UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **REAL TIME MONITORING OF SURVEILLANCE CCTV**

SESI PENGAJIAN:________________2010/2011

Saya **NADHIRAH BT ALI (880603-04-5274)**
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

□ **SULIT** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

□ **TERHAD** (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

☑ **TIDAK TERHAD**

Disahkan oleh:

____________________________	____________________________
(TANDATANGAN PENULIS)	(TANDATANGAN PENYELIA)

Alamat Tetap:

J 1095. BATU 21 ,
KAMPUNG TERENTANG,
77000 JASIN,
MELAKA

DR. KAMARUL HAWARI BIN GHAZALI

(Nama Penyelia)

Tarikh: **29 NOVEMBER 2010**

Tarikh: **29 NOVEMBER 2010**

CATATAN: * Potong yang tidak berkenaan.

** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.

♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

REAL TIME MONITORING OF SURVEILLANCE CCTV

NADHIRAH BT ALI

Thesis submitted in fulfilment of the requirements
for the award of the degree of
Bachelor of Electric Electronics Engineering (Electronics)

Faculty of Electrical and Electronic Engineering
UNIVERSITI MALAYSIA PAHANG

NOVEMBER 2010

SUPERVISOR’S DECLARATION

I hereby declare that I have checked this project and in my opinion, this project is adequate in terms of scope and quality for the award of the degree of Bachelor of Electrical & Electronics Engineering (Electronics).

Signature
Name of Supervisor: DR. KAMARUL HAWARI BIN GHAZALI
Date: 29 NOVEMBER 2010

STUDENT'S DECLARATION

I hereby declare that the work in this project is my own except for quotations and summaries which have been duly acknowledged. The project has not been accepted for any degree and is not concurently submitted for award of other degree.

Signature
Name: NADHIRAH BT ALI
Date: 29 NOVEMBER 2010

Dedicated to my parents

ACKNOWLEDGEMENTS

I take this opportunity to thanks my advisor, Dr Kamarul Hawari Bin Ghazali for his guidance throughout my thesis. It was his word of encouragement that help me to perceive my research goal. I thank the entire course mate for their time and patients in reviewing my thesis.

For the unending love and understanding from my parent who motivated me to accomplish this through, and I am very grateful to my sibling who give me the courage and strength.

I would like to take this opportunity to thanks to my friends, who give me support and help.

ABSTRACT

Traditionally the surveillance monitoring system is done in the larger room and by a mount of manpower. But nowadays, monitoring surveillance system can be done through online network. This type of monitoring is more time consuming and can reduce the manpower. Moreover, it gives the user flexibility to monitor their properties where ever they want as long as they have the internet network. Either than that, this project also manage to detect the movement in the video, this will be a greater help to the user to observe their properties. In the end, this project is not only can reduce the cost for monitoring but also give advantages to the user.

ABSTRAK

Secara tradisionalnya sistem pemantauan kamera litar tertutup dilakukan di sebuah bilik kawalan yang besar bersama dengan jumlah tenaga kerja yang ramai. Tetapi pada zaman yang lebih maju ini, sistem ini dapat dilakukan melalui rangkaian internet. Dengan cara ini ia dapat mengurangkan masa dan tenaga kerja bagi satu – satu pemantauan. Selain daripada itu, sistem ini memberikan fleksibiliti kepada pengguna untuk memantau harta mereka di mana mereka berada dengan syarat mereka mempunyai rangkaian internet. Selain dari itu, projek ini juga boleh mengesan gerakan dalam video, dan ini akan menjadi bantuan kepada pengguna untuk memantau harta mereka denan lebih mudah. Keimpulannya, projek ini bukan sahaja dapat mengurangkan kos untuk pemantauan tetapi juga memberikan kelebihan kepada pengguna.

TABLE OF CONTENTS

CHAPTER 3 METHODOLOGY

CHAPTER 4 RESULTS AND DISCUSSION

CHAPTER 5 CONCLUSION AND RECOMMENDATIONS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 PROJECT BACKGROUND.

CCTV is the acronym of Closed-circuit television (CCTV). Surveillance CCTV is one of the most important evidence when deals with wrong doings. A video surveillance system covering a large office building or a busy airport can apply hundreds and even thousands of cameras. To avoid communication bottlenecks, the acquired video is often compressed by a local processor within the camera, or at a nearby video server. The compressed video is then transmitted to a central facility for storage and display.

Based on the current technologies, with the set of personal computer (PC) and the internet connection either wire or wireless, the monitoring can be done. With this, the user may monitor the video wherever they want, and the random video playback functions can be provided. With these flexibilities, it gives more advantage to the user to monitor and ensure the safety place they want. It is also may increase the safety of the user properties; this is because there is image processing technique apply in the system.

Image processing apply in this system is to detect object movement. By using this technique it will tell the user there is movement in that particular frame. This technique will highlight movement, thus will alert the user. To apply this technique some analysis must be done. In this project, the analysis is done by using MATLAB software. MATLAB is software that widely used in the computer analysis circle. In the MATLAB there is image processing toolbox. This toolbox will guide in the image processing technique and the details about the concept apply.

The Graphical User Interface (GUI) is built in the MATLAB. This GUI is build for user friendly purpose. The built of the GUI will guide the user to monitor their CCTV. It will make programs easier to use by providing them with a consistent appearance and with intuitive controls like pushbuttons, list boxes, sliders, menus, and so forth. All of these elements are complete in MATLAB GUI builder.

1.2 PROBLEM STATEMENT

Even though, the monitoring can be done remotely, with some help in image processing will increase the efficiency of the system. In this project, image processing is applied to let the user know there is object movement. This will reduce time consuming of the user. This will make the monitoring is a lot mo easier.

This project is a web base project, but the analysis is done in the MATLAB format, thus there is conflict in the language used. There for, studies in web language are necessary in order to called back the MATLAB function in web language.

1.3 PROJECT OBJECTIVE

The objective needs to archive to complete this project are:-

1. Obtain the IP address of the IP base CCTV.
2. Build the GUI for user usage for real – time monitoring.

1.4 PROJECT SCOPE

These scopes are determined in order to complete this project:-

1. Apply the image processing technique by using MATLAB 2010a.
2. Design the GUI for real time monitoring
3. Studies of web language for web site design.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

The purpose of this chapter is to discuss of literature review about image processing and remote monitoring using web base.

2.2 MOTION DETECTION

Motion detection is the most commonly technique use nowadays. This is because their usage in many areas like video surveillance systems, traffic monitoring, gesture recognition, advanced user interfaces, sport games players tracking etc. All these application is to identify the object of interest. There are a number of proposed methods in the literature for moving object and all of them can be divided into three classes: temporal difference, optical flow and background subtraction [1].

The background subtraction methods are the most commonly used with the static cameras because of its high performance and low memory requirements [1]. Background subtraction is a method typically used to segment moving regions in video sequences by comparing each new frame to a model of the scene background. It has been used successfully for indoor and outdoor applications [2]. This method need the background and the camera remain static even though the background color intensity may change gradually base on the current situation. Thus it is necessary to compare pixel of the current frame background with the corresponding background.

This is the layback of the background subtraction [1]. This is because due to the background scenes often change, such a car running into or out off the background, someone brings things into or out off the background or even the wavering leaves. For effectiveness and accuracy in this method, the initialization and the background update is important [3].

