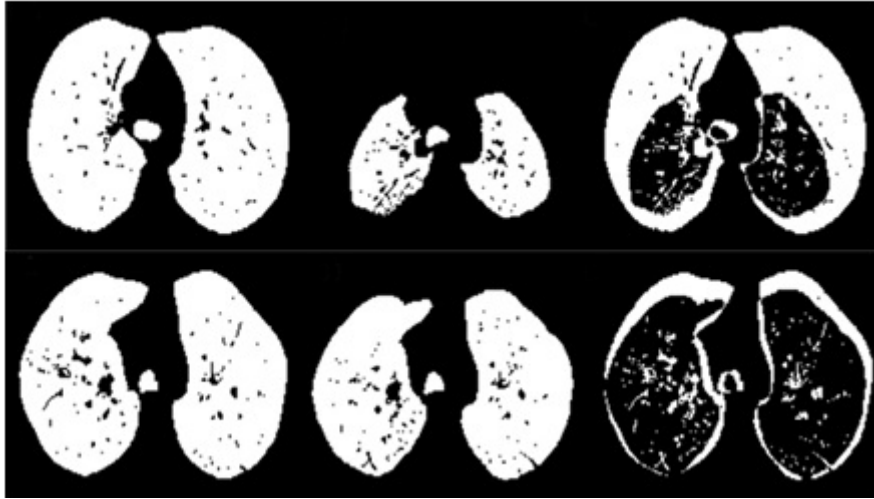
**Figure 2.10:** X-Ray and CT example application of biomedical application  
(www.engineersgarage.com, 2013)

Another application of biomedical imaging is heart disease identification. The important diagnostic features like size of a heart which shape are required to know in order to classify the heart diseases. Figure 2.11 shows the model to improve the diagnosis of heart diseases using image analysis techniques.

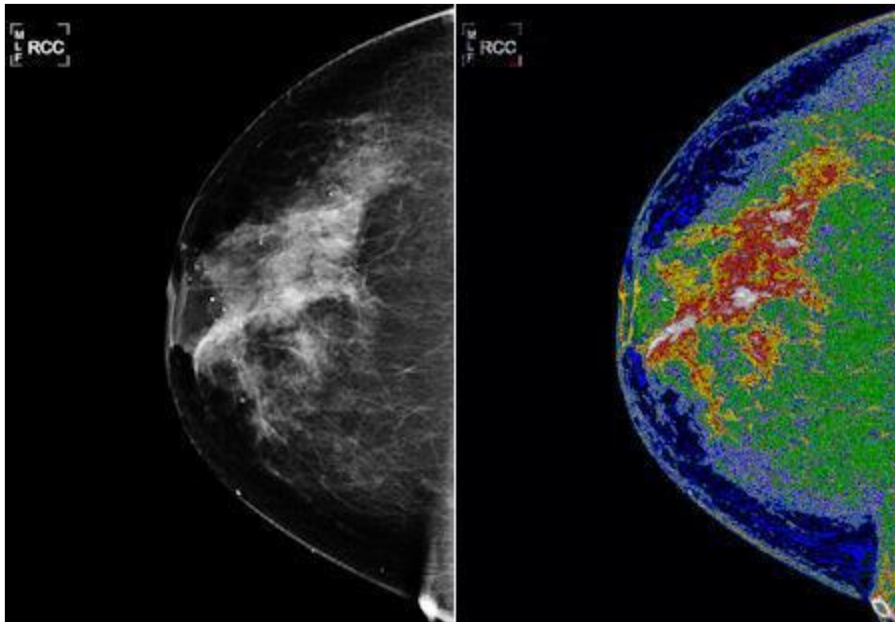


**Figure 2.11:** Diagram illustrating the processing steps for generating personalized heart models  
(Philips Research Laboratories, 2012)

Lung disease identification in X- rays film using biomedical imaging is like in the Figure2.12.



**Figure 2.12:** Lung area at inspiration and expiration in a healthy subject (Hosseini, Soltanian-Zadeh & Akhlaghpour, 2012)



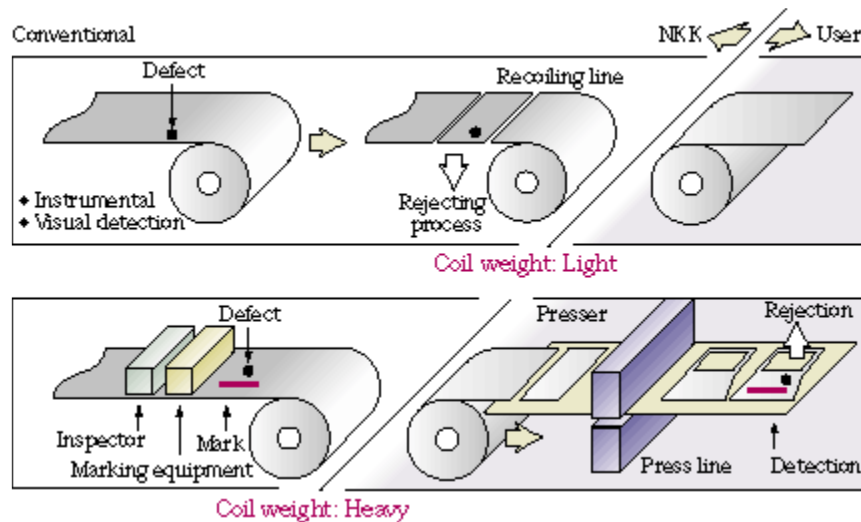
**Figure 2.13:** Processing of the digital mammogram and colorized density map of breast parenchyma (NKK Company, 2002)

Another application area is digital mammograms to detect the breast tumors. Mammograms can be analyzed using some Image processing techniques such as shape analysis, segmentation, contrast enhancement, feature extraction, etc. Example of producing a colorized density map of breast parenchyma like shown in the above Figure2.13.

6. Automatic Visual Inspection System: These systems improve productivity and quality in the industries. The example application of automatic surface inspection system in NKK metal industry can be seen on Figure 2.14 which detects the flaws on the



surfaces. To detect faulty components in electronic devices can be identified by analyzing the Infra-red images like in Figure 2.15.



**Figure 2.14:** NKK company delta eye automatic surface inspection and marking system (Maldague et al., 2001)



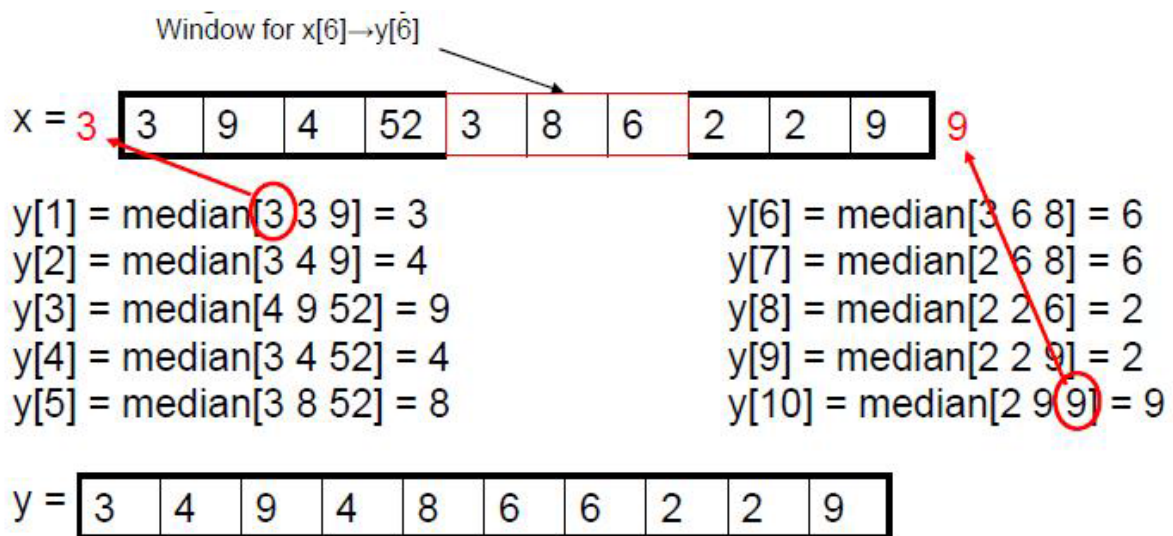
**Figure 2.15:** This thermo gram shows a fault with an industrial electrical fuse block (Huang, Yang & Tang, 1979)

## 2.4. Surveillance Model Componential

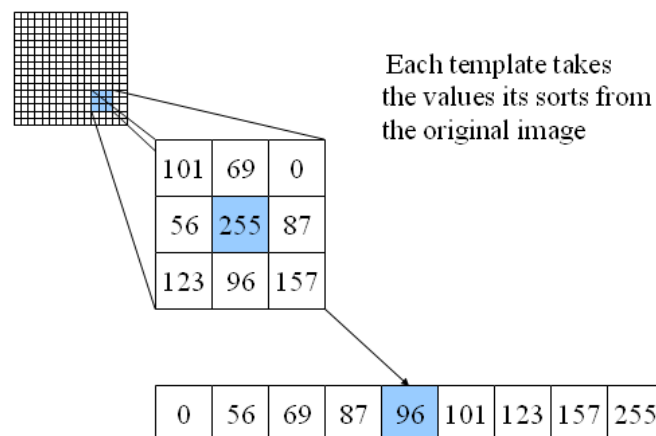
The selected method in the thesis for the surveillance application is usually known as background subtraction(BS), which is use to segmenting out objects of interest in a scene. This technique involves subtracting an observed image from a static image (background image). To reduce noise and distortion median filtering must apply to captured frames. In the second step images must convert to gray scale to make comparison efficiently. After completing the BS, thresholding must apply to binary comparison. To make the result certain Dilation techniques applied to make the final decision.

### 2.4.1 Median Filtering

Median filtering is a nonlinear method which used to remove noise from an images (Huang, Yang & Tang, 1979). This technique is widely used and it is very effective method to removing noise while preserving edges. That is effective at removing salt and pepper type noise. Median filter replaces each pixel value with a median value of neighboring pixels. The group of neighbor pixels called as window. The median value is calculated by sorting all the pixel values in the window on ascending order, and replacing the pixel with the middle pixel value which called as median. Figure 2.16 shows that the application of a median filter to a simple one-dimensional signal.

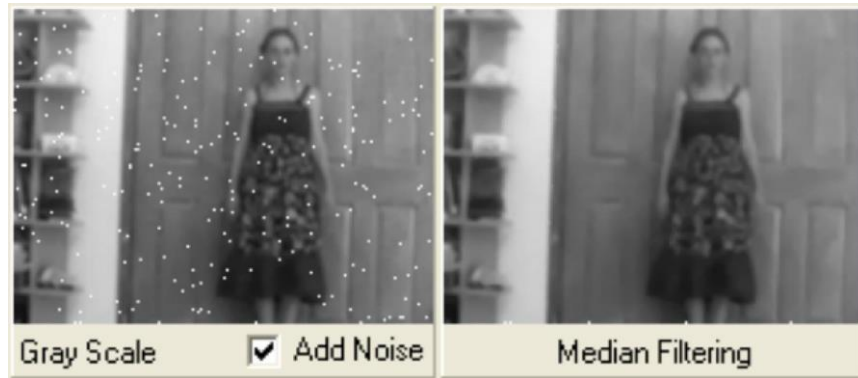


**Figure 2.16:** Application of median filter to a simple one dimensional signal (www.cs.auckland.ac.nz, 2011)



**Figure 2.17:** Application of median filter to a 3x3 pixel image (www.cs.auckland.ac.nz, 2011)

Another 3x3 window sized application can be seen in the Figure 2.17. The pixel values within this example are sorted and in the middle of the sorted list are used to replace the window central pixel as seen in the figure.

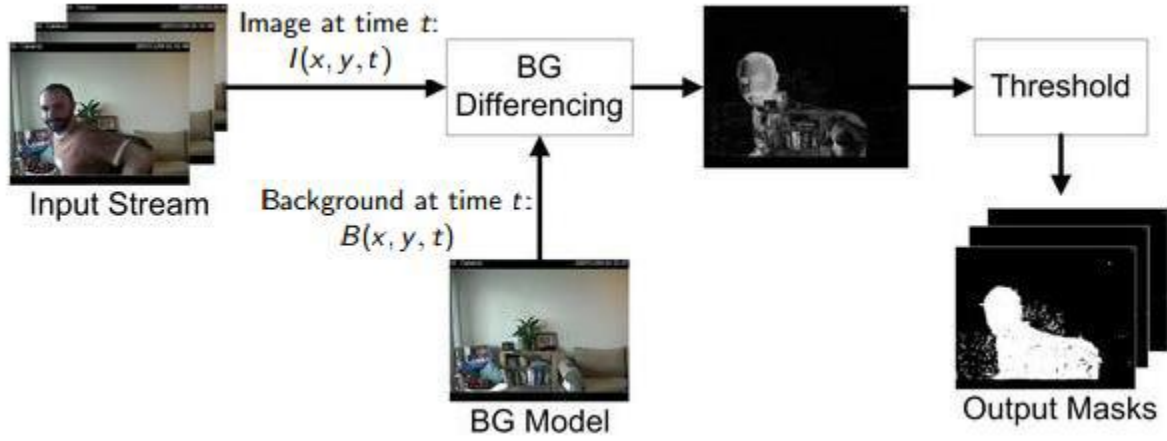


**Figure 2.18:** Noise generation and median filtering application

In the application, quality of an image captured from the camera is quite good. But for the worst case, if some distortion exists on medium for real application we try to simulate how the median filtering routine should be successful. Using median filter procedure in thesis, the result handled like in the Figure 2.18.

### **2.4.2 Background Subtraction**

The processing of a video stream to segment foreground objects from the background (BG) is the first step in many computer vision applications. The background subtraction (BGS) technique is a commonly used technique to achieving this segmentation. The BGS largely prefers because of its computational efficiency, which allows applications such as video surveillance, human-computer interaction and traffic monitoring to meet their real-time goals. To improve the results there are some of the pre-processing and post-processing techniques applied to subtracted image.



**Figure 2.19:** BGS application task example (www.spacetelescope.org)

Simple Approach is:

1. Capture the background for time t.
2. Subtract the captured background from the input frame.
3. Apply a thresholding, for the absolute difference to get the foreground mask.

Because of illumination changes, motion changes and background geometry background image not always fixed so it must adapted. So a robust background subtraction technique should be able to handle illumination changes[16] The Illumination function of  $I(x,y,t)$  as a video frame sequence where  $t$  is the time dimension, and  $(x,y)$  are the pixel location variables. E.g.  $I(1,2,0)$  is the pixel intensity at  $(1,2)$  of the image at  $t = 0$  in the video frame sequence.

### Frame Differencing

Frame difference at time  $t + 1$  is formulated as equation 2.1.

$$D(t+1) = | I(x,y,t+1) - I(x,y,t) | \quad (2.1)$$

The background is assumed to be the captured image at time  $t$ . The image difference shows some intensity for the pixel locations which have changed between two frames (Tamersoy, 2009). To make a binary decision threshold  $Th$  is considered.

$$| I(x,y,t) - I(x,y,t+1) | > Th \quad (2.2)$$

That is called as threshold on the basis of value of Th. The speed of movement in the scene effect the accuracy of this approach and faster movements may require higher thresholds.

#### Mean filter

To calculating the background image B at the instant t, a series of proceeding images that contains only the background averaged.

$$B(x, y) = 1/N \sum_{i=1}^N I(x, y, t - i) \quad (2.3)$$

Where N is the number of the captured images taken for averaging, N would depend on the video frame speed (number of frame per second in the video) and the amount of movement in the video (Benezeth et al., 2008). When the background B(x,y) calculated then it subtracted from the image I(x,y,t) at time t and thresholding applied formulated like equation 2.4.

$$| I(x,y,t) - B(x,y) | > Th \quad (2.4)$$

Where Th is threshold.

