MOTION DETECTION USING BACKGROUND SUBTRACTION TECHNIQUE

By

MOHAMED SALAH MOHAMED ELAWAD

Supervisor

Dr. Abdalrhman Ali Karrar

REPORT SUBMITTED TO

University of Khartoum

In partial fulfillment of the requirement for the degree of

B.Sc. (HONS) Electrical and Electronic Engineering

(CONTROL ENGINEERING)

Faculty of Engineering

Department of Electrical and Electronic Engineering

July 2011

DECLARATION OF ORAGINALITY

I hereby declare that the thesis is based on my original work except as cited in the references. I also declare that this thesis has not been previously or currently submitted for any other degree at University of Khartoum or any other institution.

_____

MOHAMED SALAH MOHAMED ELAWAD

Date:

ACKNOWLEDGEMENT

Thanks to ALMIGHTLY ALLAH the most merciful, benevolent and beneficial, who has enabled me to complete this work and by the grace of HIS HOLY Prophet MOHAMED, (peace be upon him) who is forever a source of enlightenment, guidance and knowledge for humanity as a whole.

At the very outset, I would like to express my heartiest and sincere gratitude to my supervisor Dr. Abdalrhman Karrar for his untiring and inspiring guidance, patience and close supervision throughout the period of my study.

I wish to thank my colleagues and friends who have extended assistance to me whenever it was needed, especially my partner Ahmed Mohamed Madani for sharing the literature and invaluable assistance.

I will forever be grateful and be indebted to my affectionate parents and brother for their blessings and encouragement. They have always motivated me to accomplish my goals.

Last but not least, I would like to express my deepest gratitude to the ENGINEERING FACULTY, UNIVERSITY OF KHARTOUM for their support.

ABSTRACT

Identifying moving objects is a critical task for many computer vision applications; it provides a classification of the pixels into either foreground or background. A common approach used to achieve such classification is background subtraction, where each video frame is compared against a reference or background model, pixels that deviate significantly from the background are considered to be moving objects. Even though there exist numerous of background subtraction algorithms in the literature, most of them follow a simple flow diagram, passing through three major steps, which are pre-processing, background modeling, foreground detection, and data validation also referred to as post-processing.

In this project we describe a new algorithm for motion detection, which is useful for applications that have static background. Firstly, the background subtraction technique is implemented to detect moving objects. Secondly, some modifications are introduced on this base algorithm with the purpose of recording the moving objects.

The results obtained from the application of the background subtraction algorithm for motion detection revealed better performance compared to the other applied techniques.

المستخلص

تعتبر عملية تميَز الأجسام المتحركة من المهام الحرجة للعديد من تطبقات الحاسوب ،لأنَها تُقسِم نقاط الشاشة الى نقاط أمامية او نقاط خلفية. ومن أكثر الطرق المستخدمة لإنجاز هذا التقسيم هي طريقة طرح الخلفية و التي يتم فيها مقارنة كل إطار فيديو بإطار مرجعي أو خلفي. النقاط ذات التغير الكبير من نقاط الإطار المرجعي أو الخلفي تعتبر الهدف المتحرك.

بالرغم من أنه توجد خوارزميات عديدة لتحديد الأهداف المتحركة في الإطار الأمامي كالتي في القسم الأدبي من التقرير إلا أنْ جميعها لها نفس مسار أو طريقة الأداء لإكتشاف الأهداف المتحركة والتي ترتكز على أربع خطوات رئيسات: معالجة اولية للإطار الأمامي، و ترتيب الإطار الخلفي، وإكتشاف الهدف المتحرك في الإطار الأمامي، ومدى صلاحيتها.

في هذا المشروع يتم إدراج خوارزمية جديدة لإكتشاف الأهداف المتحركة للتطبيقات ذات الخلفية الساكنة. اولاً طريقة طرح الخلفية أسْتُخدمَتْ لإكتشاف الأهداف المتحركة في الإطار الأمامي، بطرح الإطار الحالي مع الإطار المرجعي، ثم ثانياً تم إجراء يعض التعديلات على هذه الطريقة لتسجيل أو تخزين إطار الهدف المتحرك.