Thus, base on the idea of updating the background, the adaptive background method is proposed. Adaptive background subtraction (ABS) is a fundamental step for foreground object detection in many real-time video surveillance systems. In many ABS methods, a pixel-based statistical model is used for the background and each pixel is updated online to adapt to various background changes. But, by using this method it needs a lot of computational and memory consumption required. Thus in the paper [4] propose the usage of K-Means clustering method to adaptive background model which is constructed by the Mixture of Gaussians (MOG) for improving the computational time.

As for the busy situation, background subtraction is irrelevant to applied for motion detection, thus optical flow method is introduce. Optical flow is an approximation of the local image motion and specifies how much each image pixel moves between adjacent images. It can achieve success of motion detection in the presence of camera motion or background changing. According to the smoothness constraint, the corresponding points in the two successive frames should not move more than a few pixels. For an uncertain environment, this means that the camera motion or background changing should be relatively small. The method based on optical flow is complex, but it can detect the motion accurately even without knowing the background [5].

Most optical flow techniques are either gradient based methods [6] or region matching based methods [7]. Gradient based methods have been preferred due to speed and performance considerations. These methods analyses the change in intensity and gradient (using partial spatial and temporal derivatives) to determine the optical flow and it also yields smoother and more natural motion fields [7,8]. The popular optical flow technique; Horn and Schunck are a combination of the gradient constraint with a global smoothness term [7] to be smooth in the edge region, the speed of which always sharp changes, so the edge shape of the object easily distorts to constrain the velocity field [7,9] . As for the region matching it rely on determining the correspondence between the two images, by matching 'blocks' of one image to 'blocks' of the other [8].

By using optical flow, detection method of a moving object by mapping, which converts the motion of a stationary environment object into a linear signal trajectory. The one-dimensional optical flow is calculated by using pixels, which belong to the moving object, to eliminate the apparent motion of the stationary environment object [10]. By using optical flow also can detect a multiple motion in a frame because it is calculated base on the velocity of the particular object [8]. Base on this characteristic, it can be used to extract the candidate region based on matching optical flow values [11].

As for temporal difference, it is computes the difference between two or three consecutive frames. It is good at adapting to the dynamic environments, but generally poor at extracting enough relevant feature pixels, which resulting holes are generated in the moving object [12] and cannot detect the entire shape of a moving object with uniform intensity [13].

2.3 REMOTELY MONITORING

Monitoring remotely is analogy from the conventional monitoring technique. In the control room, there are many monitor display the condition of the respective area, same goes for online monitoring. For online monitoring there is setting that can be change either to watch single video or multiple video. In the digital CCTV monitoring also can have apply processing in there. Thus, we can detect the movement, either it is normal or abnormal. There is study in this area says that, the electronic engineers have an understanding of field theory and of flow dynamics which may provide insight into characteristics of crowd behavior, and can also provide the expertise to suggest solutions to crowd monitoring and control based on technological developments in image processing and image understanding [14].

IP-based camera used in a remote monitoring station for intelligent image processing. In order to provide enhanced security, events need to be integrated with a command and control system capable of effectively responding to hundreds of events per day in a busy, critical infrastructure facility [8]. Other than IP-based camera, pan, tilt, and zoom (PTZ) camera also can be used in remote monitoring. The PTZ itself may refer to the feature for the security camera. It also can be used to describe an entire category of cameras where a combination of sound and/or motion and/or change in heat signature may enable the camera to activate, focus and track suspected changes in the video field.

2.4 REAL TIME DETECTION

The real-time abnormal motion detection scheme uses algorithm the macro block motion vectors that are generated anyway as part of standard video compression methods. Motion features are derived from the motion vectors. Normal activity is characterized by the joint statistical distribution of the motion features, estimated during a training phase at the inspected site. During online operation, improbable-motion feature values indicate abnormal motion. Relying on motion vectors rather than on pixel data reduces the input data rate by about two orders of magnitude, and allows real-time operation on limited computational platforms [20].

In paper [1] purpose algorithm base on this block diagram.

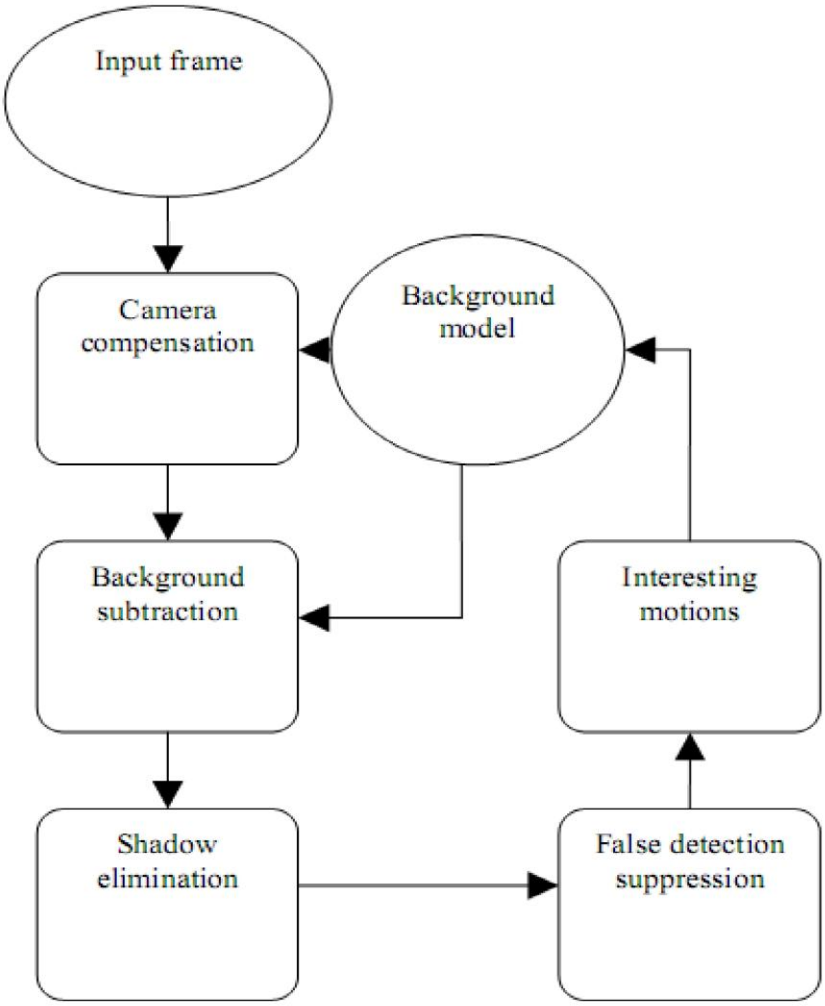


Figure 1: block diagram of algorithm [1]

The camera motion of the input frame is compensated and consecutive temporal difference is performed to extract moving areas from the image. The moving areas include the target areas (moving objects we are interested in) and some uninteresting motion areas (the moving background). Post treatments are used to optimize the detection by filling the small holes in the detected objects and removing uninteresting motions. Since shadows won't have large change between two consecutive frames and little change of shadows to be detected can be removed by the post treatments, shadows have little effect on the accuracy of detection. Thus, they are not handled here to improve the efficiency of this method. To perform this method in real-time and with high accuracy, we design every block carefully. Since the consecutive temporal difference approach requires no background model and little memory, this approach is quite efficient and accurate in that it has a low computational cost and it adapts quickly to the changes of the background.