#### **2.4.3 True Color to Gray Scale Conversion**

Conversion to grayscale is not unique but as a common strategy is to compare match the luminance of the grayscale image to the luminance of the color image. To convert any color to a grayscale represents of its luminance, and its red, green, and blue (RGB) value must be obtained to primaries in linear intensity encoding, by gamma expansion. For the sRGB color space, the gamma expansion is defined as (Johnson, 2006).

$$C_{\text{linear}} = \begin{cases} \frac{C_{\text{srgb}}}{12,92} & C_{\text{srgb}} \leq 0,04045 \\ \left( \frac{C_{\text{srgb}} + 0,0055}{1,055} \right)^{2.4} & C_{\text{srgb}} > 0,04045 \end{cases} \quad (2.5)$$

Where is gamma-compressed sRGB primaries in range [0,1], is the linear intensity value also in range [0,1]. Then, luminance is calculated as a weighted sum of the three linear-intensity values. For the ITU-R BT.709 primaries, as used in sRGB, the weighting (Johnson, 2006).

$$Y = 0.2126 R + 0.7152 G + 0.0722 B \quad (2.6)$$

Gives the CIE 1931 luminance, Y, as the result (Stokes, 1996). Linear luminance needs to be gamma compressed to get back to a grayscale representation. To encode grayscale intensity in RGB, each of the primaries can be set to equal the calculated luminance. For sRGB, the gamma compression is (Johnson, 2006).

$$C_{srgb} = \begin{cases} 12,92 C_{linear}, & C_{linear} \leq 0,0031308 \\ 1,055 C_{linear}^{1/2,4} - 0,055, & C_{linear} > 0,0031308 \end{cases} \quad (2.7)$$

This is not the method used to obtain the luma in the Y'UV and related color models, used in standard color TV and video systems as PAL, SECAM, and NTSC. These systems directly compute a gamma-compressed luma as a linear combination of gamma compressed primary intensities, rather than use linearization via gamma expansion and compression. In the YUV and YIQ models used in PAL and NTSC system, the luma (Y') computed as (Johnson, 2006).

$$Y = 0,299R + 0,587G + 0,114B \quad (2.8)$$

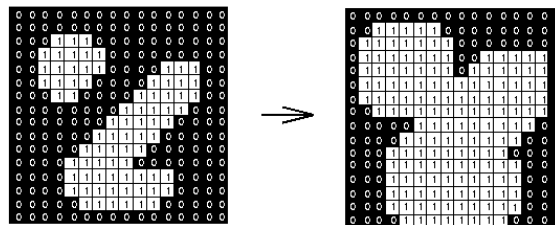
On the Figure 2.20 shows that when the gray scale conversion algorithm applied to picture on right and resulted output should be on the picture on left.



**Figure 2.20:** Gray scale conversion example RGB image, gray-scale conversion applied  
(Johnson, 2006)

#### 2.4.4 Dilation

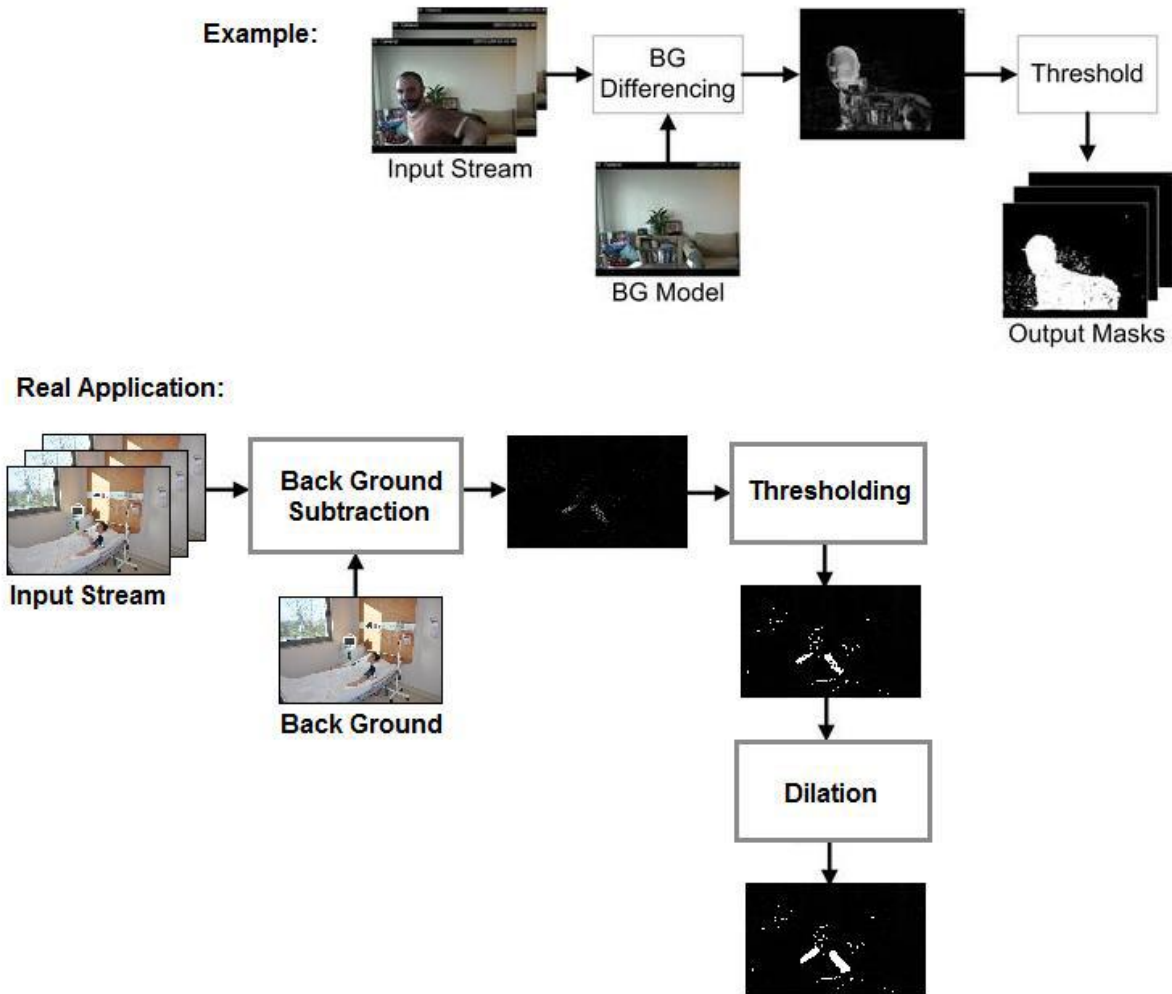
Dilation technique adds pixels to the boundaries of objects in the image to get stronger output. In the Figure, 2.21 shows a good example of the 3x3 square structured dilation applied.



**Figure 2.21:** An original image and 3x3 dilation applied image  
([http://www.robotix.in/tutorials/category/opencv/noise\\_reduction](http://www.robotix.in/tutorials/category/opencv/noise_reduction))

### 2.5. Application of Surveillance Model

Up to this stage of the chapter, tried to figure out which Image processing techniques can be use on surveillance in thesis and how. As the image processing part of the thesis contains the following componential: On the first stage, there is a camera which capturing video(Input Stream) frame by frame. Each frame consists of an RGB image that described earlier, and Median Filtering applied to remove the noise. On the first frame image captured as a background image which will use to make comparison next frames.



**Figure 2.22:** BGS real application on patient room

After the first frame, each image captured by camera will be compared with reference image (Background Subtraction). Resulted image should be in gray scale format which is described earlier. To understand there is a motion/change, it must be converted to binary image, which will contains ones and zeros using thresholding. Finally, to make the result strong, dilation will be applied. Detection algorithm will work after dilated image handled. If any percentage of whites (Ones) more than zeros on the interest areas, the alert will be triggered by the software which we developed. Image processing part of the model figure out shown in the Figure 2.22.

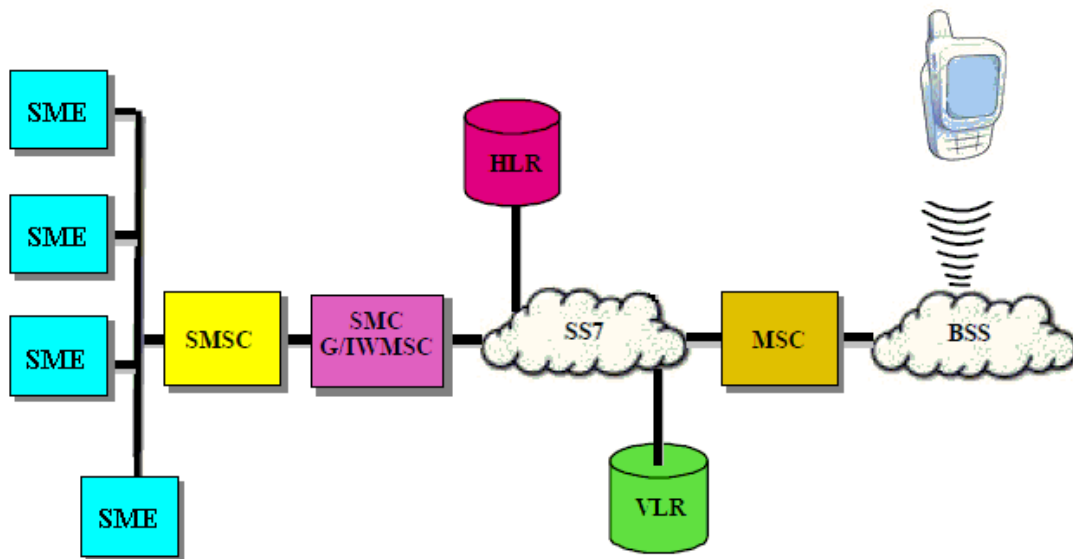


## CHAPTER 3

### GSM NETWORK, GSM MODEM, TERMINAL COMMUNICATION AND HARDWARE DESIGN

#### 3.1. GSM Network Architecture

Short Message Service (SMS) appeared on the wireless scene in 1991 in Europe, where digital wireless technology first took hold. The European standard for the global standard for mobiles (GSM) included short messaging services from the outset ([http://www.mobilein.com/SMS\\_tutorial.pdf](http://www.mobilein.com/SMS_tutorial.pdf)). That provides a mechanism to transmitting short messages between wireless handsets. To do that, use of a short message service center which work as a store and forward system for short messages? The wireless network provides for the transport of short messages between the SMSCs and wireless handsets. In contrast to existing text message transmission service elements are designed to guarantee delivery of messages to the destination. Basic network structure of the SMS is shown like in the Figure 3.1.



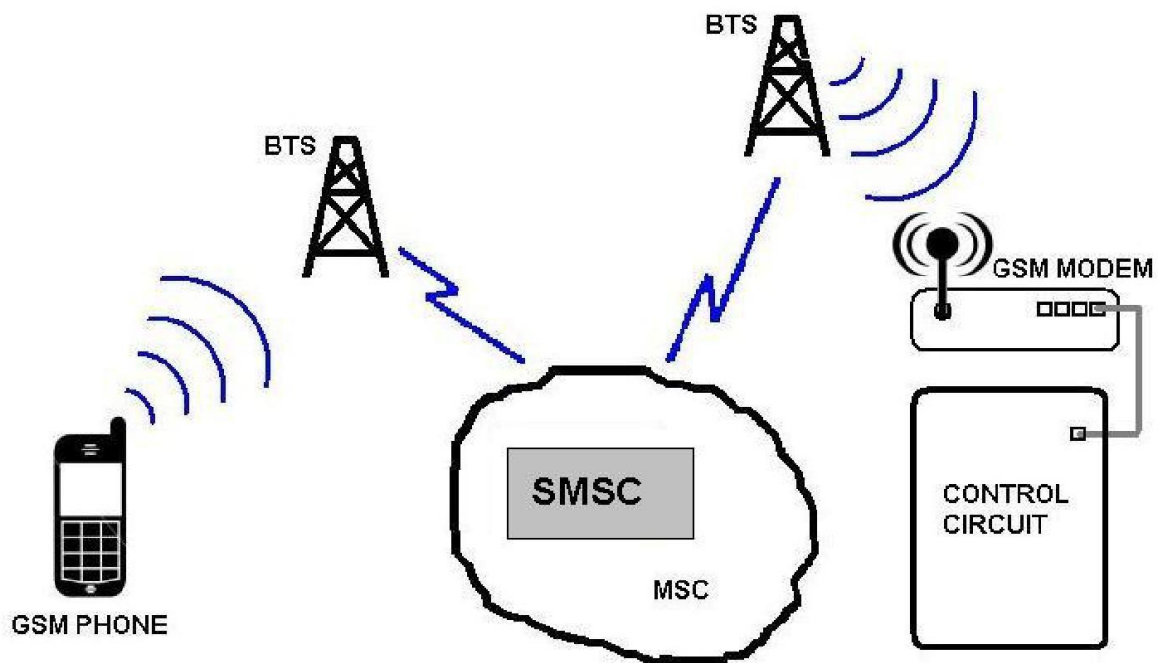
**Figure3.1:** Basic network structure of the SMS in GSM ([www.comptek.ru](http://www.comptek.ru))

Short messaging entity (SME) takes a role to receive or send short messages. It may be located in the mobile station, fixed network, or another service centre. The SMS forwarded by SME to Short Message Service Centre (SMSC) that is responsible storing and forwarding of a short message between the mobile station and SME.

The SMS gateway is an MSC capable of receiving a short message from the SMSC. The home location register (HLR) is permanent storage of subscriptions and service profiles that contain a database. Regarding to interrogation by the SMSC, the HLR provides the routing information for the subscriber indicated. SMSC also informed by HLR that has previously initiated unsuccessful delivery of short message attempts to a specific mobile station. In that way, the mobile client is recognizable by the mobile network to be accessible. The mobile switching centre (MSC) controls the switching functions of the GSM, and that controls calls between other telephone and data systems. The base station system (BSS) take a role to provide all radio-related functions are performed in it. The BSS contains base station controllers (BSCs) and a base transceiver stations (BTSs) that its responsibility is to transmit call and data traffic between the mobiles.

### 3.2. Application Basics

In this application two type equipment used as a Mobile Station (MS) which are GSM Modem (Siemens TC35) and SMS sending capable GSM phone.



**Figure 3.2:** Basic schematic, MS equipment placed on the GSM network (Yavuz, 2002)

The control circuit is a combination of PC and electronic relay circuit driven via parallel port. PC part communicate with modem via RS232 port. The schematic diagram can be seen on Figure 3.2.

### **3.3. GSM Modem**

A GSM modem is a kind of wireless modem that works over a GSM network. It behaves like a dial-up modem, but it sends and receives packets through radio waves. A GSM modem can be found a form of an external device, a PC Card or a PCMCIA Card. Typically, an external modem is connected to a computer via serial cable or USB cable. GSM modems work with a SIM card taken from a GSM network provider in order to operate. AT commands are used to control and communicate to GSM modems where AT refers to “Attention!”. Like in dial-up modems, GSM modems support a common set of standard AT commands. In addition to that, GSM modems supports extended set of AT commands that are defined in the GSM standards. Extended AT commands make the available for the following procedures:

- SMS message reading, writing and deleting.
- SMS message sending.
- The signal strength monitoring.
- The charging status and charge level monitoring of the battery.
- Phone book entries reading, writing and searching.

### **3.4. AT Commands**

AT commands are instruction sets to control a modem ([www.robotshop.com](http://www.robotshop.com)). AT is the abbreviation of “Attention” and command lines starts with "AT". Because of that they are called as AT commands. The most known commands that are used to control wired dial-up modems, like ATD (Dial), ATH (Hook control), ATA (Answer) and ATO (Return to online data state) that also supported by GSM/GPRS modems and mobile phones. GSM/GPRS modems support an extended AT command set that are specific to the GSM technology. The SMS-related extended AT commands like AT+CMGS (Send SMS message), AT+CMGL (List SMS messages), AT+CMSS (Send SMS message from storage) and AT+CMGR (Read SMS messages).The starting "AT" is the prefix of AT command that informs the modem about the start of a AT the command line. That is not a part of the AT command. For example, In ATD command, D is the name of actual AT command and in AT+CMGS,

+CMGS is the actual AT command name. Some of the tasks over GSM/GPRS modem or mobile phone that can be done using AT commands are:

- To get basic information about the GSM/GPRS modem: name of manufacturer(AT+CGMI), IMEI number (International Mobile Equipment Identity)(AT+CGSN), model number (AT+CGMM), and software version (AT+CGMR)can call via command line.
- To get basic information about the subscriber: subscriber number MSISDN(AT+CNUM) and MSC identification IMSI (International Mobile Subscriber Identity) (AT+CIMI).
- To get current status of GSM/GPRS modem: mobile phone activity status(AT+CPAS), radio signal strength (AT+CSQ), mobile network registration status(AT+CREG), battery charge status (AT+CBC).
- To establish a data connection to a remote modem or voice number (ATD, ATA, etc).
- To receive and send fax (ATD, ATA, AT+F\*).
- To access SMS related features: to send (AT+CMGS, AT+CMSS), to write(AT+CMGW), to read (AT+CMGR, AT+CMGL), to delete (AT+CMGD) SMS messages. To obtain notifications about newly received SMS messages(AT+CNMI).
- To Read (AT+CPBR), to write (AT+CPBW) or to search (AT+CPBF) phonebook entries.
- To perform security related tasks, such as sim lock opening, checking or closing facility (AT+CLCK), and to changing passwords (AT+CPWD).
- To set or change the configurations of the GSM/GPRS modem: to change the carrier (AT+COPS), bearer service type (AT+CBST), SMS center address(AT+CSCA), radio link protocol parameters (AT+CRLP) and to storage set of SMS messages (AT+CPMS).
- To save and restore configurations of the GSM/GPRS modem: to save (AT+CSAS)and to restore (AT+CRES) settings SMS messaging like SMS center address.