أظهرت النتائج أن إستخدام خوارزمية طرح الخلفية في إكتشاف الأهداف المتحركة أدى الى اداء أفضل مقارنة مع الطرق الاخرى المستخدمة.

Table of Contents

Declaration of originality ……...……...i

Acknowledgement ……..……ii

Abstract ……...…iii

المستخلص ……...iv

Table of contents ……...……...v

List of figure ……..……...viii

List of table ……..……...ix

Chapter 1 1

1.1 Introduction 1

1.1.1 Computer Vision 1

1.1.2 Motion Detection 2

1.2 Problem Statement 3

1.3 System Setup 3

1.4 Project Objectives 3

1.5 Scope of Project 4

1.6 Thesis Layout 4

2 Chapter 2 5

2.1 Object Detection 5

2.1.1 Background Subtraction Technique 5

2.1.1.1 Preprocessing ……...……6

2.1.1.2 Background modeling ……..……6

2.1.1.3 Foreground detection ……7

2.1.1.4 Data validation ……..……..7

2.1.2 Frame differencing Technique 8

2.1.3 Comparison between the two algorithms 9

2.2 IP Cameras 10

2.2.1 Overview 10
2.2.2 Camera types 11
2.2.3 Video Streaming 11
2.3 Applied Algorithms 12
2.3.1 Background Subtraction Techniques [Alan M. McIvor, 2000] 12
2.3.2 Background Subtraction Techniques [Nikom Suvonvorn, 2007] 13
3 Chapter 3 14
3.1 Overview 14
3.2 System Block Diagram 16
3.2.1 IP Camera Connection 17
3.2.2 Video Acquisition 17
3.2.3 Grayscale Conversion 18
3.2.4 Background Subtraction 19
3.2.5 Thresholding 21
3.2.6 Noise Filtering 24
3.2.7 Motion Detection (Decision Making) 25
3.2.8 Alarm and Recording 27
4 Chapter 4 28
4.1 Results from Motion Detection Algorithm 28
4.2 Discussion of the Results 32
4.3 System validation 33
4.3.1 Sudden Illumination Changes 33
4.3.2 Unable to Detect Minor Motion 34
4.3.3 Unable to detect motion at far distances 34
5 Chapter 5 36
5.1 Summary 36
5.2 Conclusion 36
5.3 Obstacles of the research project 36
5.4 Recommendations 37

Appendix A …………………………………….…………………………….……..……A-1

Appendix B………………………………………………………….…………..B-1

List of Figures

Figure 2.1: Fixed network cameras 11
Figure 3.1: Compro IP 50w 14
Figure 3.2 Vision system block diagram 17
Figure 3.3: Image shows gray-scale conversion 19
Figure 3.4: Gray-scale image 20
Figure 3.5: Gray-scale image 20
Figure 3.6: Image shows the absolute difference between figures 3.5 and figure 3.6 21
Figure 3.7: Image shows binary threshold type 23
Figure 3.8: The flow of noise filtering 25
Figure 3.9: Image shows the result of background subtraction without filtering 26
Figure 3.10: Image shows figure 3.9 after filtering 27
Figure 4.1: Colour image show the initial frame taken by the camera 29
Figure 4.2: Gray-scale image show the initial frame taken by the camera 30
Figure 4.3: Gray-scale video (stream) shows the current frames taken by the camera 31
Figure 4.4: Binary image shows the result of background subtraction process 32
Figure 4.5: Image shows effect of sudden illumination changes 33
Figure4.6: Image shows that the significant motion 34
Figure 4.7: shows the failure of detecting objects at far distance 35

List of Tables

Table 2.1: Difference between Background subtraction and Frame differencing algorithms 9

Table 3.1: Various type of threshold operations 23

Chapter 1

INTRODUCTION

1.1 Introduction

The capability of extracting moving objects from a video sequence is a fundamental and critical problem of many computer vision applications that include video surveillance, traffic monitoring, human detection and tracking for video teleconferencing or human-machine interface, video editing, among other applications.