First, the input frame is compensated by using a camera motion compensation algorithm, which takes some spots in the frame as basic pixels and estimates the motion of the camera with a square neighborhood matching method. The whole input frame is then adjusted to make up the motion of the camera. The square neighborhood matching method will be fully described in the following subsection. It shows how it records the deviation of the image.

Second, an improved consecutive temporal difference approach is used to quickly obtain moving areas of the input frame. This approach makes use of three consecutive frames. The three frames are divided into two groups. The first group includes the two previous frames (the two consecutive frames that go before the input frame), while the second group includes the input frame and the frame before it. By subtracting the two groups separately, we get two results of different areas from the two subtractions. Since the intersection of the two results is just the very part of the moving area in the frame previous to the input frame, we can obtain the moving areas by subtracting the second results by the intersection of the two difference frames

Third, post treatments are used in this phase to fill the small holes in the detected objects and to remove uninteresting motions from the moving areas obtained in the second phase. Since shadows are not handled in this method, post treatments deal mostly with the two problems mentioned above. The math morphology techniques are used.

The Algorithm Based Object Recognition and Tracking (ABORAT) system presented in this paper is a vision-based intelligent surveillance system, capable of analyzing video streams. These streams are continuously monitored in specific situations for several days (even weeks), learning to characterize the actions taking place there. This system also infers whether events present a threat that should be signaled to a human operator. The concept of the ABORAT system is to apply intelligent vision algorithms on images acquired at the system’s edge (the camera), thus reducing the workload of the processor at the monitoring station and the network traffic for transferring high resolution images to the monitoring station [15].

A layered architecture for real-time surveillance systems which for each layer includes objects that model the "real world" at a specific abstraction level, from raw data up to domain concepts. Each layer performs abstractions on perceptions coming from the lower layer and formulates timed hypotheses about domain objects. The failure of a hypothesis causes a perception to flow up-stream. In turn, hypotheses flow down-stream, so that their verification is delegated to the lower layers. The proposed architectural patterns have been reified in a Java framework, which has been used in an experimental multi-camera tracking system [16].

For multiple object detection to extract and do well reveal the foreground object, the process of object detection in the proposed algorithm is based on the background subtraction and the double-difference image. Typically, background subtraction is the first step in automated visual surveillance applications and a method used to segment moving regions in video sequences taken by a camera by comparing each new frame to the scene background model. We first create a clean background model which is similar to that taken by an adaptive background modeling algorithm based on a Gaussian-mixture model . This model can reduce the noise generated from a camera itself or from the lamplight twinkling [17].

2.5 SYSTEM NETWORK

The system architecture of the present networked visual monitoring system is shown in Fig. 8. There are types of two cameras in the monitored site. One is the global camera, which captures the global view of interest. The other are tracking cameras, controlled by VMS server to track the intruders. In order to keep the recorded images safely, the captured images are stored in a VMS server as well as VMS DB server. In the case users request real time image monitoring, the system connects to the VMS Server. On the other hand, the system connects to the VMS DB server if users query stored images [18].

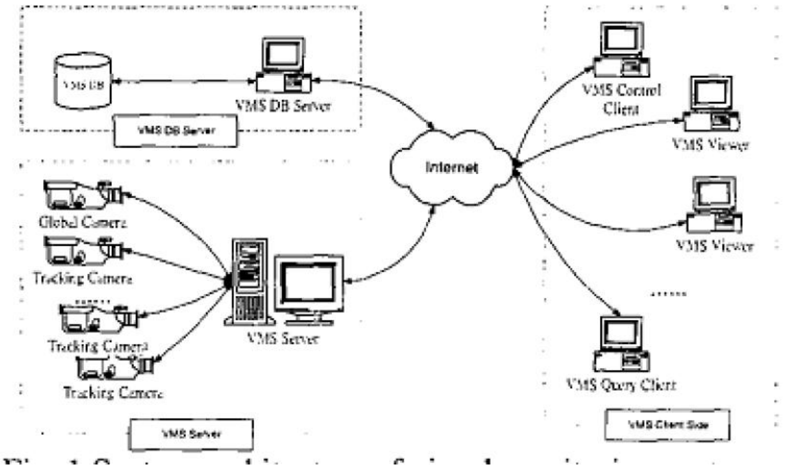


Figure 2: system architecture [18].

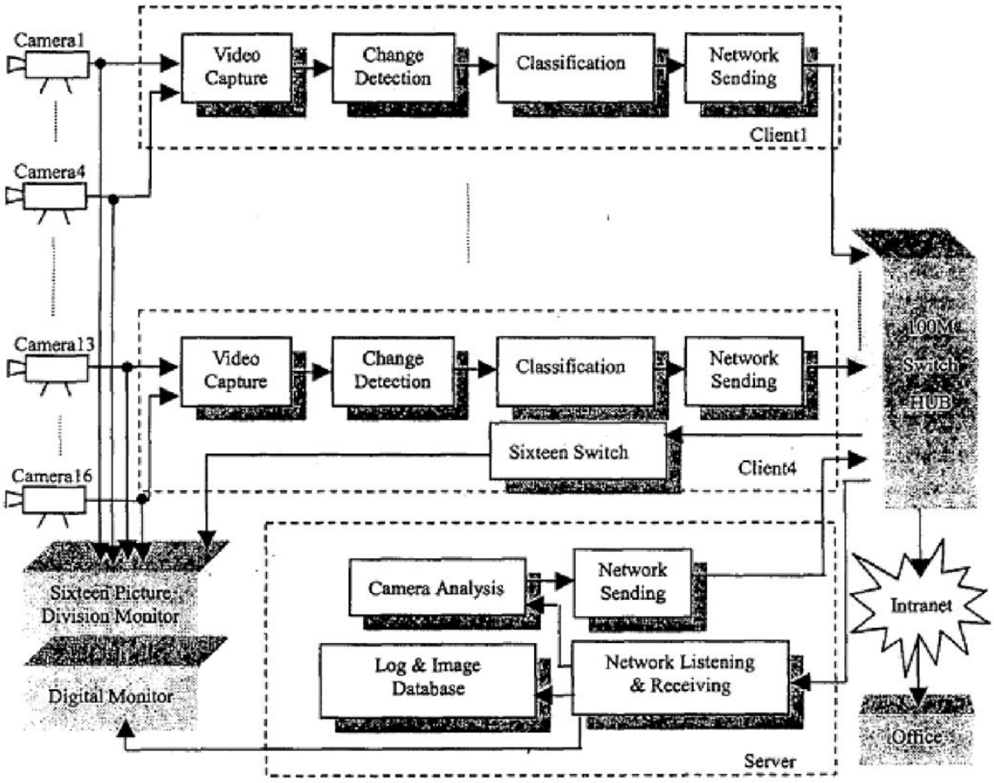


Figure 3: system architecture for [19]

A network-based visual intelligent surveillance system is mainly composed of cameras, client computers, a server computer and a HUB. The clients include video capture module, change detection module, classification module and network sending module. The server includes network listening & receiving module, camera analysis module, a digital monitor and log & image database. The architecture of a surveillance system with sixteen cameras is shown in Fig.9. The video capture Module on each client can capture video sequence from four cameras. So there are four clients in all. All sixteen cameras are connected to the corresponding video capture card on the client computers on the one hand, and connected to an analog sixteen-picture-division monitor on the other hand. On the natural situation, sixteen pictures come from sixteen cameras are simultaneously shown on the analog division monitor [19].