Note that manufacturers usually do not implement all the AT commands, parameters and values in their mobile phones. Also, the behavior of the AT commands may be differ from manufacturer to manufacturer. In general, GSM/GPRS modems designed for GSM network have better support of AT commands than the mobile phones. In addition, some AT

commands requires support of GSM network providers. Although SMS over GPRS are supported by most of mobile device with the +CGSMS command (Select Service for MOSMS Messages), it does not support the transmission of SMS over GPRS if the GSM provider does not support this feature ([www.zeeman.de](http://www.zeeman.de))

### 3.5. Hardware Design

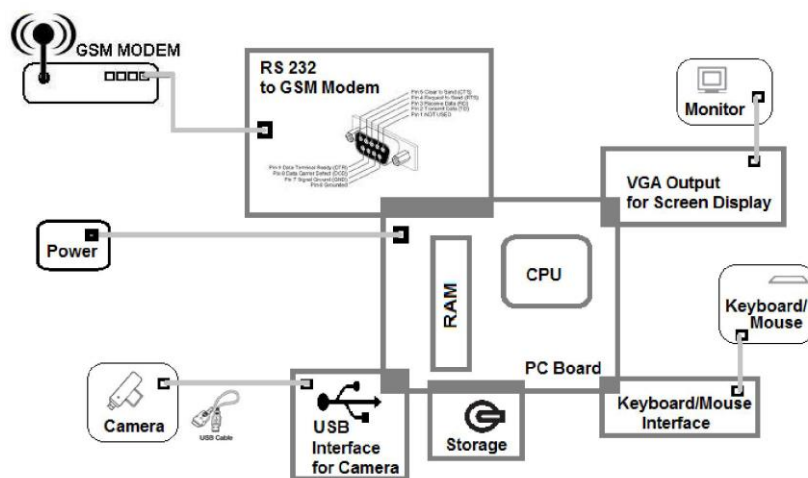
In thesis, PC board and its peripherals are used to reduce complication. PC board is prepared to load windows XP system and my software running on XP which will control all the process. The basic block diagram and design are shown in the following Figure 3.3.

The hardware specifications used in design are:

- PC Board
- Keyboard and Mouse used to control software
- Monitor connected to display outputs, and making visual settings.
- HDD connected to store necessary software
- 1 GB of RAM assembled.
- 300Watt Power Supply Unit (PSU) used to support necessary energy for both GSM

Modem and PC Board with Peripherals

- GSM Modem connected to board using RS232 port to send SMS via GSM networking secure and trusted way.
- Basic USB camera connected to board to capture necessary images. (But, good quality camera will give the best performance; tried to explain on result part.)



**Figure 3.3:** Basic hardware schematic

As shown in hardware schematic, Camera connected to PC board via USB interface to capture video frame. Each captured frame processed by software which runs on PC Board/CPU /Ram. The process controlled via keyboard and mouse via PS/2 interface. The output, visual alert and changes appear on the VGA monitor via VGA interface. If any changes considered and alert triggered, software calls the GSM Modem routine to fire SMS message. Vice-versa, any response or query can be get via GSM Modem to call appropriate special functions (Yavuz, 2002).

## **CHAPTER 4**

### **SOFTWARE DESIGN AND IMPLEMENTATION**

#### **4.1. Flowchart of Vision Based Patient Room Surveillance System**

When software runs, the modem parameters are set via AT commands and RS232 port, in that way modem should be ready to communicate GSM network. If it is failed to connect to network, software will wait until GSM Network available. After negotiate GSM network, the frames start to be capture via Video camera and usb port. When the first frame captured it should be set and store as background image. On the next step, it should be converted to grayscale as described in chapter 2. If any distortion on the medium or usb communication (which is unusual, but worst case scenario), median filtering applied to gray scale background image described like in the chapter 2 to get a smooth and undistorted image. This flow can be call any time of the process if some changes happen on the background which is not related to interest area. As an example, if any changes happen on illumination on the day time, background can be reset using this procedure.

Secondly, Interest region coordinate must be set as square (x1,y1) and (x2,y2). After each region entered, software start to check is there any changes happened on that region. If somewhere on the detection process any terminate interrupt taken, flow jump to terminate step and detection process will be end.

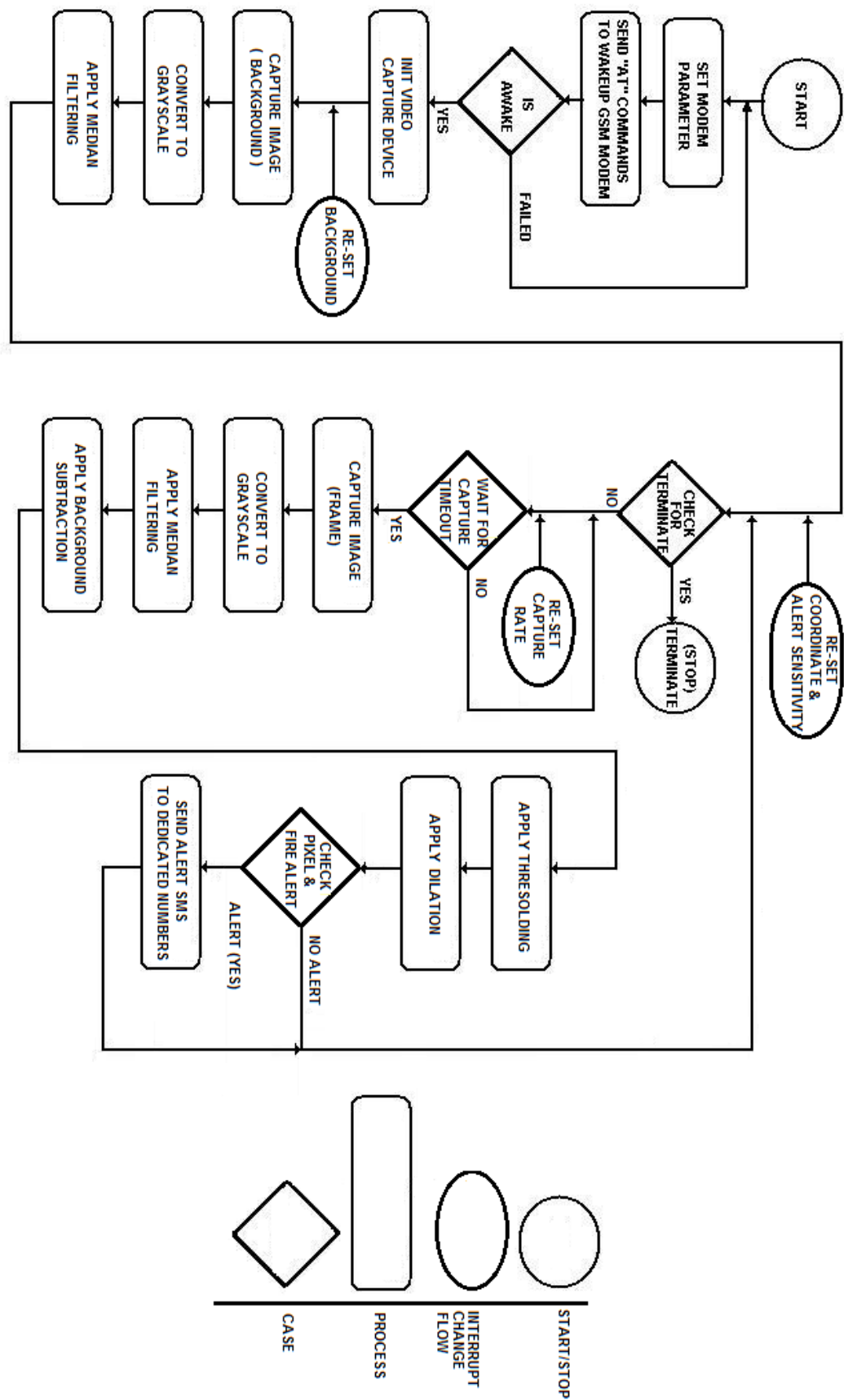
After each frame captured from the camera, will be converted to grayscale. Then median filtering rules should be applied to gray scale captured frame for smoothing and undistorted imaging. Then Background Subtraction technique will apply to find any changes between frames. But to make a decision that is not enough because the pixels must contain 1(one) s and 0(zero) s. So thresholding will apply to subtracted frame. In that way, the entire pixel on subtracted frame should be contained 1 and 0. Finally, Dilation will apply to make the result much more clear for the decision process.

After this stage, the software will check any percentages of whites (1-one-s) available in the given square coordinate. If it is more than threshold it will fire an appropriate alert. Alert may be a text message to inform someone via GSM modem or activating any electronic switch to connect to the hardware. Also, that is possible to make a query to the system via SMS Text to take the information about the current status of patient room. And that is possible to run any

hardware via SMS text. But in the case stated on thesis focused on mono direction alert, so all alerts are belongs to sending SMS text message to warn or inform someone for the emergency case. But if that is interesting for the future work, necessary codes can be found in the appendix as an example.

Finally, the flow loop turn back to the new frame capture process. That is continuing up to any interrupt request via keyboard or mouse. All the flow of software like shown in the Figure 4.1 on the next page.

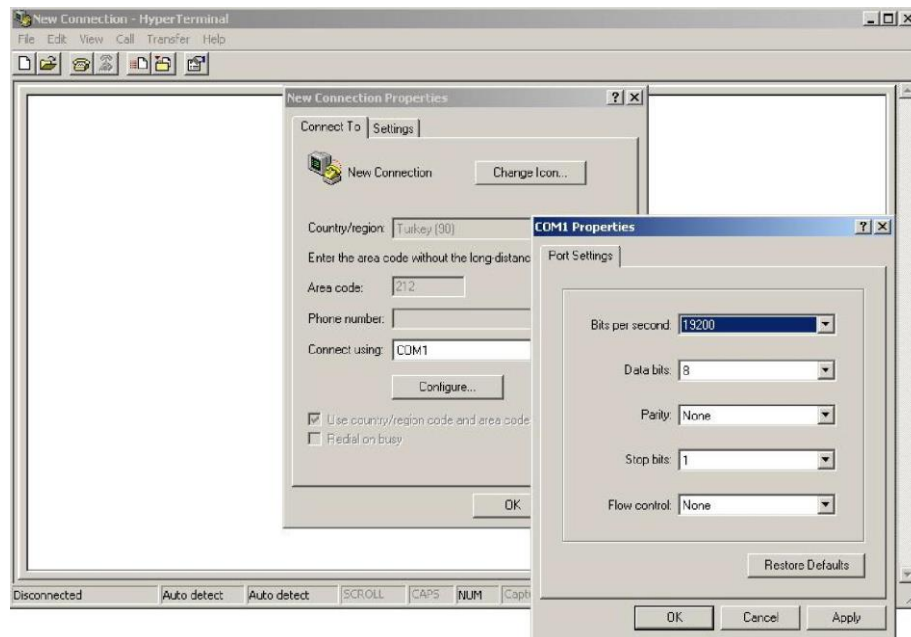




**Figure 4.1:** Basic software schematic flowchart

## 4.2. GSM Modem Setup

As shown in Figure 4.2, first need to set GSM modem parameter appropriately. In this part, settings of GSM modem is connected to Com1 (RS232, Com port#1), speed of Com1 port set to 19200 baud rate, and none parity. Then SMSC number must be set with the related network, and PIN of sim card must be entered to the appropriate field.



**Figure 4.2:** HyperTerminal terminal settings

When the program run, it should send related AT commands explained on part II to the GSM terminal and retrieve the results.

```

JOY - HyperTerminal
File Edit View Call Transfer Help

ATE10
OK
AT+CPIN:
+CPIN: SIM PIN

OK
AT+CPIN="1234"
OK
AT+CREG:
+CREG: 0,2

OK
AT+CREG=10
OK
AT+CPMS:
+CPMS: "SM",0,50,"SM",0,50,"SM",0,50

OK
AT+CSMS:
+CSMS: 0,1,1,1

OK
AT+CSMA="+905428700044"
OK
AT+CMGF=10
OK
AT+CSDH=10
OK
AT+CNMI=?
+CNMI: (0-3),(0,1),(0,2,3),(0,2),(1)

OK

CONTINUED From Log Files
06.12.2012 14:18:45: AT+CNMI:
06.12.2012 14:18:45: +CNMI: 0,0,0,0,1
06.12.2012 14:18:45: OK
06.12.2012 14:18:46: AT+CNMI=2,0,0,0,10
06.12.2012 14:18:46: OK
06.12.2012 14:18:46: AT+CMGL0
06.12.2012 14:18:46: OK
06.12.2012 14:18:46: AT+CGMM0
06.12.2012 14:18:46: SIEMENS
06.12.2012 14:18:46: OK
06.12.2012 14:18:46: AT+CGMM0
06.12.2012 14:18:47: AT+CGMM0
06.12.2012 14:18:47: TC35
06.12.2012 14:18:47: OK
06.12.2012 14:18:47: AT+CGMR0
06.12.2012 14:18:47: REVISION 03.10
06.12.2012 14:18:47: OK
06.12.2012 14:18:47: AT+CGSM0
06.12.2012 14:18:47: 520338414029855
06.12.2012 14:18:47: OK
06.12.2012 14:18:47: AT+CSMP=17,,0,2400
06.12.2012 14:19:34: OK
06.12.2012 14:19:34: AT+CMGS="+905428745949",1450
06.12.2012 14:19:37: > First Test-Message
06.12.2012 14:19:37: +CMGS: 74
06.12.2012 14:19:37: OK
06.12.2012 14:19:37: OK

Connected 00:00:38 Auto detect Auto detect SCROLL CAPS NUM Capture Print echo

```

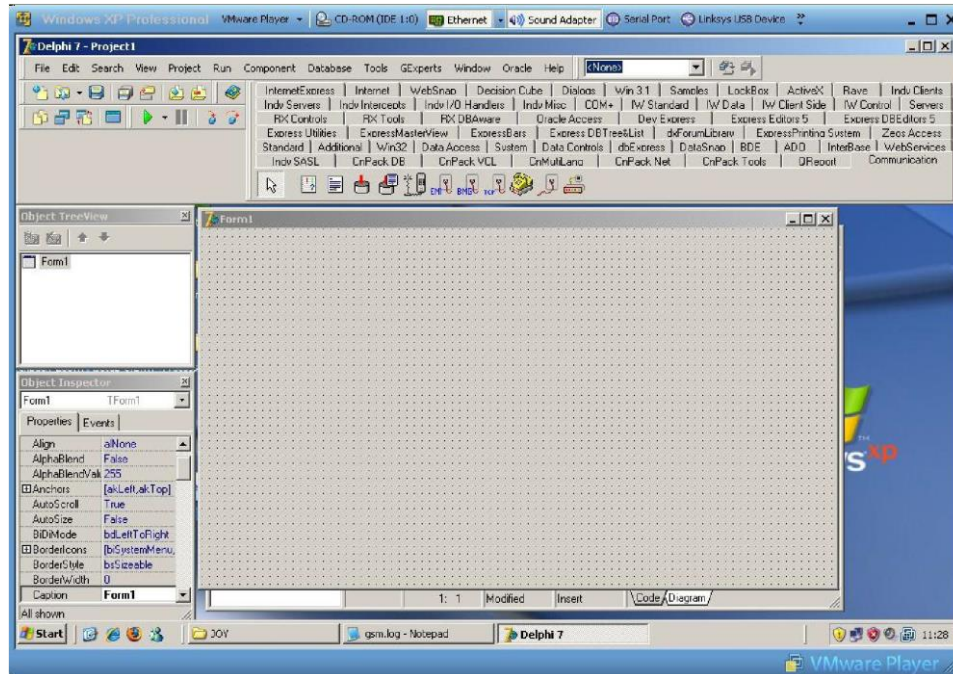
**Figure 4.3:** HyperTerminal GSM Modem session and Log view

As shown on Figure 4.3 tests completed and GSM modem fully worked with AT Commands. Then it starts to implement on Delphi. As mentioned before, Delphi has very large of communication library for RS232 Serial and modem communication. According to result returned to the PC, it should understand GSM modem is ON and registered to GSM network properly.

### 4.3. Delphi Integrated Development Environment (IDE)

Delphi is a very easy to design to program view and coding in Pascal. Unlike other traditional visual programming environment, at the top of the main menu more settings can make for the IDE. Delphi IDE combined of a number of separate windows and the main window positioned on the top of the screen which can be a list as the main menu, component palette and toolbar menu. The name of the current project displays on the title bar of the main window. On the center, a form to displays when the program runs. On the right side, object Inspector available to make settings on the object on the form. Below the main menu, large component plate (library) available to include objects into the project. The Component Palette contains a component which are grouped by the function they perform. Each page tab displays

a group of icons related to the components. For example, like an edit box, a button or scroll box components are placed on the Standard and Additional pages. As shown in the figure below, "Communication" tab appears with the related components.