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

In this chapter will describe the details steps taken to detect motion movement analysis. The details also will cover for real time monitoring and how web base is build.

3.2 METHODOLOGY

3.2.1 Setting the IP camera.

The first step is to setup the IP camera at the desire place. After place the IP camera is to obtain the IP address of the IP camera. This is importance since to connect the IP camera with the web is the IP address. IP address is a numerical label that is assigned to devices participating in a computer network that uses the Internet Protocol for communication.

3.2.2 Image processing.

Mean filtering is a simple, intuitive and easy to implement method of smoothing images, such as to reducing the amount of intensity variation between one pixel and the next. It is often used to reduce noise in images. It is simply to replace each pixel value in an image with the mean (`average') value of its neighbors, including itself. This has the effect of eliminating pixel values which are unrepresentative of their surroundings. Mean filtering is usually thought of as a convolution filter. Like other convolutions it is based around a kernel, which represents the shape and size of the neighborhood to be sampled when calculating the mean. Often a 3×3 square kernel is used.

The Gaussian smoothing operator is a 2-D convolution operator that is used to `blur' images and remove detail and noise. In this sense it is similar to the mean filter, but it uses a different kernel that represents the shape of a Gaussian (`bell-shaped') hump. This kernel has some special properties which are A box is scanned over the whole image and the pixel value calculated from the standard deviation of Gaussian is stored in the central element.

The median filter is normally used to reduce noise in an image, somewhat like the mean filter. However, it often does a better job than the mean filter of preserving useful detail in the image. Noise is removed by calculating the median from all its box elements and stores the value to the central element.

Image segmentation refers to the process of dividing image into regions with characteristics, extracting the targets of interest and deleting the useless part. It is one of the most basis and important image processing issue for pattern recognition and low-level computer vision [20].

Histogram based segmentation is one of the easiest way to do image segmentation. Histogram is computed for all the image pixels. The peaks in histogram are produced by the intensity values that are produced after applying the threshold and clustering. The pixel value is used to locate the regions in the image. Based on histogram values and threshold we can classify the low intensity values as object and the high values are background image (most of the cases).

Single gaussian background model is used to separate the background and foreground objects. It is a statically method of separation. In this a set of frames (previous frames) are taken and the calculation is done for separation. The separation is performed by calculating the mean and variance at each pixel position.

Frame difference calculates the difference between 2 frames at every pixel position and store the absolute difference. It is used to visualize the moving objects in a sequence of frames. It takes very less memory for performing the calculation.

Feature Extraction plays a major role to detect the moving objects in sequence of frames. When the input data to an algorithm is too large to be processed and it is suspected to be notoriously redundant (much data, but not much information) then the input data will be transformed into a reduced representation set of features. Every object has a specific feature like color or shape.

The most essential feature of image is the image edge. The edge detection is widely applied in the image recognition, image division, image enhancement and image compress and so on, and it is always the focus in the area of digital image processing. Edges are formed where there is a sharp change in the intensity of images. If there is an object, the pixel positions of the object boundary are stored and in the next sequence of frames this position is verified. Corner based algorithm uses the pixel position of edges for defining and tracking of objects.

The rectangular bounding box with the dimensions of the object produced from residual image. This is because of the segmentation is using the frame difference technique. In this features is extracted by color and here the intensity value describes the color. The pixel values from the first hit of the intensity values from top, bottom, left and right are stored. By using this dimension values a rectangular bounding box is plotted within the limits of the values produced.

Extraction of objects using the features is known as object detection. As stated above, every object have their own special feature extraction. Thus by applying the feature extraction algorithm, the object can be pointed out in each frame.

Optical flow is one way to detect moving objects in a sequence of frames. Typically the motion is represented as vector position of pixels. The pixels is calculated and compared in sequence of frames for the pixel position.

The block matching algorithm is a standard technique for encoding motion in video sequences. It aims at detecting the motion between two images in a block-wise sense. The blocks are usually defined by dividing the image frame into non-overlapping square parts. Each block from the current frame is matched into a block in the destination frame by shifting the current block over a predefined neighborhood of pixels in the destination frame.

Object tracking is a process of a method of following an object through successive image frames to determine how it is moving relative to other objects. This is most commonly done by using the feature extraction of objects and detecting the objects in sequence of frames.

The velocity of moving object is calculated by the distance it travelled with respect to the time. By using the values of distance with respect to frame rate, the velocity of the object is defined.

3.2.3 Graphical User Interface.

The GUI design in MATLAB is complete with the feature need in the design purpose such as push button, radio button, axis and slider. In the MATLAB GUI design, it will save two type of files consist of .m and .fig. the .m file is consist of the function need to run the GUI, while the .fig file is a display of the GUI that have been design.

3.2.4 Website.

Website is the final part of this project. The purpose of the website of this project is to display the video from the IP camera and the remote monitoring. Website may used it own language. But with the MATLAB language still in used since the analysis part is done in MATLAB there have slight difference that usual website design. For MATLAB it complete with the .NET and JAVA platform. Thus it may used either this 2 way to deploy it to the website.

3.3 TECHNIQUE

3.3.1 IP address

The IP camera must be setup first at the desire place. For this project, the IP camera is located in the lab. Next is to obtain the IP address of the IP camera. The IP address of the IP camera is 172.20.180.44. After manage to obtain the IP address is to proceed with the next step which is image processing part.

3.3.2 Image Processing.

For image processing part is to detect movement in the video. Thus it purpose four step to complete this part, which is:-

- Noise removal: To improve the image quality
- Segmentation: To separate multiple regions in image
- Feature Extraction: To analyze the regions in image
- Tracking : Analyzing the position, velocity and moving direction of object

For this project, a noise removal algorithm is developed by using the median filter.

1. Read the current or input image.
2. For (present position=initial position: final position)

    a. Scan all the surrounding elements
    b. Use bubble sort technique to sort all the values
    c. Calculate the median of it

3. Assign the value to the 'present position' of the input image.

For segmentation part is using frame difference algorithm. This is because it takes less processing time. Frame difference algorithm performs separation of two sequential frames. Here are the details:-

1. Read the input images
2. For (present position=initial position: final position)

    a. Difference between the pixels values at present position of two images is calculated
    b. Calculate the absolute value
    c. Store the difference in new image at same pixel position that is at present position.

Every object has a special feature which is used to visualize the object and used for tracking. A rectangular bounding box is plotted with the dimensions of the object produced in the residual image after the image segmentation step is done. The bounding box is defined as below:-

1. Read the image difference
2. For (present position=initial value: final value) of Y resolution
3. For (present position=initial value: final value) of X resolution
    a. Calculate the sharp change in intensity of image from top and bottom
    b. Store the values in an array
4. Height of the bounding box is = bottom value- top value
5. For (present position=initial value: final value) of X resolution
6. For (present position=initial value: final value) of Y resolution
    a. Calculate the sharp change in intensity of image from left and right
    b. Store the values in an array
7. Width of the bound box = right value - left value
8. Using the dimensions, draw boundary to the image

    Initial value: the starting position of the pixel in an image. Final value: the ending position of the pixel in an image.

$$Height \; = \frac{bottom{:}value \quad top{:}value}{2} \tag{1}$$

$$Width = \frac{right{:}\, value \quad left{:}\, value}{2} \tag{2}$$

9. Add the height value with the top value and store it in a variable like mid top
10. Add the width value to the left value and store it in a variable like mid left
11. Assign the max intensity to the pixel at pixel value at (mid top, mid left)

Distance is calculated using Euclidean distance formula. Euclidean distance formula is a he distance between two points in the plane with coordinates (X1, Y1) and (X2,Y2) is given by

$$Distance = \; \sqrt{(X2 - X1)^2 + (Y2 - Y1)^2} \tag{3}$$

1. Read the centroid position of each image.
2. Calculate the distance between two centroid images
3. for (present position=initial value: final value) of X resolution
4. For (present position=initial value: final value) of Y resolution.
5. Calculate change in distance by using (3) where:-
   X1 = previous pixel position

   X2 = present pixel position in width

   Y1 = previous pixel position

   Y2 = present pixel position in height

As for the velocity, it is determine using the distance travelled by the centroid to the frame rate of the video. The calculating explains as below:-

1. 1. Read the distance travelled by the object

$$velocity = \frac{distance.travelled}{frame.rate}$$

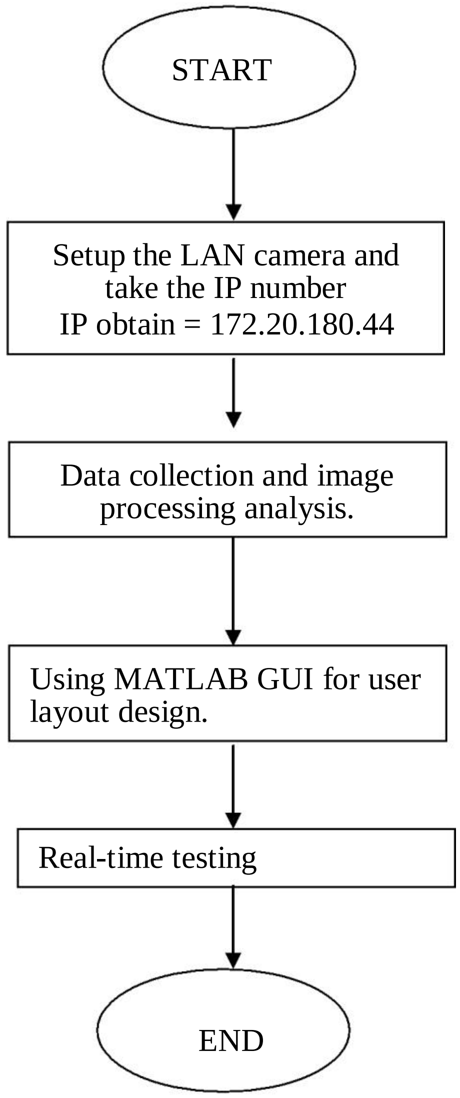
2. Save the value in an array
3. The velocity of moving object in the sequence frames is defined in pixels / second

3.3.3 Graphical User Interface.

This is importance to put all the element need at the GUI design. There are a few steps to follow for designing the GUI.

1. Type “guide” in command window or go to file > new > GUI
2. The new window will prompt and choose “Blank GUI (Default)”.
3. Than the new GUI will start.
4. At the left side is the paraphernalia used such as axes, push button and more.
5. For the video display used the axes, size depends.
6. Double click the axes will prompt a pop ups which is contain the properties of the axes.
7. Same goes to the push button, put as many as for usage and change the tag and string for easy to put the coding for the next part.
8. After complete the GUI, save it. It will automatically save two type of file which is figure file and m file.

The other way to build GUI is custom GUI. This type of GUI is created base on the coding created in the m file. In the m file will have all the properties need for the GUI such as push button and axes. All the paraphernalia need will be initialize and will callback to write the coding for the responding item. To display the GUI is by run the coding.


START
Setup the LAN camera and take the IP number
IP obtain = 172.20.180.44
Data collection and image processing analysis.
Using MATLAB GUI for user layout design.
Real-time testing
END

CHAPTER 4

RESULT AND DISCUSSION

4.1 INTRODUCTION

In this chapter will be discuss about the overall result from the project which is contain from the offline detection to the real-time monitoring.

4.2 Offline Detection.

The analysis of the video is made through to detect the movement done in the video. The video shows a man walk through the frame. The analysis is started by changing the format of the video from RGB to black and white format. Then plot the rectangle around the object base on the pixel change.

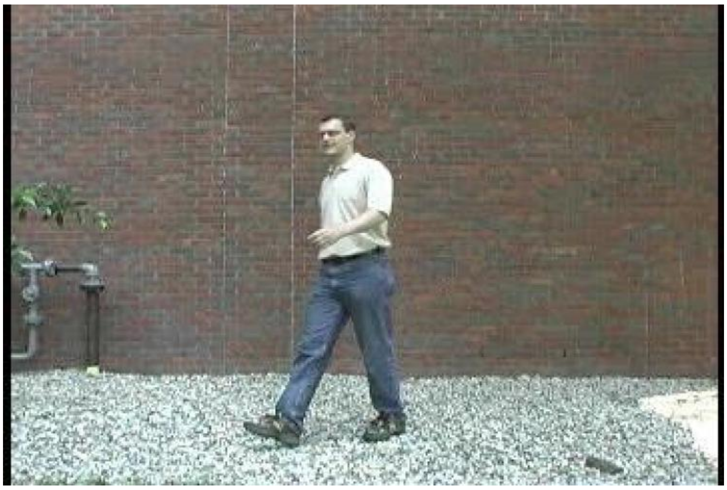


Figure 4: RGB format of the video.

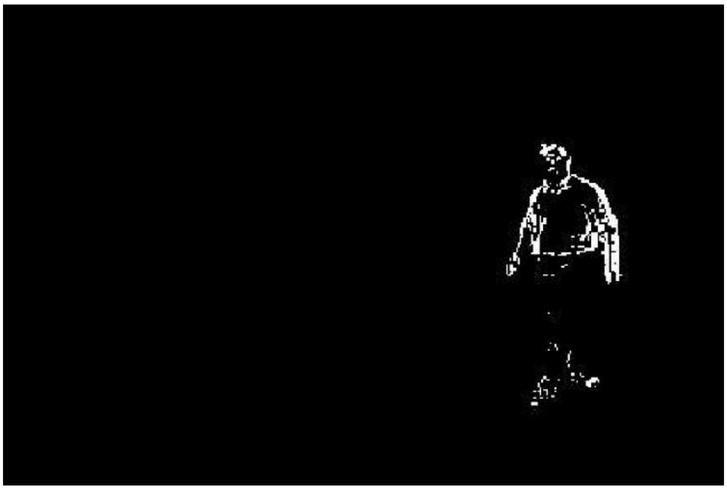


Figure 5: Black and white format of the video.

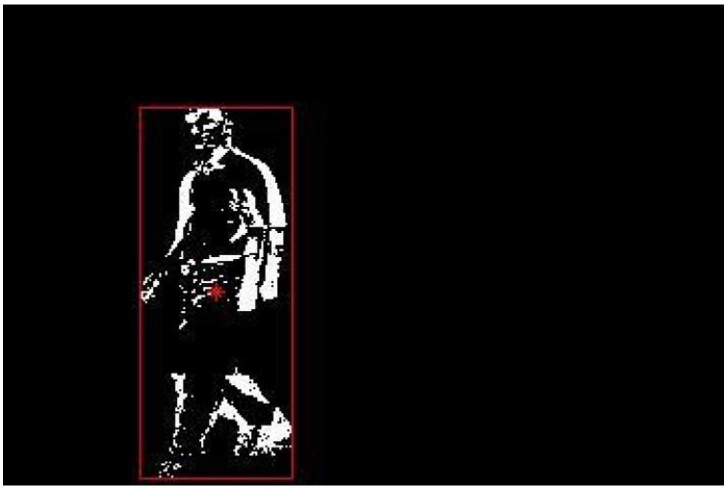


Figure 6: rectangle around the object.