**Figure 4.4:** Delphi IDE initial view

#### 4.4. Delphi Video Grabber

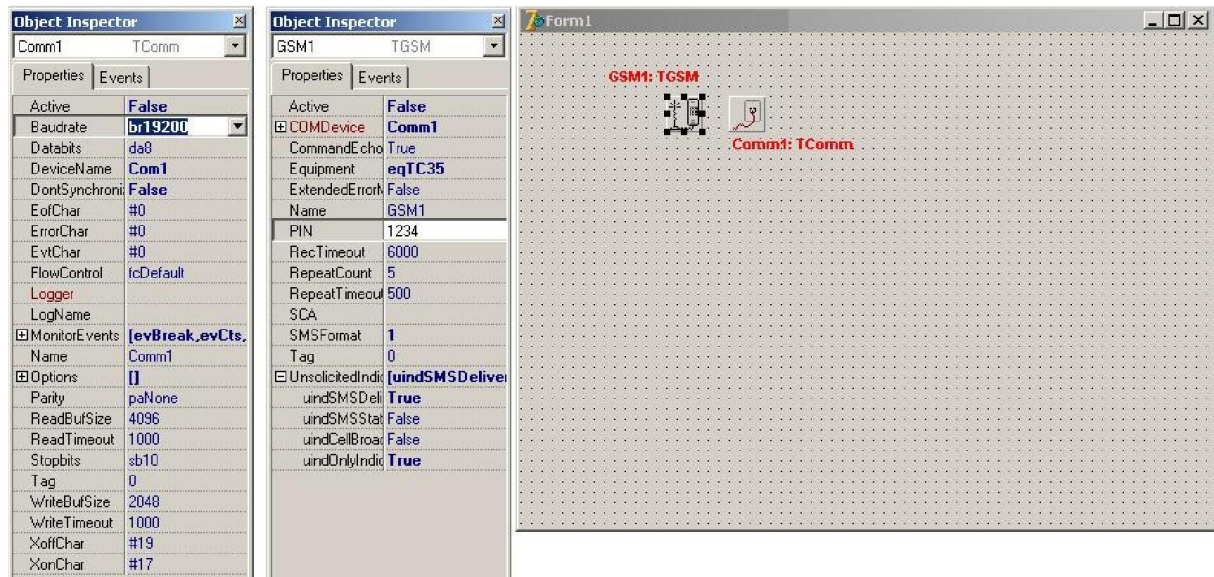
In the software application, to capture the video frames from the camera, "tscap32" Delphi Video Capture Component has been used. "tscap32" is a very famous Delphi component that used to capture video streams. It makes easy:

- Watch a live video source
- Capture videos
- Capture snapshots

Basically, it uses Microsoft's Video For Windows API and gives to the programmer an easy-to-use tool to handle video input. This tool was developed for medical purposes at the beginning and was used in a range of medical applications. So the code itself is proven and stable. But most important think to prefer that it is a freeware.

## 4.5. Delphi GSM Modem Communication

In the design, as shown like in the figure 4.4, a new empty application created, and then *TGSM* and *TComm* components placed on that form. Both components located on *Communication* palette.



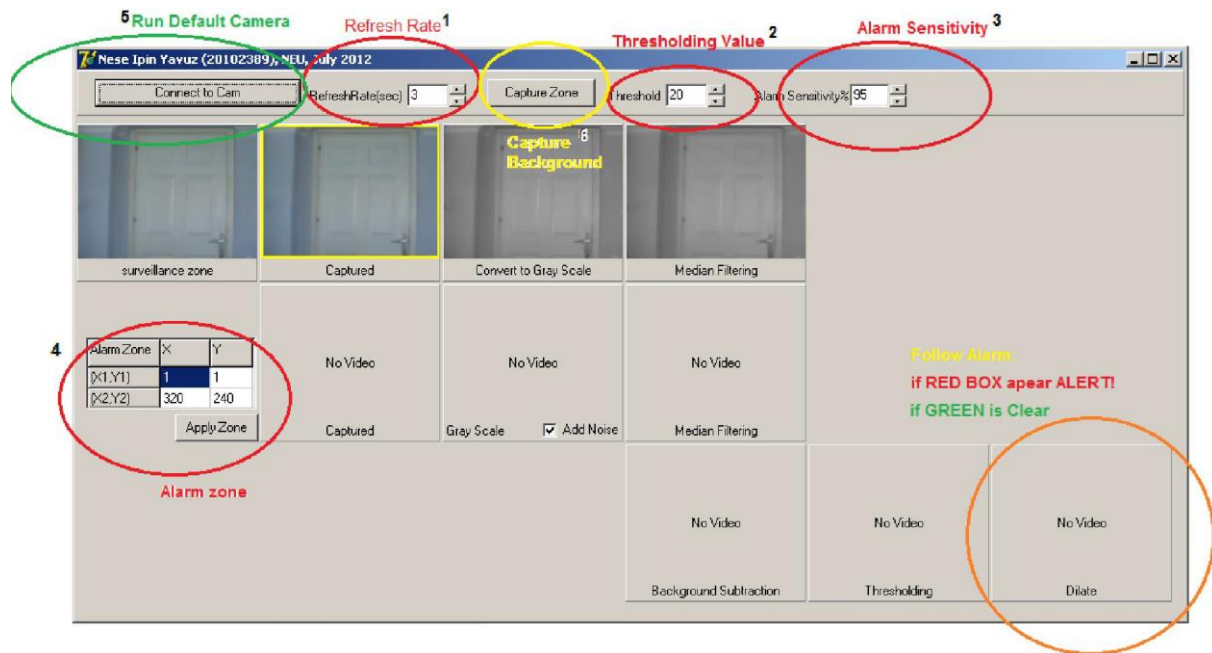
**Figure 4.5:** Initial delphi project view and components settings

On the object inspector of an object, the settings made shown like in the Figure 4.6. When the program runs GSM and COMM components are acts like the settings on object inspector properties tab. If any event occurs on the modem or com port, it reflects (fires) an event which can be shown in the object inspector events tab. On the events tab, Pascal function can be specified to execute. Also, if any case available to notify someone the procedure send SMS will be called with necessary field setup on the function which are GSM Number and Alert Message. That is not the case in the thesis but vice-versa when SMS-MT received via modem can be fired (runs) any machine on the environment via parallel port interfacing.

## 4.6. Software Flow

When the software initialized the camera and starts to capture the background for background subtraction technique. Then program waits for to press set area button. If any area set by a person who controls the software, the process starts to detect changes on that area. For the area which selected by the person, following step by step procedures apply to detect any changes. The block diagram of the process can be summarized like shown in the Figure 4.7.





**Figure 4.6:** Software design view

To run the software, some parameter must be set properly to run process correctly. When the software runs, it will set the modem on the background. And Interface will pop up like on the figure 4.6. When interface appear, the following must be set:

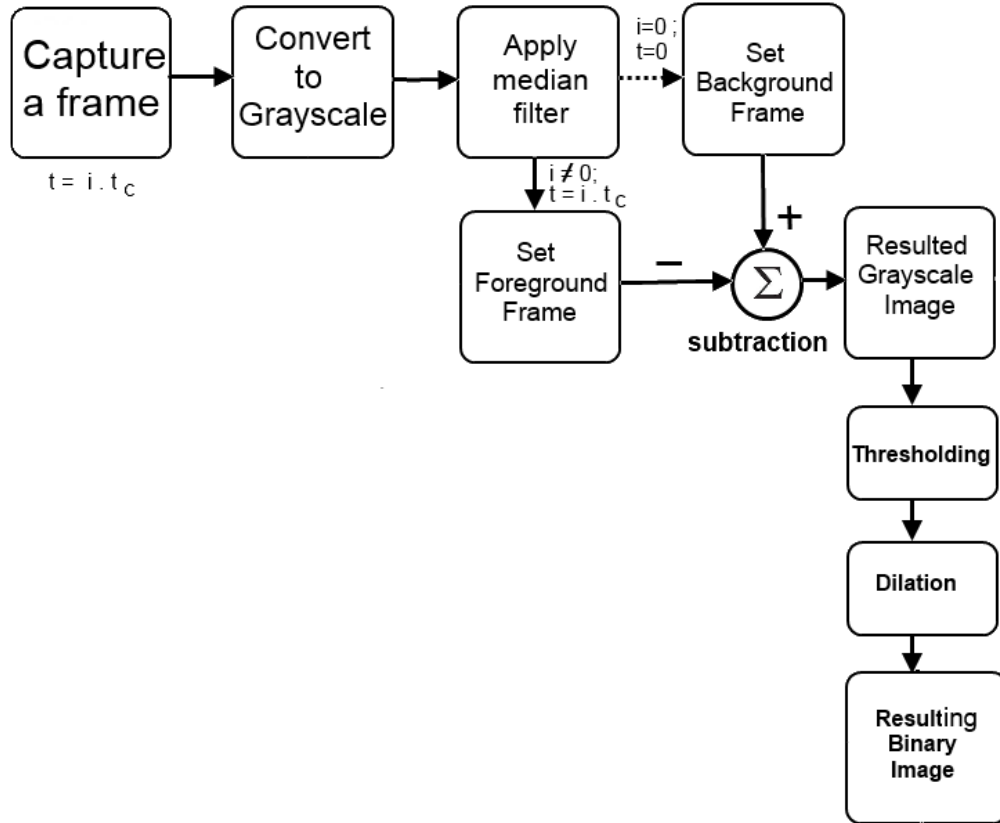
1. Refresh Rate (delay to capture next frame)
2. Thresholding level
3. Alarm sensitivity (percentage of white pixels)
4. Coordinate for the surveillance

After set all those parameters properly, "Connect to Cam" button will start to process. On each level of the process will be seen on the interface view.

- Step by step procedure contains the following key points:
- Capture background image ( $t=0$ )
- Apply Greyscale to background image ( $t=0$ )
- Capture frame image in given interval ( $t=t_i$ )
- Apply Greyscale to captured frame image ( $t=t_i$ )
- Subtract background image from captured frame ( $t=t_i$ )
- Apply thresholding technique ( $t=t_i$ )
- Apply dilation technique ( $t=t_i$ )
- Check square coordinates given

- If any change (white pixel ratio) appear alert to dedicated group/person via SMS using GSM modem.

Where  $t$ :time,  $t_i$ : time on  $i$ th interval,  $i$ :  $i$ th interval,  $t_c$ : interval time constant



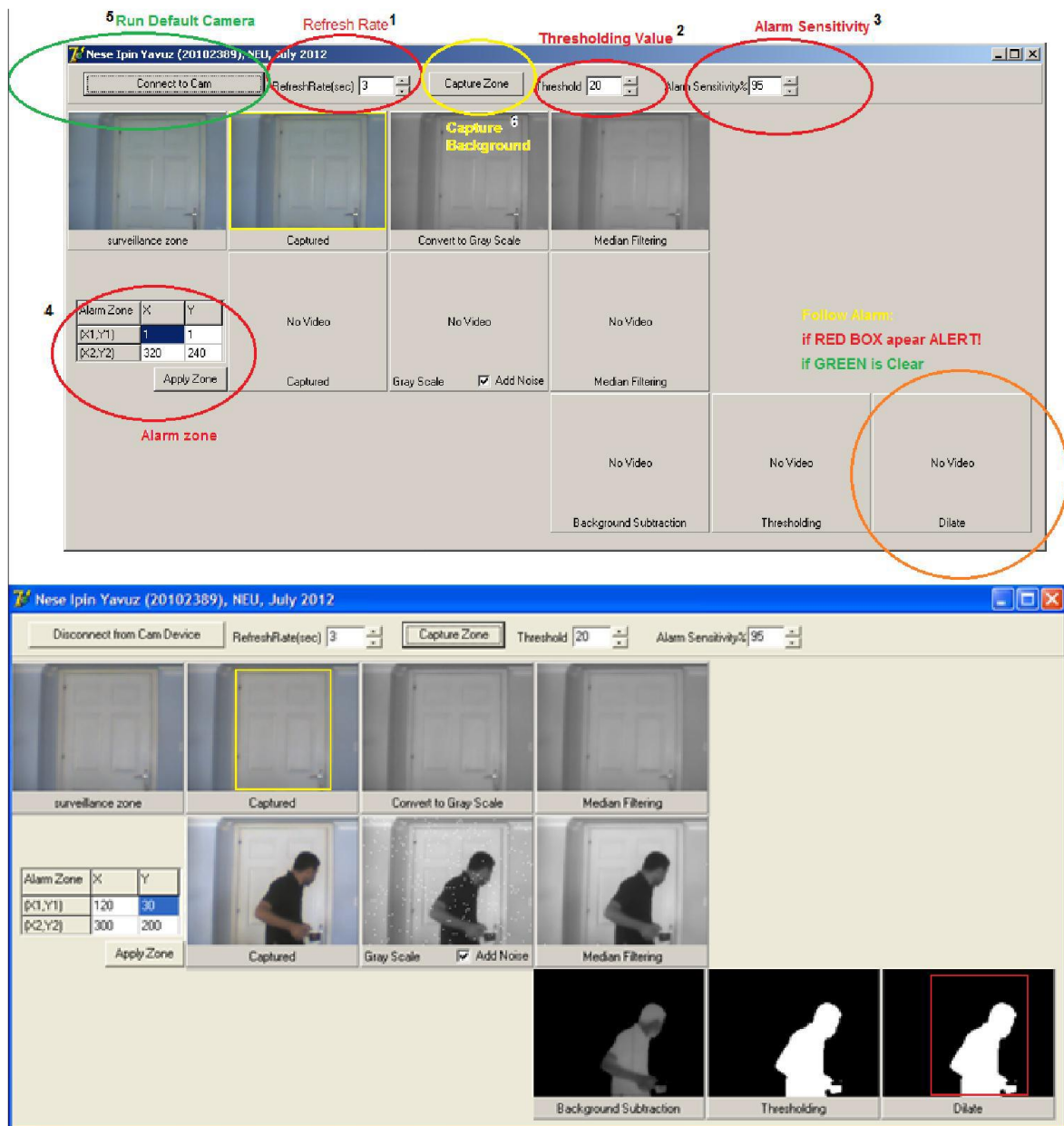
**Figure 4.7:** Step by step process block diagram of the flow

## CHAPTER 5

### RESULTS AND DISCUSSIONS

#### 5.1. Basic Scenario

After set up hardware and software basic tests made at home environment. The basic scenario was carried out for security of any place. The section will be monitored; using an usb camera to make the project easily.

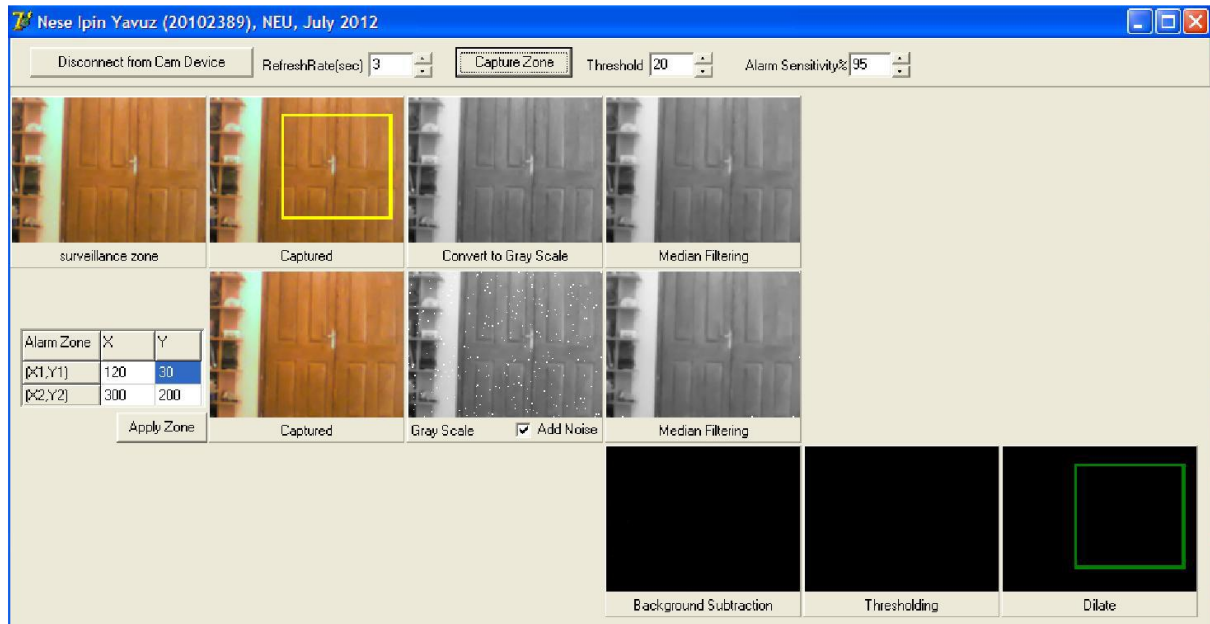


**Figure5.1: Basic scenario**

Video taken from the camera should be extracted than will be processed to find any action has been detected in the region of interest. Background subtraction technique was the most



involved technique used in the thesis. After retrieving the coordinates of action, the software will examine whether the action is in the region of interest or out. If it is in the interest area Red (alert) square has drawn around the interest area if not green (secure) square appears that likes in the Figure 5.1. Another good example has shown in the Figure 5.2.