4.3 Real – time monitoring

For this section, the Graphical user interface is constructed. The purpose of this GUI is for the user usage. In the GUI is complete with the button that have their own purpose.

The GUI is construct base on the custom GUI, which is means the GUI is construct base on the coding. The other way to construct GUI is to using guide in the command window and will produce two types of file which is figure file and m file. For the custom GUI will only produce one type of file which is m file. Once the m file is running, the GUI will be display.

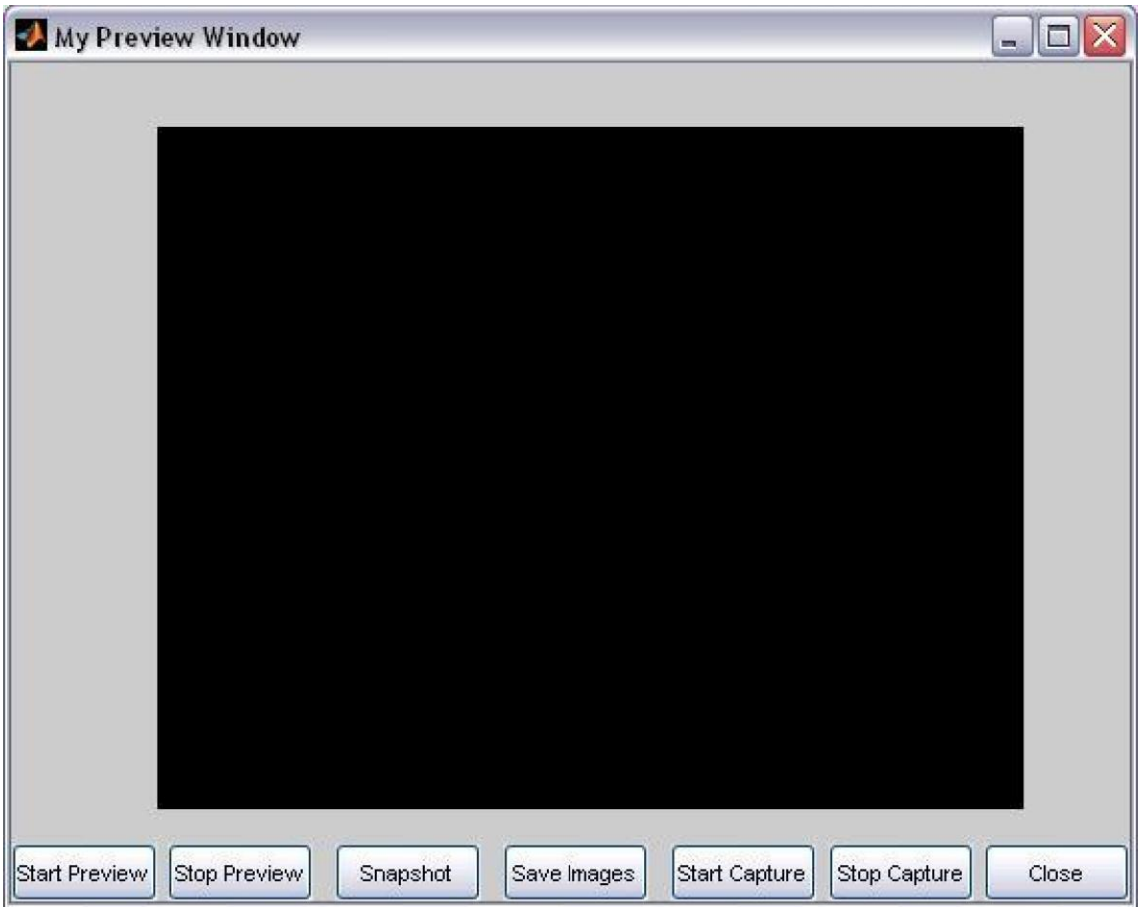


Figure 7: layout of the GUI.

Buttons function.

1. Start preview. This button is for user to start the video preview.
2. Stop preview. This button is to stop the preview of the video.
3. Snapshot. Is to the snap the current frame.

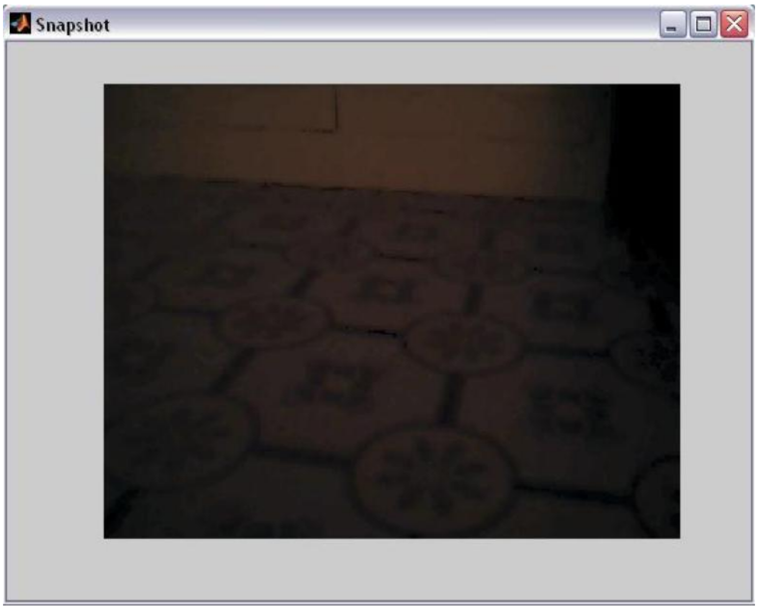
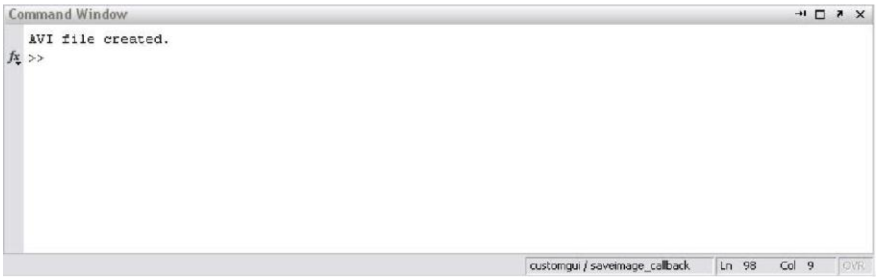


Figure 8: Snapshot figure.

4. Save image. To capture and save the image in the memory.

Figure 9: example of image save in the memory

5. Start capture. To start and save the video in the memory.
6. Stop capture. To stop the video and save in the memory.



```
AVI file created.
>>
```

Figure 10: indicator the video file is created.