**Figure 5.2:** Snapshot 1 for basic scenario



**Figure 5.3:** Snapshots 2 for basic scenario

Yellow square shows area of interest. In snapshot 1; green area shows there is no activity detected. After a while object (lady) appear at the door and because of activity detected, the area shows in the red.

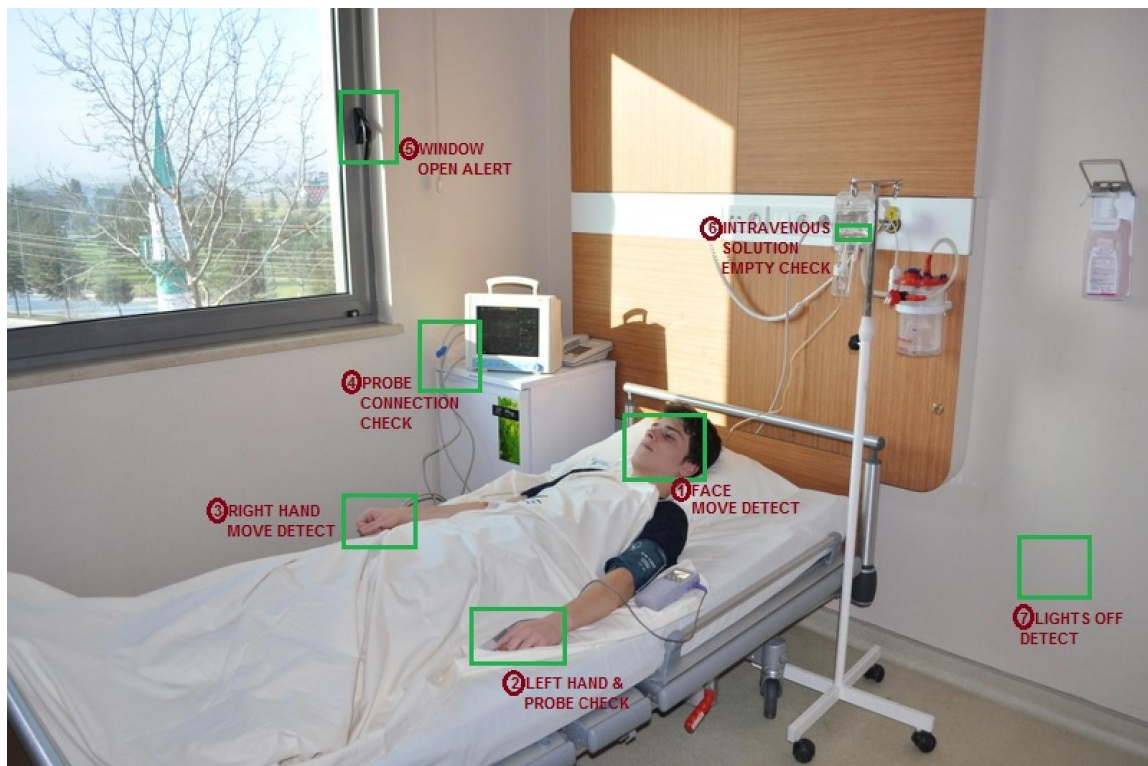
Snapshot 1: No activity

Snapshot 2: Object appears at the door.

In the case, there is not enough noise to see the median filtering effect, so additional noise added to captured frame than applied median filter to remove it as shown in snapshot 2.

## 5.2. Patient Room Scenarios

After complete basic steps to detect the object in the area of the interest, complex software developed to detect multiple area and sending SMS alert over gsm network. Using advantages of GSM network alert should reach to the dedicated person or group in secure and trusted way.



**Figure 5.4:** Patient room scenario

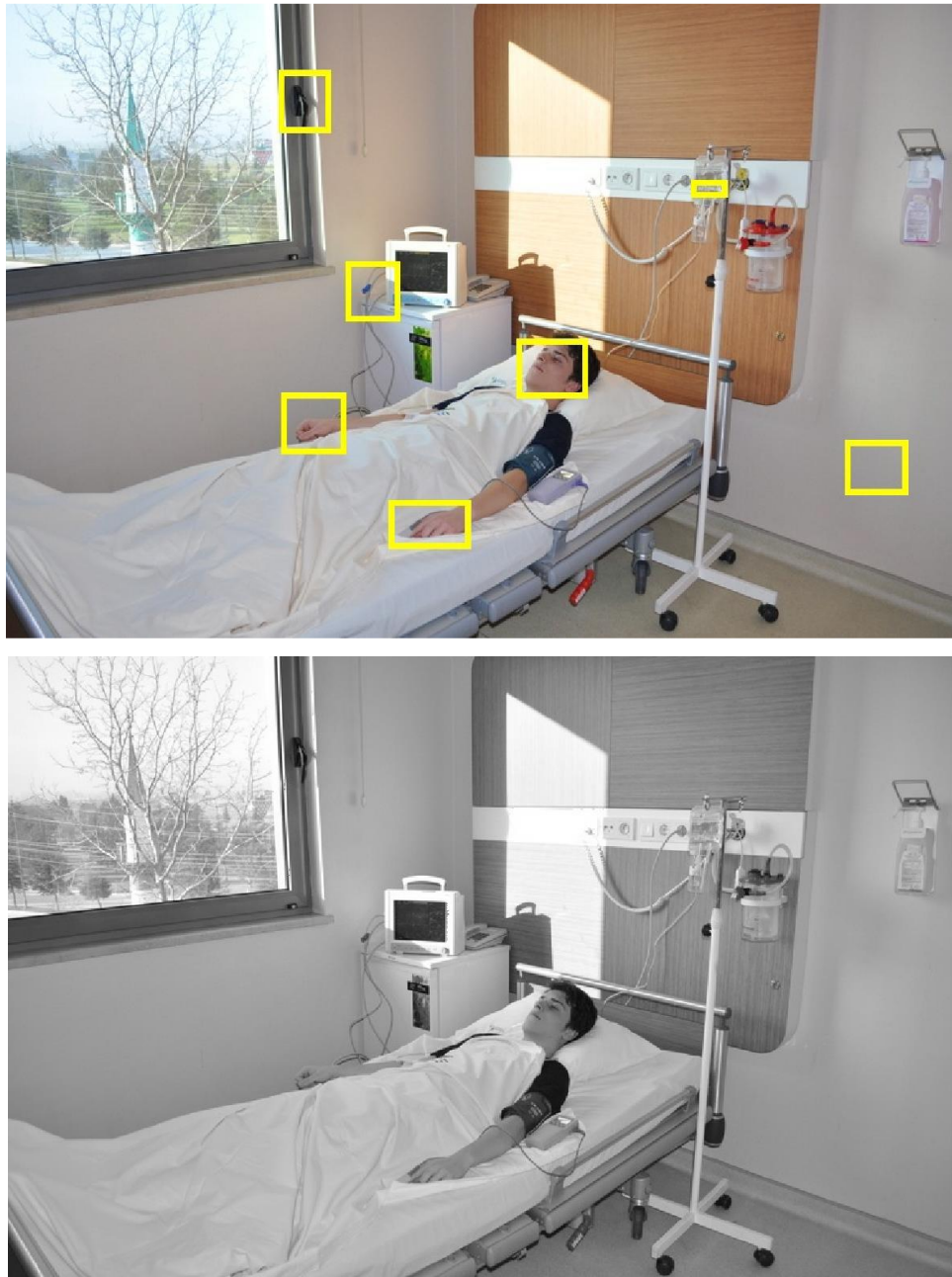
In this scenario, seven square coordinates defined for areas of interest. Those coordinates defined as:

1. Face movement detection
2. Left hand movement detection & Oxygen probe connection check
3. Right hand movement detection

4. Probe connection check
5. Window opened alert
6. Intravenous solution empty alert
7. Lights off or Electricity cut off check

### 5.3. Patient Room Scenarios step by step application

1. Background captured shown in the Figure 5.5



**Figure 5.5:** Snapshot patient room background and gray scale conversion



2. Right hand movement easily seen like in the Figure 5.5 and gray scale conversion on Figure 5.5 following snapshot. But also there are some other movements but not easily seen in the figure. On final stage they shown and the software detected there are more than one move.

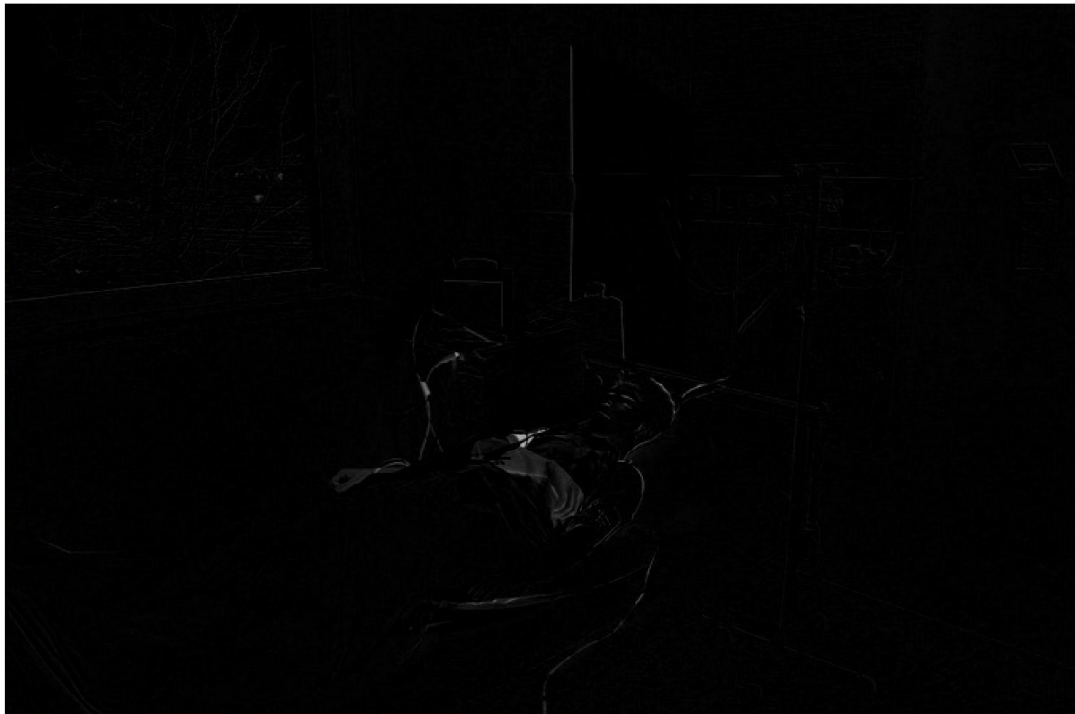


**Figure 5.5:** Snapshot 1 for patient room right hand moved and snapshot in gray scale conversion



**Figure 5.6:** Snapshot 2 for patient room right hand moved and snapshot in gray scale conversion

3. Background subtraction applied and results like shown in the Figure 5.6.



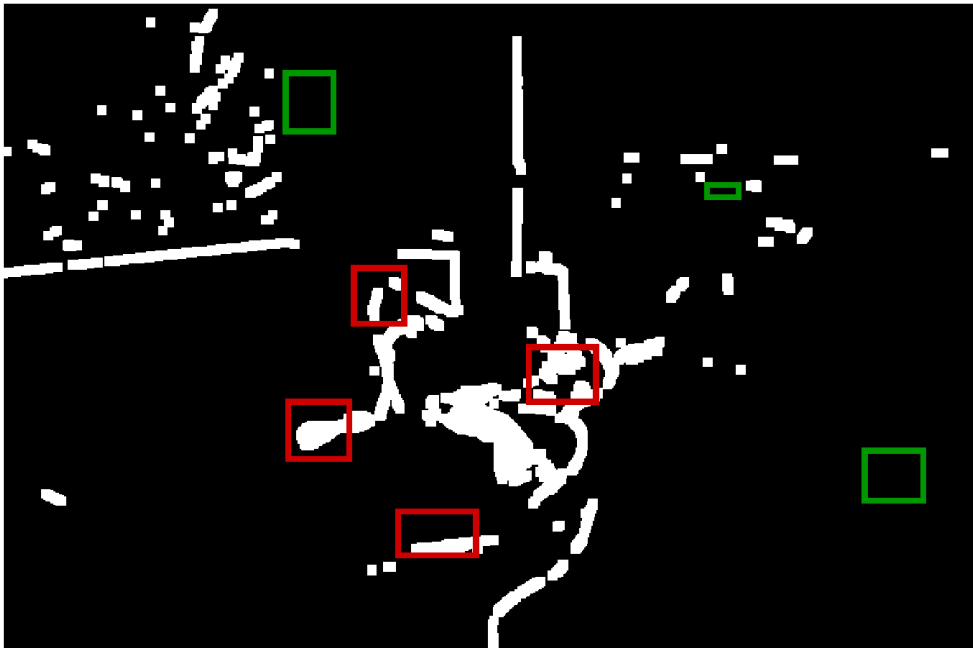
**Figure 5.7:** Snapshot patient room background subtracted

4. When the 3x3 thresholding procedure applied, result like shown in the Figure 5.7.



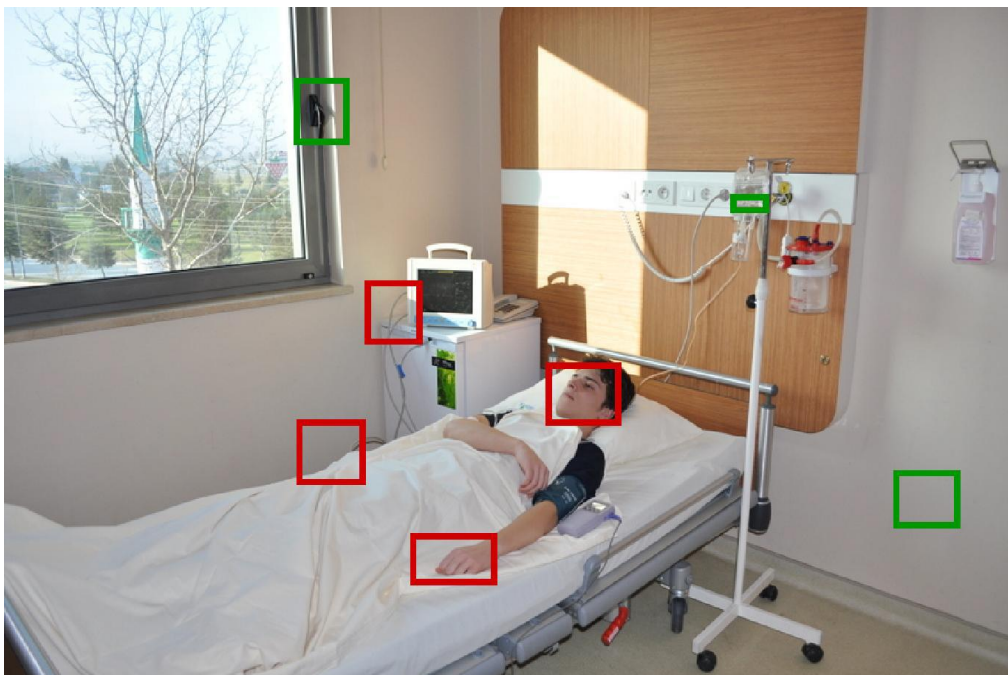
**Figure 5.8:** Snapshot patient room background subtracted and thresholding applied

5. Before the final stage dilate procedure (8 pixel) applied to get more certain results which shown in the Figure 5.8.



**Figure 5.9:** Patient room BGS, thresholding and 8 pixel dilation applied

6. Finally four alerts fired which are 1, 2, 3 and 4 described in figure 5.3. There are no change on coordinates 5, 6, and 7. Alerts fired like shown in Figure 5.9



**Figure 5.10:** Four alerts fired

7. As shown in figure 5.9, four types of SMS send over the GSM network, and interested person notified. Types of alerts are:

- a) “Face movement on room 201” to doctor.
- b) “Oxygen sensor probe problem on 201” to sick nurse.
- c) “Right hand moved on 201”, sick nurse and doctor.
- d) “Monitor probe disconnected on 201” to sick nurse.

Those alerts appear in red square shown in the figure 5.9. Unfired alerts shown in the green square which are:

- a) “Window opened in room 201” to security and sick nurse.
- b) “Empty intravenous solution detect on 201” to sick nurse.
- c) “Lights off or Electric cut in room 201” to sick nurse and maintenance guy.

Finally, after demonstration on “Özel Medi-ce İnegöl Hastanesi”, the opinion of the specialists like in the following:

- Op.Dr.Tunç Polat Edizsoy said that “if this kind of systems is running in our patient rooms, we do not need any attendant, which reduces medical complication and costs”, orthopedist and traumatology specialist.
- Op.Dr.Özgür Oktay Nar said that “As shown in the demonstration, if this system lives in our hospital, more jobs can make with fewer people that reduce the costs. Also, most important thing is, It helps to reduce mistake and tolerance depends on the tired”, orthopedist and traumatology specialist.

#### **5.4. Alarm Sensitivity and Error Calculation**

Alarm sensitivity is a threshold parameter that shows the minimum white pixel ratio percentage in the square area to fire an alert. To detect an alert on the given square area alarm level is calculated by

$$\text{Alarm Level} = \text{White Pixel in the Square} / \text{Total Pixel in the Square} * 100 \quad (5.1)$$

and rule applied to fire an alert

$$\text{if Alarm Level} > \text{Alarm Sensitivity then fire an appropriate alert} \quad (5.2)$$

In the patient room application, luminance will not vary with time because of an isolated room for patient health and it is always fixed. So the possible false alert is zero when the alarm sensitivity set to 10 percent in the application that was eliminates all the false alerts error on continues test.



## **CHAPTER 6**

### **CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK**

In this thesis, tried to explain image processing techniques and GSM communication can be combined for the successful use of vision based patient room surveillance system. The result is taken from demonstration show that it is an efficient, robust and mobile for this application. On the final stage of demonstration, all the staffs at the hospital agreed that it is a very helpful system for any hospital to reduce costs and medical complexity. Also, they agree that it should help to reduce human mistake, and application tolerance depends on the tired. On the other hand if this kind of application appears on the hospitals, more services can be given with fewer people who also reduce the costs for patient, hospital and also the government. Also, the application costs are very feasible and absolutely nothing when the patient lives thought.

Finally, a live application of the thesis is successfully completed, and the result show that it is making the life easy for the people who want to control to the patient room in secure, efficient, cheap and from very long distances. Moreover, the magic is that, there is no any alternative (except other GSM services) to the surveillance room (without wire) too far away from room point. In my opinion, this kind of applications will be very useful in the future.

For the further work, some amendments can be made on the application for automated triggering to gain extra time for emergency cases. For example, after any staff notified via SMS-MO for emergency case, Staff can be given a command to activate any alert or machine replying the text message which has been received.

## REFERENCES

Anderson, M., Motta, R., Chandrasekar, S., & Stokes, M. (1996, January). Proposal for a standard default color space for the internet—srgb. In *Color and imaging conference* (Vol. 1996, No. 1, pp. 238-245). Society for Imaging Science and Technology.

AuntMinnie; Internet site for radiologists and related professionals in the medical imaging industry. Revised on July 27, 2013, <http://www.auntminnie.com/index.aspx?sec=ser&sub=def&pag=dis&ItemID=92420>.

Benezeth, Y., Jodoin, P. M., Emile, B., Laurent, H., & Rosenberger, C. (2008, December). Review and evaluation of commonly-implemented background subtraction algorithms. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* (pp. 1-4). IEEE.

Channa, M. I., & Ahmed, K. M. (2010). Emergency response communications and associated security challenges. *arXiv preprint arXiv:1010.4887*.

Datasheet Archive. A data set archive for researchers. Revised on November 24, 2013. <http://www.datasheetarchive.com/-20/siemens%20modem%20tc35-datasheet.html>.

Engineers Garage (2010). Collection of inspiring engineering creations from all around the world. Revised on July 08, 2013, <http://www.engineersgarage.com/articles/image-processing-tutorial-applications>.

Hosseini, M. P., Soltanian-Zadeh, H., & Akhlaghpour, S. (2012). Detection and severity scoring of chronic obstructive pulmonary disease using volumetric analysis of lung CT images. *Iranian Journal of Radiology*, 9(1), 22-7.

Huang, T., Yang, G., & Tang, G. (1979). A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(1), 13-18.

Hubble Telescope Official Web Site. Revised on March 03, 2013, [https://www.spacetelescope.org/projects/fits\\_liberator/improc](https://www.spacetelescope.org/projects/fits_liberator/improc).

Indian Institute of Technology Kanpur. Revised on November 07, 2013. [http://home.iitk.ac.in/~luckys/cs300A/wikipedia/Background%20subtraction\\_%20Difference%20between%20revisions%20-%20Wikipedia,%20the%20free%20encyclopedia.pdf](http://home.iitk.ac.in/~luckys/cs300A/wikipedia/Background%20subtraction_%20Difference%20between%20revisions%20-%20Wikipedia,%20the%20free%20encyclopedia.pdf).

International Engineering Consortium. Wireless Short Message Service. Revised on December 10, 2012. [ftp://www.comptek.ru/pub/Dialogic/doc/tutorial/wire\\_sms.pdf](ftp://www.comptek.ru/pub/Dialogic/doc/tutorial/wire_sms.pdf)

Johnson, S. (2006). *Stephen Johnson on Digital Photography*. O'Reilly Media, Inc..

Malan, D., Fulford-Jones, T., Welsh, M., & Moulton, S. (2004, April). Codeblue: An ad hoc sensor network infrastructure for emergency medical care. In *International workshop on wearable and implantable body sensor networks* (Vol. 5).

Maldague, X. P. V., Jones, T. S., Kaplan, H., Marinetti, S., & Prystay, M. (2001). Chapter 2: Fundamentals of Infrared and Thermal Testing: Part 1. Principles of Infrared and Thermal Testing. *Nondestructive Handbook, Infrared and Thermal Testing*, 3, 718.

Official Website of the Department of Computer Science at Princeton University. Revised on November 11, 2013. <http://www.cs.princeton.edu/~pshilane/class/mosaic/>.

Seemann, T. (2003). *Digital Image Processing Using Local Segmentation*. Monash University.

Seimens Mobile. TC35 Hardware Data Sheet, Revised on November 10, 2013 <http://www.robotshop.com/media/files/pdf/datasheet-gsm-tc35.pdf>

Seminar Topics Project Ideas on Computer Science, Electronics, Electrical, Mechanical Engineering, Civil, MBA, Medicine, Nursing, Science, Physics, Mathematics and Chemistry. Revised on June 03, 2013, <http://seminarprojects.com/s/moving-object-tracking-and-velocity-determination-project-report>.

TE Connectivity. An online support website which helps engineers to transform their ideas to innovation. Revised on November 11, 2013. <http://www.te.com/industries/networking.html>.

Technology Robotix Society.Revised on March 03, 2013, [http://www.robotix.in/tutorials/category/opencv/noise\\_reduction](http://www.robotix.in/tutorials/category/opencv/noise_reduction).

The Electronic Intifada; Independent online news publication and educational resource focusing on Palestine, its people, politics, culture and place in the world (2001).Revised on June 18, 2013, <http://electronicintifada.net/content/un-environment-agency-set-begin-aerial-surveillance-lebanese-oil-spill/2723>.

The JFE Group. Company which aims contributing to society with the world's most innovative technology. Revised on September 14, 2013, <http://www.jfeholdings.co.jp/en/investor/zaimu/ar/nkk/>.

The University of Auckland, Department of Computer Science. Revised on September 14, 2013, <http://www.cs.auckland.ac.nz/compsci373s1c/PatricesLectures/2011/CS373-LN28->

UbiNetics Ltd. GSM AT Command Set, Revised on November 20, 2013. <http://www.zeeman.de/wp-content/uploads/2007/09/ubinetics-at-command-set.pdf>

United Kingdom Open Government License for Public Sector Information. Revised on June 14, 2013, <https://www.gov.uk/government/organisations/office-of-surveillance-ommissioners>.

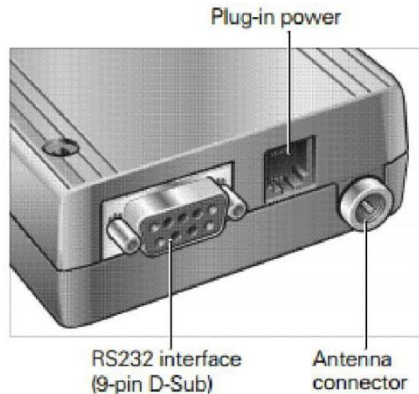
Web.com Group, Inc. Leading Mobile In Site On Internet. Revised on November 11, 2013, [http://www.mobilein.com/SMS\\_tutorial.pdf](http://www.mobilein.com/SMS_tutorial.pdf).

Yavuz N. İ. (2002), Remote control and monitoring via SMS, Department of Electrical and Electronic Engineering, EMU.

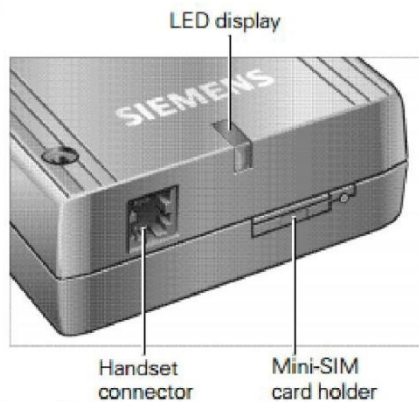
## APPENDIX A

### TC35 GSM Modem

Front view



Rear view



- Highlights**
- Dual-band EGSM900 / GSM1800
  - Data, voice, SMS and fax
  - R&TTE approval, GCF approval
  - Easy to integrate
  - Industrial interfaces
  - LED display
  - Wide input voltage range
  - Highly compact, light and powerful

- Applications**
- Fleet management
  - Teleservice
  - Security systems
  - Telematics
  - Telemetry
  - Remote monitoring
  - Remote meter reading
  - Vending machines

#### Product description

The TC35 Terminal is a compact GSM modem for the transfer of data, voice, SMS and faxes in GSM networks. Industrial standard interfaces and an integrated SIMcard reader mean it can be used rapidly, easily and universally as a dual band GSM Terminal. Its performance band width and the robust housing make it easier to quickly implement new applications in areas such as telemetry and telematics.

The features, functions and interfaces of the GSM Terminal TC35 T are described below.

The functionality of the Terminal corresponds to the features of the TC35 module, to which a SIM card reader, an RS232 interface, an analog interface for the handset interface and a greater supply voltage range have been added.

All the external interfaces of the GSM Terminal TC35 T are integrated permanently in the housing. The plug-in connections are standardised and suitable for use under vibration.

#### Features

- Product data**
- Dual-band EGSM900 and GSM1800
  - Certified in accordance with GSM phase2/2+
  - Output performance:
    - Class 4 (2 W) for EGSM900
    - Class 1 (1 W) for GSM1800
  - Control via AT commands
  - Input voltage range +8V ... +30V
  - Dimensions: 65 x 74 x 33mm
  - Weight: approx. 130g

- Certification**
- R&TTE approval
  - GCF approval

- Audio**
- Half rate (HR)
  - Full rate (FR)
  - Enhanced full rate (EFR)

- SMS**
- Point-to-point MT and MO
  - SMS cell broadcast
  - Text and PDU mode

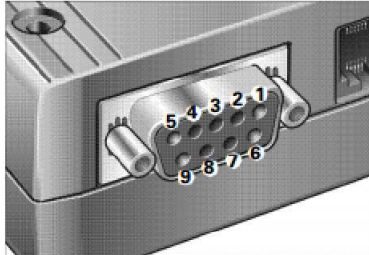
- Data**
- CSD up to 14.4kbps
  - USSD
  - Non-transparent mode
  - V.110, V.32

- Fax**
- Group 3, classes 1, 2

## Interface description

**RS232 interface** The RS232 interface is the interface for the application software and the connection to the PC. The customer application communicates with the TC35 Terminal or the TC35 GSM engine by means of ATcellular commands. The RS232 interface is implemented as a 9-pin D-Sub socket with a screw fitting.

**RS232 interface – 9-pin D-Sub**



Purpose of the connectors/connections				
Signal name	PIN	I/O	Description	Parameters
DCD	1	O	The functions correspond to those of a serial interface on the basis of a V.24 protocol.	active high >5V low <-5V
RXD	2	O		logical 1=low <-5V logical 0=high >+5V
TXD	3	I		active high >2.4V low <1.8V
DTR	4	I		active high >2.4V low <1.8V
GND	5			0V
DSR	6	O		active high >5V low <-5V
RTS	7	I		active high >2.4V low <1.8V
CTS	8	O		active high >5V low <-5V
RI	9	O		active high >5V low <-5V

### Use and operation

In order to control the Terminal and transfer data, the customer application (e.g. host computer) is connected via the RS232 cable.

## APPENDIX B

### **USurveillance.pas, Pascal Main Program Entrance and Functions Implemented for Image Processing**

```
unit USurveillance;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, tscap32_rt, StdCtrls, ComCtrls, ShellCtrls, math,
  Grids;
type
  TForm1 = class(TForm)
    pnl1: TPanel;
    tsCap321: TtsCap32;
    pnl2: TPanel;
    ButtonConnect: TButton;
    Timer1: TTimer;
    Panel1: TPanel;
    image11: TImage;
    Panel2: TPanel;
    Image12: TImage;
    Label1: TLabel;
    Label2: TLabel;
    Panel3: TPanel;
    Image0: TImage;
    Label3: TLabel;
    Panel4: TPanel;
    Image13: TImage;
    Label4: TLabel;
    pnl3: TPanel;
    Image01: TImage;
    lbl1: TLabel;
    pnl4: TPanel;
    Image02: TImage;
```

```

lbl2: TLabel;
pnl5: TPanel;
Image03: TImage;
lbl3: TLabel;
btn2: TButton;
pnl6: TPanel;
Image21: TImage;
lbl4: TLabel;
Panel5: TPanel;
Image22: TImage;
Label5: TLabel;
EditThreshold: TEdit;
Label6: TLabel;
UpDown1: TUpDown;
Panel6: TPanel;
Image23: TImage;
Label7: TLabel;
CheckBoxAddNoise1: TCheckBox;
StringGrid1: TStringGrid;
ButtonApplyZone: TButton;
Label8: TLabel;
EditAlarmSensitivity: TEdit;
UpDown2: TUpDown;
Label9: TLabel;
EditRefreshRate: TEdit;
UpDown3: TUpDown;
procedure ButtonConnectClick(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure btn2Click(Sender: TObject);
procedure ButtonApplyZoneClick(Sender: TObject);
private
{ Private declarations }
procedure LoadSurveillanceZone(FileName:String);

```



```

procedure CaptureFrame(FileName:String);
public
{ Public declarations }
end;
var
Form1: TForm1;
implementation
{$R *.dfm}
type
TRGBArray = array[0..32767] of TRGBTriple;
pRGBArray = ^TRGBArray;
var
ZoneX1, ZoneX2, ZoneY1, ZoneY2:Integer;
Procedure ConvertBitmapToGrayscale(Bitmap: TBitmap);
var
i, j: Integer;
Grayshade, Red, Green, Blue: Byte;
PixelColor: Longint;
begin
with Bitmap do
for i := 0 to Width - 1 do
for j := 0 to Height - 1 do
begin
PixelColor := ColorToRGB(Canvas.Pixels[i, j]);
Red := PixelColor;
Green := PixelColor shr 8;
Blue := PixelColor shr 16;
Grayshade := Round(0.3 * Red + 0.6 * Green + 0.1 * Blue);
Canvas.Pixels[i, j] := RGB(Grayshade, Grayshade, Grayshade);
end;
end;
procedure BackgroundSubtraction(BitmapBG,Bitmap,BitmapBS:TBitmap);
var
X,Y: Integer;

```

```

PixelColorBG, PixelColor:Byte;
begin
  BitmapBS.Assign(BitmapBG);
  for X := 0 to BitmapBG.Width - 1 do
    for Y := 0 to BitmapBG.Height - 1 do
      begin
        PixelColorBG := ColorToRGB(BitmapBG.Canvas.Pixels[X, Y]);
        PixelColor := ColorToRGB(Bitmap.Canvas.Pixels[X, Y]);
        if PixelColorBG<=PixelColor then
          PixelColorBG:=0
        else
          PixelColorBG:=PixelColorBG-PixelColor;
        BitmapBS.Canvas.Pixels[X,Y]:=RGB(PixelColorBG,PixelColorBG,PixelColorBG);
      end;
    end;
  end;
  procedure Thresholding(Bitmap:TBitmap;Threshold:Integer);
  var
    X,Y: Integer;
    PixelColor: Longint;
    Grayshade: Byte;
  begin
    with Bitmap do
      for X := 0 to Width - 1 do
        for Y := 0 to Height - 1 do
          begin
            PixelColor := ColorToRGB(Canvas.Pixels[X, Y]);
            Grayshade:= PixelColor;
            if Grayshade >= Threshold then
              Canvas.Pixels[X, Y]:=RGB(255, 255, 255)
            else
              Canvas.Pixels[X, Y]:=RGB(0, 0, 0);
            end;
          end;
        end;
      end;
  end;
  procedure Dilate(Bitmap:TBitmap;NoNeighbourhood:Integer);