7. Close. To close the GUI.

4.4 DISCUSSION.

In this project, the GUI method used is the custom GUI. In MATLAB, there is another way around to produce the GUI which is by using GUIDE. By using GUIDE, it will produce two types of file which is figure file and m file. The difference between these two file is figure file is where all the thing need in the GUI will be display such as button, axes and drop down list. It can be customize. The coding for the buttons or else will be put at the respective place in the m file. The different between the normal m file between m file generate from the GUIDE is the coding inside. The m file generate from the GUIDE already have the coding of the figure file indicate the structure of the GUI. The way to write the coding also difference since GUIDE use handle function.

Custom GUI is created base on the m file solely. The different is it did not have two types of file to display the GUI. The GUI is constructing on the m file. The function used also same as common m file. To arrange the paraphernalia in the GUI is by coordinate and arrange it in term of coding. The drawback is to design the GUI, since it is not like the GUIDE where the programmer could drag and drop the paraphernalia used.

CHAPTER 5

CONCLUSION AND RECOMANDATION.

5.1 Conclusion

In the first part of the project is able to set the CCTV by obtain the IP address is successfully done. The next part is to set the remote monitoring, this part is done by setup the GUI for the real – time monitoring. In the GUI the user able to save the picture up to 100 pictures base on the user interest and in what frame. Other than that, the user also able to save the video. It also depends on the user when to start the video saving and when to stop the video saving. All the pictures and video will be save in the user computer memory.

The last part is remote monitoring, which is allow the user to monitor the CCTV even though they are far away from the CCTV location. Which is by setup the web site for the remote monitoring. This part could not be done because of the certain circumstance happens.

5.2 Recommendation.

The result obtains need improvement by adding the image analysis during real – time or in the other word the image analysis could be done along the video play. Thus during real – time the system able to give the decision either there is movement in the particular frame or not. By adding this feature will add extra features to the monitoring. It also gives the fast alert to the user.

The next that could be done to improve the system is by make it as web site. From the MATLAB itself, it can be integrate by using visual basic, but because of the certain circumstance, the visual basic that runs from the laptop cannot be complete. Other than visual basic, it also can be integrated using JAVA. The other way around that can be done the remote monitoring is by build the system as stand alone. Which means even the user without installing MATLAB also can run the GUI.

REFERENCES

This guide is prepared based on the following references;

[1] Yuriy Kurylyak - *A Real-Time Motion Detection for Video Surveillance System* (IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications 21-23 September 2009, Rende (Cosenza), Italy)

[2] Tianci Huang, Jingbang Qiu, Takahiro Sakayori, Satoshi Goto, Takeshi Ikenaga - *Motion Detection Based On Background Modeling And Performance Analysis For Outdoor Surveillance* (International Conference on Computer Modeling and Simulation)

[3] Lijing Zhang Yingli Liang - *Motion human detection based on background subtraction* (2010 Second International Workshop on Education Technology and Computer Science)

[4] Theekapun Charoenpong, Ajaree Supasuteekul, Chaiwat Nuthong - *Adaptive Background Modeling from an Image Sequence by Using K-Means Clustering*

[5] Nan Lu, Jihong Wang, Q.H. Wu and Li Yang - *An Improved Motion Detection Method for Real-Time Surveillance* (IAENG International Journal of Computer Science, 35:1, IJCS_35_1_16)

[6] Ravi Krishnamurthy, Pierre Moulin and John Woods - *Optical Flow Techniques Applied To Video Coding* (1995 IEEE)

[7] J.L. Barron, D.J. Fleet, S.S. Beauchemin, T.A. Burkitt - *Performance of Optical Flow Techniques* (1992 IEEE)

[8] Simon Denman, Clinton Fookes, Sridha Sridharan - *Improved Simultaneous Computation of Motion Detection and Optical Flow for Object Tracking* (Digital Image Computing: Techniques and Applications, 2009. DICTA '09.)

[9] Meng Liu, Chengdong Wu and Yunzhou Zhang - *Motion vehicle Tracking based on Multi-resolution Optical Flow and Multi-scale Harris Corner Detection* (Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics December 15 - 18, 2007, Sanya, China)

[10] Koji Kinoshita, Masaya Enokidaniy, Masanori Izumida and Kenji Murakami - *Tracking of a Moving Object Using One-Dimensional Optical Flow with a Rotating Observer* (Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference on Digital Object Identifier)

[11] Simon Denman, Clinton Fookes, Sridha Sridharan, and Vinod Chandran - *Object Tracking using Multiple Motion Modalities* (2008 IEEE)

[12] WEI Shuigen, CHEN Zhen, DONG Hua - *Motion Detection Based on Temporal Difference Method and Optical Flow field* (2009 Second International Symposium on Electronic Commerce and Security)

[13] Zhen Yu, Yanping Chen - *A real-time motion detection algorithm for traffic monitoring systems based on consecutive temporal difference* (Asian Control Conference, 2009. ASCC 2009. 7th)

[14] Anthony C. Davies, Jia Hong Yin and Sergio A. Velastin - *Crowd Monitoring Using Image Processing* (IEE Electronic and Communications Engineering Journal, Vol. 7, No. 1 (Feb), pp.37-47)

[15] Appiah, K., Hunter, A., Owens, J., Aiken, P., Lewis, K. - *Autonomous real-time surveillance system with distributed IP cameras* (Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on)

[16] Micucci, D., Oldani, M., Tisato, F. - *Real-time reasoning: the case of surveillance systems* (Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference)

[17] Young-Sook Lee; HoonJae Lee; - *Multiple Object Tracking for Fall Detection in Real-Time Surveillance System* (Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference)

[18] Chin-Hwn Kuo and Tay-Shen Wang - *Design of Networked Visual Monitoring Systems* (ISCAS 2000 - IEEE International Symposium on Circuits and Systems, May 28-31, 2000, Geneva, Switzerland)

[19] Xiaodong Liu Guangda Su - *A New Network-Based Intelligent Surveillance System* (Proceedinas of ICSP2000)

[20] Nahum Kiryati, Tammy Riklin Raviv, Yan Ivanchenko, Shay Rochel - *Real-time Abnormal Motion Detection in Surveillance Video*

APPENDIX A

GANTT CHART

Activity/Month	January	February	March	April	May	June	July	August	September	October
Literature review	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥		
Setup LAN CCTV and IP address	🟥									
Data collection		🟥	🟥							
Image processing analysis.				🟥	🟥	🟥				
GUI design						🟥	🟥	🟥		
Report writing.								🟥	🟥	🟥