```

```

var
X,Y: Integer;
Grayshade: Byte;
x1,y1,x2,y2,i,j:Integer;
BM:TBitmap;
begin
BM:=TBitmap.Create;
try
BM.Assign(Bitmap);
for Y:=0 to Bitmap.Height -1 do
for X:= 0 to Bitmap.Height -1 do
begin
Grayshade := ColorToRGB(Bitmap.Canvas.Pixels[X, Y]);
if Grayshade = 255 then
begin
Y1:=Y-NoNeighbourhood;
Y2:=Y+NoNeighbourhood;
if Y1<0 then Y1:=0;
if Y2>=Bitmap.Height then Y2:=Bitmap.Height;
for i:=Y1 to Y2 do BM.Canvas.Pixels[i, Y] :=
RGB(255,255,255);
end;
end;
Bitmap.Assign(BM);
X:=0;Y:=0;
with Bitmap do
while Y<=Height - 1 do
begin
X:=0;
while X<=Width - 1 do
begin
Grayshade := ColorToRGB(Canvas.Pixels[X, Y]);
if Grayshade = 255 then
begin

```

```

X1:=X-NoNeighbourhood;
X2:=X+NoNeighbourhood;
if X1<0 then X1:=0;
if X2>=Width then X2:=Width;
for i:=X1 to X2 do BM.Canvas.Pixels[i, Y] :=RGB(255,255,255);
X:=X+NoNeighbourhood div 2 + 1;
end;
end;
Y:=Y+1;
end;
Bitmap.Assign(BM);
finally
BM.Free;
end;
end;
procedure AddNoise(Bitmap:TBitmap;NoNoisyPixel:Integer);
var
X,Y,i:Integer;
begin
with Bitmap do
for i:=1 to NoNoisyPixel do
begin
X:=random(Width);
Y:=random(Height);
Canvas.Pixels[X, Y]:=RGB(255, 255, 255);
end;
end;
function InvertBitmap(MyBitmap: TBitmap): TBitmap;
var
x, y: Integer;
ByteArray: PByteArray;
begin
MyBitmap.PixelFormat := pf24Bit;
for y := 0 to MyBitmap.Height - 1 do

```

```

begin
ByteArray := MyBitmap.ScanLine[y];
for x := 0 to MyBitmap.Width * 3 - 1 do
begin
ByteArray[x] := 255 - ByteArray[x];
end;
end;
Result := MyBitmap;
end;
procedure MedianFilter(Bitmap:TBitmap);
var
x, y, i, j, k, buffer, buffer2, F:integer;
PixLtns:array[0..7] of Integer;
PixIndx:array[0..7] of Integer;
RowL, RowM, RowH: PRGBArray;
Pict: TBitmap;
r1,r2,g1,g2,b1,b2:byte;
begin
for y:=1 to Bitmap.Height - 2 do
begin
RowL:=Bitmap.ScanLine[y-1];
RowM:=Bitmap.ScanLine[y];
RowH:=Bitmap.ScanLine[y+1];
for x:=1 to Bitmap.Width - 2 do
begin
// -
for k:=0 to 7 do PixIndx[k]:=k;
// calc lightness of surrounding 8 pixels
PixLtns[0]:=(MinIntValue([RowL[x-1].rgbtRed,RowL[x-1].rgbtGreen,RowL[x-1].rgbtBlue])+MaxIntValue([RowL[x-1].rgbtRed,RowL[x-1].rgbtGreen,RowL[x-1].rgbtBlue]))div 2;
PixLtns[1]:=(MinIntValue([RowL[x].rgbtRed,RowL[x].rgbtGreen,RowL[x].rgbtBlue])+MaxIntValue([RowL[x].rgbtRed,RowL[x].rgbtGreen,RowL[x].rgbtBlue]))div 2;
PixLtns[2]:=(MinIntValue([RowL[x+1].rgbtRed,RowL[x+1].rgbtGreen,RowL[x+1]

```

```

.rgbtBlue])+MaxIntValue([RowL[x+1].rgbtRed,RowL[x+1].rgbtGreen,RowL[x+1].
rgbtBlue]))div 2;
PixLtns[3]:=(MinIntValue([RowM[x-1].rgbtRed,RowM[x-1].rgbtGreen,RowM[x-
1].rgbtBlue])+MaxIntValue([RowM[x-1].rgbtRed,RowM[x-1].rgbtGreen,
RowM[x-1].rgbtBlue]))div 2;
PixLtns[4]:=(MinIntValue([RowM[x+1].rgbtRed,RowM[x+1].rgbtGreen,RowM[x+1]
.rgbtBlue])+MaxIntValue([RowM[x+1].rgbtRed,RowM[x+1].rgbtGreen,RowM[x+1].
rgbtBlue]))div 2;
PixLtns[5]:=(MinIntValue([RowH[x-1].rgbtRed,RowH[x-
1].rgbtGreen,RowH[x-1].rgbtBlue])+MaxIntValue([RowH[x-1].rgbtRed,RowH[x-
1].rgbtGreen,RowH[x-1].rgbtBlue]))div 2;
PixLtns[6]:=(MinIntValue([RowH[x].rgbtRed,RowH[x].rgbtGreen,RowH[x].rgbtB
lue])+MaxIntValue([RowH[x].rgbtRed,RowH[x].rgbtGreen,RowH[x].rgbtBlue]))
div 2;
PixLtns[7]:=(MinIntValue([RowH[x+1].rgbtRed,RowH[x+1].rgbtGreen,RowH[x+1]
.rgbtBlue])+MaxIntValue([RowH[x+1].rgbtRed,RowH[x+1].rgbtGreen,RowH[x+1].
rgbtBlue]))div 2;
//find median pixels
for j:=0 to 7 do
for k:=1 to 7-j do
begin
if PixLtns[j+k] <= PixLtns[j] then
begin
buffer:=PixLtns[j];
PixLtns[j]:=PixLtns[j+k];
PixLtns[j+k]:=buffer;
buffer2:=j;
PixIndx[j]:=PixIndx[j+k];
PixIndx[j+k]:=buffer2;
end;
end;
// get rgb values of median pixels
if PixIndx[3]=0 then
begin

```

```

r1:=RowL[x-1].rgbtRed;
g1:=RowL[x-1].rgbtGreen;
b1:=RowL[x-1].rgbtBlue;
end;
if PixIndx[3]=1 then
begin
r1:=RowL[x].rgbtRed;
g1:=RowL[x].rgbtGreen;
b1:=RowL[x].rgbtBlue;
end;
if PixIndx[3]=2 then
begin
r1:=RowL[x+1].rgbtRed;
g1:=RowL[x+1].rgbtGreen;
b1:=RowL[x+1].rgbtBlue;
end;
if PixIndx[3]=3 then
begin
r1:=RowM[x-1].rgbtRed;
g1:=RowM[x-1].rgbtGreen;
b1:=RowM[x-1].rgbtBlue;
end;
if PixIndx[3]=4 then
begin
r1:=RowM[x+1].rgbtRed;
g1:=RowM[x+1].rgbtGreen;
b1:=RowM[x+1].rgbtBlue;
end;
if PixIndx[3]=5 then
begin
r1:=RowH[x-1].rgbtRed;
g1:=RowH[x-1].rgbtGreen;
b1:=RowH[x-1].rgbtBlue;
end;

```

```

if PixIndx[3]=6 then
begin
r1:=RowH[x].rgbtRed;
g1:=RowH[x].rgbtGreen;
b1:=RowH[x].rgbtBlue;
end;
if PixIndx[3]=7 then
begin
r1:=RowH[x+1].rgbtRed;
g1:=RowH[x+1].rgbtGreen;
b1:=RowH[x+1].rgbtBlue;
end;
if PixIndx[4]=0 then
begin
r2:=RowL[x-1].rgbtRed;
g2:=RowL[x-1].rgbtGreen;
b2:=RowL[x-1].rgbtBlue;
end;
if PixIndx[4]=1 then
begin
r2:=RowL[x].rgbtRed;
g2:=RowL[x].rgbtGreen;
b2:=RowL[x].rgbtBlue;
end;
if PixIndx[4]=2 then
begin
r2:=RowL[x+1].rgbtRed;
g2:=RowL[x+1].rgbtGreen;
b2:=RowL[x+1].rgbtBlue;
end;
if PixIndx[4]=3 then
begin
r2:=RowM[x-1].rgbtRed;
g2:=RowM[x-1].rgbtGreen;

```



```

b2:=RowM[x-1].rgbtBlue;
end;
if PixIndx[4]=4 then
begin
r2:=RowM[x+1].rgbtRed;
g2:=RowM[x+1].rgbtGreen;
b2:=RowM[x+1].rgbtBlue;
end;
if PixIndx[4]=5 then
begin
r2:=RowH[x-1].rgbtRed;
g2:=RowH[x-1].rgbtGreen;
b2:=RowH[x-1].rgbtBlue;
end;
if PixIndx[4]=6 then
begin
r2:=RowH[x].rgbtRed;
g2:=RowH[x].rgbtGreen;
b2:=RowH[x].rgbtBlue;
end;
if PixIndx[4]=7 then
begin
r2:=RowH[x+1].rgbtRed;
g2:=RowH[x+1].rgbtGreen;
b2:=RowH[x+1].rgbtBlue;
end;
// gives a good but grayscale image
RowM[x].rgbtRed:=(PixLtns[3]+PixLtns[4]) div 2;
RowM[x].rgbtGreen:=(PixLtns[3]+PixLtns[4]) div 2;
RowM[x].rgbtBlue:=(PixLtns[3]+PixLtns[4]) div 2;
end;
end;
end;
procedure DrawZone(Bitmap:TBitmap;X1,Y1,X2,Y2:Integer;C:TColor);

```

```

begin
Bitmap.Canvas.Pen.Color:=C;
Bitmap.Canvas.Pen.Width := 5;
Bitmap.Canvas.brush.style := bsclear;
Bitmap.Canvas.Rectangle(X1,Y1,X2,Y2);
end;

procedure TForm1.ButtonConnectClick(Sender: TObject);
begin
if Not tscap321.Connected then
ButtonConnect.Caption := 'Connecting...';
try
tscap321.Connected := Not tscap321.Connected;
finally
begin
if tscap321.Connected then
ButtonConnect.Caption := 'Disconnect from Cam Device'
else
ButtonConnect.Caption := 'Connect to Cam Device';
end;
end;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
Timer1.Enabled:=False;
try
CaptureFrame(ExtractFilePath(Application.ExeName)+'\test.bmp');
finally
Timer1.Enabled :=True;
end;
end;

procedure TForm1.LoadSurveillanceZone(FileName:String);
begin
if FileExists(FileName) then
begin

```

```

image0.Picture.LoadFromFile(FileName);
Image01.Picture.Bitmap.Assign(Image0.Picture.Bitmap);
Image02.Picture.Bitmap.Assign(Image0.Picture.Bitmap);
ConvertBitmapToGrayscale(Image02.Picture.Bitmap);
Image03.Picture.Bitmap.Assign(Image02.Picture.Bitmap);
MedianFilter(Image03.Picture.Bitmap);
DrawZone(Image01.Picture.Bitmap,ZoneX1,ZoneY1,ZoneX2,ZoneY2,clYellow);
end;
end;
function
CheckAlarm(Bitmap:TBitmap;ZoneX1,ZoneY1,ZoneX2,ZoneY2:Integer;AlarmSensit
ivity:Integer):Boolean;
var
Pixel:Byte;
White,i,j:Integer;
begin
White:=0;
with Bitmap do
for i := ZoneX1-1 to ZoneX2-1 do
for j := ZoneY1-1 to ZoneY2-1 do
begin
Pixel:=ColorToRGB(Canvas.Pixels[i, j]);
if Pixel >0 then
White:=White+1;
end;
Result:=white/(ZoneX2-ZoneX1+1)*(ZoneY2-ZoneY1+1)*100>=(100-AlarmSensitivity);
end;
procedure
ProcessFrame(Frame,GrayScale,Median,BackGround,Thresold,Dilate:TBitmap;NoiseLevel:I
nteger);
var
i, j, X,Y: Integer;
Grayshade, Red, Green, Blue: Byte;
PixelColor: Longint;

```

```

begin
with GrayScale do
begin
for i := 0 to Width - 1 do
for j := 0 to Height - 1 do
begin
PixelColor := ColorToRGB(Canvas.Pixels[i, j]);
Red := PixelColor;
Green := PixelColor shr 8;
Blue := PixelColor shr 16;
Grayshade := Round(0.3 * Red + 0.6 * Green + 0.1 * Blue);
Canvas.Pixels[i, j] := RGB(Grayshade, Grayshade, Grayshade);
end;
for i:=1 to NoiseLevel do
begin
X:=random(Width);
Y:=random(Height);
Canvas.Pixels[X, Y]:=RGB(255, 255, 255);
end;
Median.Assign(GrayScale);
if NoiseLevel > 0 then
MedianFilter(Median);
end;
end;
procedure TForm1.CaptureFrame(FileName:String);
var
NoiseLevel:Integer;
begin
if tscap321.Connected then
begin
ButtonConnect.Caption:='Processing Frame...';
tsCap321.CapOrder:=Start;
tsCap321.CapOrder:=Grab;
tscap321.SaveAsBMP := FileName;

```

```

{ image11.Picture.LoadFromFile(FileName);
Application.ProcessMessages;
Image12.Picture.Bitmap.Assign(Image11.Picture.Bitmap);
Application.ProcessMessages;
Image21.Picture.Bitmap.Assign(Image03.Picture.Bitmap);
Application.ProcessMessages;
//if CheckBoxAddNoise1.Checked then NoiseLevel:=1000 else
NoiseLevel:=0;
//ProcessFrame(Image11.Picture.Bitmap,
// Image12.Picture.Bitmap,
// Image13.Picture.Bitmap,
// Image21.Picture.Bitmap,
//
Image22.Picture.Bitmap,Image23.Picture.Bitmap,NoiseLevel);
}
image11.Picture.LoadFromFile(FileName);
Application.ProcessMessages;
Image12.Picture.Bitmap.Assign(Image11.Picture.Bitmap);
ConvertBitmapToGrayscale(Image12.Picture.Bitmap);
Application.ProcessMessages;
if CheckBoxAddNoise1.Checked then
AddNoise(image12.Picture.Bitmap,1000);
Application.ProcessMessages;
Image13.Picture.Bitmap.Assign(Image12.Picture.Bitmap);
MedianFilter(Image13.Picture.Bitmap);
Application.ProcessMessages;
BackgroundSubtraction(Image03.Picture.Bitmap,Image13.Picture.Bitmap,Image.Picture.Bitm
ap);
Application.ProcessMessages;
//Image21.Picture.Bitmap:=InvertBitmap(Image21.Picture.Bitmap);
Image22.Picture.Bitmap.Assign(Image21.Picture.Bitmap);
Thresholding(Image22.Picture.Bitmap,StrToInt(EditThreshold.Text));
Application.ProcessMessages;
Image23.Picture.Bitmap.Assign(Image22.Picture.Bitmap);

```

```

//Dilate(Image23.Picture.Bitmap,10);
if
CheckAlarm(Image23.Picture.Bitmap,ZoneX1,ZoneY1,ZoneX2,ZoneY2,StrToInt(Ed
itAlarmSensitivity.Text)) then
DrawZone(Image23.Picture.Bitmap,ZoneX1,ZoneY1,ZoneX2,ZoneY2,ClRed)
else
DrawZone(Image23.Picture.Bitmap,ZoneX1,ZoneY1,ZoneX2,ZoneY2,ClGreen);
tsCap321.CapOrder:=Stop;
ButtonConnect.Caption := 'Disconnect from Cam Device';
Application.ProcessMessages;
end;
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
LoadSurveillanceZone(ExtractFilePath(Application.ExeName)+'\surveillancez
one.bmp');
StringGrid1.Cells[0,0]:='Alarm Zone';
StringGrid1.Cells[1,0]:='X';
StringGrid1.Cells[2,0]:='Y';
StringGrid1.Cells[0,1]:='(X1,Y1)';
StringGrid1.Cells[0,2]:='(X2,Y2)';
StringGrid1.Cells[1,1]:='1';
StringGrid1.Cells[2,1]:='1';
StringGrid1.Cells[1,2]:=IntToStr(Image0.Picture.Bitmap.Width);
StringGrid1.Cells[2,2]:=IntToStr(Image0.Picture.Bitmap.Height);
ZoneX1:=1;
ZoneY1:=1;
ZoneX2:=Image0.Picture.Bitmap.Width;
ZoneY2:=Image0.Picture.Bitmap.Height;
DrawZone(Image01.Picture.Bitmap,1,1,Image0.Picture.Bitmap.Width,Image0.Pi
cture.Bitmap.Height,clYellow);
end;
procedure TForm1.btn2Click(Sender: TObject);
begin

```

```

Timer1.Enabled:=False;
try
if tsCap321.Connected then
begin
tscap321.SaveAsBMP :=
ExtractFilePath(Application.ExeName)+'\surveillancezone.bmp';
LoadSurveillanceZone(ExtractFilePath(Application.ExeName)+'\surveillancezone.bmp');
tsCap321.CapOrder:=Stop;
end
else
ShowMessage('Please Press "Connect to Camera" button to make theconnection active for
camera. ');
finally
begin
timer1.Interval := StrToInt(EditRefreshRate.Text)*1000;
Timer1.Enabled:=True;
end;
end;
end;
procedure TForm1.ButtonApplyZoneClick(Sender: TObject);
var
X1,Y1,X2,Y2:Integer;
begin
StringGrid1.Cells[0,0]:='Alarm Zone';
StringGrid1.Cells[1,0]:='X';
StringGrid1.Cells[2,0]:='Y';
StringGrid1.Cells[0,1]:='(X1,Y1)';
StringGrid1.Cells[0,2]:='(X2,Y2)';
try
X1:=StrToInt(StringGrid1.Cells[1,1]);
Y1:=StrToInt(StringGrid1.Cells[2,1]);
X2:=StrToInt(StringGrid1.Cells[1,2]);
Y2:=StrToInt(StringGrid1.Cells[2,2]);
if ( X1 < 1 ) or ( X1 >= Image0.Picture.Bitmap.Width ) then X1:=1;

```

```

if ( X2 < X1 ) or ( X2 > Image0.Picture.Bitmap.Width ) then
X2:=Image0.Picture.Bitmap.Width;
if ( Y1 < 1 ) or ( Y1 >= Image0.Picture.Bitmap.Height ) then Y1:=1;
if ( Y2 < Y1 ) or ( Y2 > Image0.Picture.Bitmap.Height ) then
Y2:=Image0.Picture.Bitmap.Height;
StringGrid1.Cells[1,1]:=IntToStr(X1);
StringGrid1.Cells[2,1]:=IntToStr(Y1);
StringGrid1.Cells[1,2]:=IntToStr(X2);
StringGrid1.Cells[2,2]:=IntToStr(Y2);
ZoneX1:=X1;
ZoneY1:=Y1;
ZoneX2:=X2;
ZoneY2:=Y2;
LoadSurveillanceZone(ExtractFilePath(Application.ExeName)+"\surveillancezone.bmp");
except
end;
end;
end.

```



## APPENDIX C

### UGSM.pas, GSM Interface Module

```
unit DMGSM;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, Connect,
  CommConnect, GSM, OleServer, Terminal, XSMS;
const
  cm_NetRegistrationChanged = wm_User+100;
  cm_SMSReceived = wm_User+101;
  cm_ModemStatusChanged = wm_User+102;
  NO_LOG_FILE = 'NUL';
type
  TGSMDataModule = class(TDataModule)
  GSM: TGSM;
  Comm1: TComm;
  procedure DataModule1Create(Sender: TObject);
  procedure GSMDataModuleDestroy(Sender: TObject);
  procedure GSMAfterOpen(DataSet: TConnection);
  procedure GSMNetworkRegistration(Sender: TObject; aNewStatus: Integer);
  procedure GSMBeforeOpen(DataSet: TConnection);
  procedure GSMAfterClose(DataSet: TConnection);
  procedure GSMBusyChanged(Sender: TObject; aBusy: Boolean);
  procedure GSMUnsolicitedSMS(Sender: TObject; Idx: Integer; aSMS: TSMS);
  procedure GSMRxChar(Sender: TObject; aChar: Char);
  procedure Comm1RxChar(Sender: TObject; Count: Integer);
  private
    FGSMStr: string;
    RecSMSBuffer: TList;
    FGSMLogFile: string;
    FSMSLogFile: string;
    FSMSProtocol: TSMSProtocolStack;
  function GetActive: Boolean;
  procedure SetActive(aValue: Boolean);
```

```

procedure LogToFile(aFilename, aText: string);
procedure ReadIniFile;
procedure WriteIniFile;
procedure SetGSMLogFile(const Value: string);
procedure SetSMSLogFile(const Value: string);
function LogSafe(S: string): string;
public
ModemLog, SMSLog: TTerminal;
property GSMLogFile: string read FGSMLogFile write SetGSMLogFile;
property SMSLogFile: string read FSMSLogFile write SetSMSLogFile;
property Active: Boolean read GetActive write SetActive;
procedure SendSMS(aSMS: TSMSSubmit);
procedure ModemStatusChanged(aNewStatus: Integer);
procedure ProcessSMS(aSMS: TSMS);
procedure LogSMS(aSMS: TSMS);
procedure LogGSM(aChar: Char);
end;
var
GSMDataModule: TGSMDDataModule;
implementation
uses
AuxGsm, LogFrm, AuxStr, parallelport, Main;
{$R *.DFM}
Var
ParallelPortDataReg:Byte=0;
procedure TGSMDDataModule.DataModule1Create(Sender: TObject);
begin
RecSMSBuffer:= TList.Create;
FSMSProtocol:= TSMSProtocolStack.Create;
FGSMStr := "";
ReadIniFile;
end;
procedure TGSMDDataModule.ReadIniFile;
var

```

```

BR: TBaudrate;
Timeout: Integer;
PB: ^Byte;
begin
with GetGSMINI do
try
Comm1.DeviceName:= ReadString('Modem', 'DeviceName',Comm1.DeviceName);
GSM.Equipment:=          Str2GMSEquipment(UpperCase(ReadString('Modem','Device',
GSMEquipmentShortName[GSM.Equipment]))));
GSM.PIN:= ReadString('Modem', 'PIN', GSM.PIN);
GSM.SCA:= ReadString('Modem', 'SCA', GSM.SCA);
PB:= @GSM.UnsolicitedIndication;
PB^:= ReadInteger('Modem','UnsolicitedIndication',PB^);
GSM.SMSFormat:= ReadInteger('Modem','SMSFormat',0);
GSM.SetDefaults;
Timeout := ReadInteger('Modem','RecTimeout',0);
if Timeout > 0 then
GSM.RecTimeout := Timeout;
if Int2Baudrate(ReadInteger('Modem', 'BaudRate', 0), BR) then
GSM.COMDevice.Baudrate:= BR;
FGSMLogFile :=
ReadString('Modem.Files', 'GSMLog', GetProgramPath+'gsm.log');
FSMSLogFile :=
ReadString('Modem.Files', 'SMSLog', GetProgramPath+'sms.log');
finally
Free;
end;
end;
procedure TGSMDDataModule.WriteIniFile;
begin
with GetGSMINI do
try
WriteString('Modem', 'DeviceName',Comm1.DeviceName);
WriteInteger('Modem', 'BaudRate',BaudRate2Int(Comm1.BaudRate));

```

```

WriteString('Modem', 'Device', GSMEquipmentShortName[GSM.Equipment]);
WriteString('Modem', 'PIN', GSM.PIN);
WriteString('Modem', 'SCA', GSM.SCA);
WriteInteger('Modem','SMSFormat',GSM.SMSFormat);
WriteInteger('Modem','UnsolicitedIndication',Byte(GSM.UnsolicitedIndication));
WriteString('Modem.Files', 'GSMLog', FGSMLogFile);
WriteString('Modem.Files', 'SMSLog', FSMSLogFile);
finally
Free;
end;
end;

function TGSMDDataModule.GetActive;
begin
Result:= GSM.Active;
end;

procedure TGSMDDataModule.SetActive;
begin
GSM.Active:= aValue;
end;

procedure TGSMDDataModule.GSMDDataModuleDestroy(Sender: TObject);
begin
WriteIniFile;
RecSMSBuffer.Free;
FSMSProtocol.Free;
end;

procedure TGSMDDataModule.ProcessSMS;
var
S2: string;
I: Integer;
NBS: TSMSProtocol;
Addr: Word;
SM: TSmartMessage;
EMS: TEMS;
resourcestring

```

```

sFragment = '*ref:%.2x, #d/%d/%d, , addr:%.4x';
sComplete = 'complete';
begin
LogSMS(aSMS);
if aSMS is TSMSDeliver then
begin
S2:= "";
fSMSProtocol.CleanSMSProtocols(5{min}/60/24, False);
if fSMSProtocol.ProcessSMS(aSMS, NBS) then
begin
Addr:= 0;
if NBS is TSMSProtocol2 then
Addr:= TSMSProtocol2(NBS).DestinationAddress;
S2:= Format(sFragment, [NBS.Reference, NBS.LastInserted,NBS.InsertedCount,
NBS.FragmentCount, Addr]);
if NBS.Status = smspsOK then
begin
S2:= S2+' '+sComplete;
if NBS is TSMSProtocol2 then
begin
SM:= TSmartMessage.CreateSM(NBS.Data,
TSMSProtocol2(NBS).DestinationAddress);
if SM <> nil then
try
S2:= S2 + ' SmartMessage:'+SM.ClassName;
{if SM is TSMMultipartMessage then
S2:= S2+'Multipart message'
else if SM is TSMRingingTone then
S2:= S2+'Ringing tone' classname inheritsfrom }
finally
SM.Free;
end
else if NBS is TUDHProtocol then
begin

```

```

EMS:= TEMS.Create;
try
try
EMS.ReadFrom(TUDHProtocol(NBS));
S2:= S2+ ' EMS';
for I:= 0 to EMS.Objects.Count-1 do
begin
if I > 0 then
S2:= S2+',';
S2:= S2+EMS.Objects[I].ClassName;
end;
S2:= S2+' '+EMS.Text+"";
except
end;
finally
EMS.Free;
end;
end;
end
else if NBS is TSiemensOTA then
begin
S2:= S2 + Format(' SEO: %s, Name: "%s"',
[string(TSiemensOTA(NBS).ObjectType),
string(TSiemensOTA(NBS).ObjectName)]);
end;
fSMSProtocol.RemoveSMSProtocol(NBS);
NBS.Free; // delete processed
data
end;
end;
if S2 <> " then
begin
if SMSLog <> nil then
SMSLog.Log(S2);

```

```

end;
end;
NotifyForms(cm_SMSReceived, 0, Longint(aSMS));
end;
procedure TGSMDDataModule.GSMAfterOpen(DataSet: TConnection);
var
Sg: TStrings;
I, J: Integer;
begin
if GSM.Equipment in [eqM20, eqNokia9110, eqWavecom, eqFasttrack] then
Sg:= GSM.GetSMSList(4{all})
else
Sg:= GSM.GetSMSList(-1);
try
for I:= 0 to Sg.Count-1 do
begin
J:= StrToInt(Sg.Names[I]);
if StrToInt(Sg.Values[Sg.Names[I]]) in [0{unread}] then
begin
if Sg.Objects[I] <> nil then
ProcessSMS(Sg.Objects[I] as TSMS);
end;
GSM.DeleteSMS(J);
end;
finally
Sg.Free;
end;
ModemStatusChanged(1 or $80);
end;
procedure TGSMDDataModule.GSMNetworkRegistration(Sender: TObject;aNewStatus:
Integer);
begin
NotifyForms(cm_NetRegistrationChanged, aNewStatus, 0);
end;

```

```

procedure TGSMDDataModule.GSMBeforeOpen(DataSet: TConnection);
begin
  WriteIniFile;
  ModemStatusChanged(0);
end;

procedure TGSMDDataModule.ModemStatusChanged;
begin
  NotifyForms(cm_ModemStatusChanged, aNewStatus, 0);
end;

procedure TGSMDDataModule.GSMAfterClose(DataSet: TConnection);
begin
  ModemStatusChanged(-1);
end;

procedure TGSMDDataModule.GSMBusyChanged(Sender: TObject; aBusy: Boolean);
begin
  ModemStatusChanged(Integer(not aBusy));
end;

procedure TGSMDDataModule.SendSMS;
begin
  GSM.SendSMS(aSMS);
end;

procedure TGSMDDataModule.GSMUnsolicitedSMS(Sender: TObject; Idx: Integer;
aSMS: TSMS);
var
  Stat: Integer;
begin
  if aSMS = nil then { PDU mode }
  begin
    try
      aSMS:= GSM.ReadSMS(Idx, Stat);
      if Stat in [0{unread}] then
        ProcessSMS(aSMS);
      finally
        GSM.DeleteSMS(Idx);

```



```

end;
end
else
begin { text mode }
ProcessSMS(aSMS);
if Idx <> -1 then
GSM.DeleteSMS(Idx);
end;
end;
procedure TGSMDDataModule.LogToFile(aFilename, aText: string);
var
St: TStream;
M: Word;
begin
if aFilename <> NO_LOG_FILE then
begin
if FileExists(aFilename) then
M:= fmOpenWrite+fmShareDenyWrite
else M:= fmCreate;
St:= TFileStream.Create(aFilename, M);
try
St.Position:= St.Size;
St.WriteBuffer(aText[1], Length(aText));
finally
St.Free;
end;
end;
end;
procedure TGSMDDataModule.LogGSM(aChar: Char);
begin
FGSMStr := FGSMStr + aChar;
if aChar = LF then
begin
LogToFile(FGSMLogFile,

```

```

Format('%s: %s',[DateTimeToStr(Now),FGSMStr]));
FGSMStr := "";
end;
end;
procedure TGSMDDataModule.LogSMS;
var
S2, S3: string;
// I: Integer;
phandle:THandle ;
Reg:Byte;
begin
S3:= "";
{ if aSMS is TSMS2 then
if TSMS2(aSMS).DCS and dcsmAplhabet <> dcsAlphabetDefault) then
// 8bit taky hexa
begin
S3:= "";
for I:= 1 to Length(TSMSSubmit(aSMS).UD) do
S3:= S3+Format('% .2x', [Byte(TSMSSubmit(aSMS).UD[I]))]);
S3:= S3+' / ';
end; }
if aSMS is TMSMSTDeliver then
begin
with aSMS as TMSMSTDeliver do
begin
S2:= Format('>%s, OA: %s, PID: %.2x, UD: %s%s',
[DateTimeToStr(SCTS), OA, PID, S3, LogSafe{SMStoISO}(UD)]);
S3:= UpperCase(Format('%s', [ LogSafe{SMStoISO}(UD)]));
if S3='LIGHTS ON' then
begin
phandle:=OpenParallelPort(1);
ParallelPortDataReg:=ParallelPortDataReg or $01;
WriteParallelPort(phandle, TPRData, ParallelPortDataReg);
CloseParallelPort(phandle);

```

```

GSMCentralForm.ShapeD0.Brush.Color:=clgreen;
end
else if S3='LIGHTS OFF' then
begin
phandle:=OpenParallelPort(1);
ParallelPortDataReg:=ParallelPortDataReg and $FE;
WriteParallelPort(phandle, TPRData, ParallelPortDataReg);
CloseParallelPort(phandle);
GSMCentralForm.ShapeD0.Brush.Color:=clwhite;
end
else if S3='FAN ON' then
begin
phandle:=OpenParallelPort(1);
ParallelPortDataReg:=ParallelPortDataReg or $02;
WriteParallelPort(phandle, TPRData, ParallelPortDataReg);
CloseParallelPort(phandle);
GSMCentralForm.ShapeD1.Brush.Color:=clgreen;
end
else if S3='FAN OFF' then
begin
phandle:=OpenParallelPort(1);
ParallelPortDataReg:=ParallelPortDataReg and $FD;
WriteParallelPort(phandle, TPRData, ParallelPortDataReg);
CloseParallelPort(phandle);
GSMCentralForm.ShapeD1.Brush.Color:=clwhite;
end
end
end
else if aSMS is TSMSSubmit then
begin
with aSMS as TSMSSubmit do
S2:= Format('<%s, DA: %s, PID: %.2x, UD: %s%s',
[DateTimeToStr(Now), DA, PID, S3, LogSafe{SMStoISO}(UD)]);
end

```

```

else if aSMS is TSMSStatusReport then
begin
with aSMS as TSMSStatusReport do
S2:= Format('!%s, OA: %s, Status: %d, DT: %s',
[DateTimeToStr(Now), OA, Status, DateTimeToStr(DT)]);
end
else
Exit;
if SMSLog <> nil then
SMSLog.Log(S2);
S2 := S2+#13#10;
LogToFile(FSMSLogFile,S2);
end;
procedure TGSMDDataModule.GSMRxChar(Sender: TObject; aChar: Char);
begin
if ModemLog <> nil then
ModemLog.LogTerm(aChar);
LogGSM(aChar);
end;
procedure TGSMDDataModule.Comm1RxChar(Sender: TObject; Count: Integer);
var
I: Integer;
S: string;
begin
S:= (Sender as TComm).Retrieve(Count);
for I:= 1 to Length(S) do
GSMRxChar(nil, S[I]);
end;
procedure TGSMDDataModule.SetGSMLogFile(const Value: string);
begin
FGSMLogFile := Value;
end;
procedure TGSMDDataModule.SetSMSLogFile(const Value: string);
begin

```

```

FSMSLogFile := Value;
end;
function TGSMDDataModule.LogSafe(S: string): string;
var
I: Integer;
begin
Result:= "";
for I:= 1 to Length(S) do
if S[I] = #13 then
Result:= Result+'<CR>'
else if S[I] = #10 then
Result:= Result+'<LF>'
else if S[I] in [#0..#31] then
begin
Result:= Bin2Hex(S);
Break;
end
else
Result:= Result+S[I];
end;
end.

```