APPENDIX B

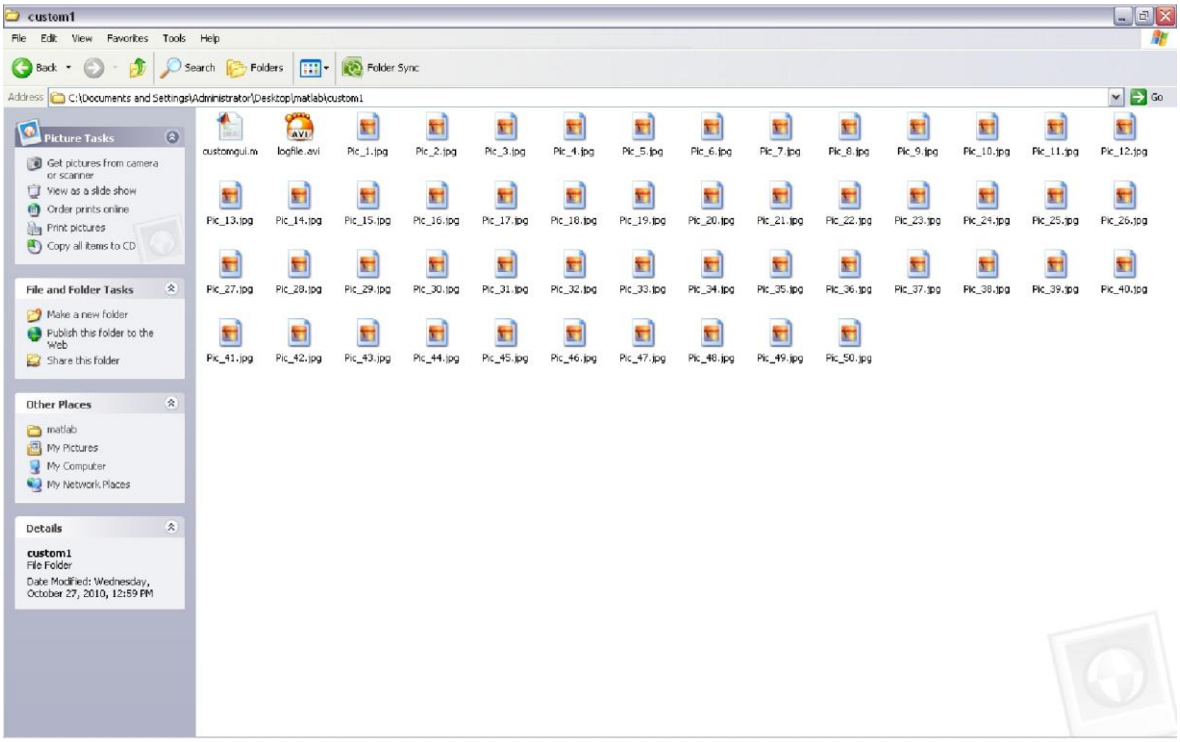
SOURCE CODE

```
function customgui()
%% Create a video input object.
vid = videoinput('winvideo', 1, 'YUY2_320x240');
set(vid,'ReturnedColorSpace','rgb');
%% Create a figure window. This example turns off the
default % toolbar, menubar, and figure numbering.

hFig = figure('Toolbar','none','Menubar', 'none','NumberTitle','Off',...
       'Name','My Preview Window');

%% Set up the push buttons
uicontrol('String', 'Start Preview','Callback',
{@startpreviewvideo_callback},...
    'Units','normalized','Position',[.0 .0 0.13 .07]);

uicontrol('String', 'Stop Preview','Callback',{@stoppreview_callback},...
    'Units','normalized','Position',[.14 .0 0.13 .07]);

uicontrol('String', 'Snapshot','Callback', {@snapshot_callback},...
    'Units','normalized','Position',[0.29 .0 0.13 .07]);

uicontrol('String','Save Images','Callback',{@saveimage_callback},...
    'Units','normalized','Position',[.44 .0 .13 .07]);

uicontrol('String','Start Capture','Callback',
{@startcapture_callback},...
    'Units','normalized','Position',[.59 .0 .13 .07],'Parent',hFig);

uicontrol('String', 'Stop Capture','Callback',
{@stopcapture_callback},...
    'Units','normalized','Position',[.73 .0 .13 .07],'Parent',hFig);

uicontrol('String', 'Close','Callback', 'close(gcf)',...
    'Units','normalized','Position',[.87 .0 .13 .07]);

%% Create the image object in which you want to display
% the video preview data. Make the size of the image
% object match the dimensions of the video frames.

vidRes = get(vid, 'VideoResolution');
nBands = get(vid, 'NumberOfBands');
hImage = image( zeros(vidRes(2), vidRes(1), nBands) );

% figSize = get(hFig,'Position');
% figWidth = figSize(3);
% figHeight = figSize(4);
%
% set(gca,'unit','pixels','position',[ ((figWidth -
imWidth)/2) ((figHeight - imHeight)/2) imWidth imHeight ]);

%% Set up the update preview window function.
```

```
setappdata(hImage,'UpdatePreviewWindowFcn',@mypreview_fcn);
preview(vid, hImage);

%%  setup
    function  mypreview_fcn(obj,event,himage)
    % Example  update  preview  window  function.

    % Get  timestamp  for  frame.
    tstampstr  =  event.Timestamp;

    %  Get  handle  to  text  label  uicontrol.
    ht  =  getappdata(himage,'HandleToTimestampLabel');

    % Set the value of the text label.
    set(ht,'String',tstampstr);

    % Display  image  data.
    set(himage, 'CData',
    event.Data) end

%%  Display  the  video  data  in  your  GUI.
    function startpreviewvideo_callback(hObject,
    eventdata) figure(hFig);
    preview(videopre);
    end
%%  Stop  display  Data  in  GUI
    function stoppreview_callback(hObject,
    eventdata) stoppreview(vid);
    figure(hFig);
    end
%%  snapshot

    function snapshot_callback(hObject, eventdata)
    hFig2 = figure('Toolbar','none','Menubar',
'none','NumberTitle','Off',...
    'Name','Snapshot');
    imagesc(getsnapshot(vid));
    axis('off'); axis('tight');

    %imagesc(ycbcr2rgb(getsnapshot(vid)));
    figure(hFig);
      end
%%  save  video
    function  startcapture_callback(hObject,  eventdata)

%     Create  and  configure  a  video  input  object.

%     vid  =  videoinput('winvideo',  1);
    set(vid,'TriggerFrameDelay',3);
    set(vid,'TriggerRepeat',0);
    set(vid,'Timeout',Inf);
    set(vid,'LoggingMode','disk');
    set(vid,'FramesPerTrigger',1000000);
    file = avifile('logfile.avi','compression','none', 'fps',
30, 'quality', 50);
    set(vid,'DiskLogger',file);
    start(vid)
    end

%%  stop  save  video
    function stopcapture_callback(hObject,
    eventdata) stop(vid);
    file = close(vid.DiskLogger);
        if(exist('logfile.avi')==2)
        disp('AVI file created.')
        end
    end
end
```

APPENDIX C

RESULT OUTPUT



C:\Documents and Settings\Administrator\Desktop\matlab\custom1

customgui.m logfile.avi Pic_1.jpg Pic_2.jpg Pic_3.jpg Pic_4.jpg Pic_5.jpg Pic_6.jpg Pic_7.jpg Pic_8.jpg Pic_9.jpg Pic_10.jpg Pic_11.jpg Pic_12.jpg

Pic_13.jpg Pic_14.jpg Pic_15.jpg Pic_16.jpg Pic_17.jpg Pic_18.jpg Pic_19.jpg Pic_20.jpg Pic_21.jpg Pic_22.jpg Pic_23.jpg Pic_24.jpg Pic_25.jpg Pic_26.jpg

Pic_27.jpg Pic_28.jpg Pic_29.jpg Pic_30.jpg Pic_31.jpg Pic_32.jpg Pic_33.jpg Pic_34.jpg Pic_35.jpg Pic_36.jpg Pic_37.jpg Pic_38.jpg Pic_39.jpg Pic_40.jpg

Pic_41.jpg Pic_42.jpg Pic_43.jpg Pic_44.jpg Pic_45.jpg Pic_46.jpg Pic_47.jpg Pic_48.jpg Pic_49.jpg Pic_50.jpg

custom1
File Folder
Date Modified: Wednesday, October 27, 2010, 12:59 PM

Figure 11: Files save in the memory