1.1.1 Computer Vision

Computer vision is the transformation of data from a still or video camera into either a decision or a new representation. All such transformations are done for achieving some particular goal. The input data may include some contextual information such as the mounted in a car or laser range finder indicates an object is far away. The decision may be there is a person in this scene or there are tumor cells on this side. A new representation might mean turning a color image into a gray scale image or removing camera motion from an image sequence [1].

Computer vision can also be described as a complement (but not necessarily the opposite) of biological vision. In biological vision, the visual perception of humans and various animals are studied, resulting in models of how these systems operate in terms of physiological process. Computer vision, on the other hand, studied and describes artificial vision systems that are implemented in software and/or

hardware. Interdisciplinary exchange between biological and computer vision has proven increasingly fruitful for both fields.

Computer vision systems have been developed to simulate most biological systems which have the ability to cope up with changing environments such as moving objects, changing illumination and changing viewpoints. Detection and tracking of moving objects can be viewed as lower level vision tasks to achieve higher level event understanding [2].

1.1.2 Motion Detection

Motion detection is the action of sensing physical movement in a given area. Motion detection in general is handled in many different libraries, and has been utilized in several applications.

For the motion detection module we chose to implement the project using Processing. Processing is a C++ based visualization language that is simple to learn, and would create the best opportunity for the development of user friendly features. OpenCV [1] was another library that was considered for the project. It is a C/C++ based visualization library. Processing was chosen because it already had much functionality we were looking for in this project.

A simple algorithm for motion detection by a fixed camera compares the current image with a reference image and simply counts the number of different pixels. Since images will naturally differ due to factors such as varying lighting, camera flicker, and preprocessing is essential to reduce the number of false positive alarms.

More complex algorithms are necessary to detect motion when the camera itself is moving, or when the motion of a specific object must be detected in a field containing other movement which can ignored.

Moving object detection provides a classification of the pixels in the video sequence into either foreground (moving objects) or background. A common approach used to achieve such classification is background removal, sometimes

referred to as background subtraction, where each video frame is compared against a reference or background model, pixels that deviate significantly from the background are considered to be moving objects [2].

1.2 Problem Statement

Automatic motion detection is an important issue for many applications such as: surveillance and security applications.

In this project I try to develop a background subtraction algorithm for motion detection applied in static scene.

1.3 System Setup

This system will only require a simple IP camera setup on a computer. Other interactive surfaces might require different setups, but for the scope of this project the focus will stay on IP camera motion detection. I used a 150w-ip that has an Ethernet connection.

1.4 Project Objectives

The main aim of this project is to:

1. Build static IP camera that can detect motion in two dimensional views.
2. Develop an algorithm that would detect motion from a sequence of images by using background subtraction technique and record the detected motion.

1.5 Scope of Project

This project is developed in a Microsoft Visual C++. The software developed will be run on a PC for processing and monitoring purposes. Scopes of this project:

1. Perform access to IP camera.
2. The algorithm will be intelligent to detect motion in the IP camera capture frame.

1.6 Thesis Layout

This thesis consists of 5 chapters and each chapter is briefly discussed here.

- Chapter 1 gives an introduction to the project and overview of this project. The project's problem statement, system setup, objective and scope also included in this chapter.
- Chapter 2 discussed some algorithms available and the selected algorithm and some of the literature reviews that are related to this project.
- Chapter 3 discussed the methodologies implement in this project. The motion detection algorithm is discussed in this chapter. The process and implementation is presented in details through the block diagrams.
- Chapter 4 elaborates the results or outcomes of this project and its limitations.
- Chapter 5 covers the conclusion and recommendations for future development.

Chapter 2

LITERATURE REVIEW

2.1 Object Detection

In general, the project concerned for motion detection. After reviewed several publication papers and algorithms available, there are different types of algorithms can be used.

Several concepts, theories and algorithms of image processing are discussed. It is aimed to provide related information needed to fully understand this project and reasons of the algorithm chosen for the project.

2.1.1 Background Subtraction Technique

The common approach for discriminating moving object from the background scene is background subtraction. The idea is to subtract the current image from a reference image, which is acquired from a static background during a period of time. The subtraction leaves only non-stationary or new objects, which include the object's entire silhouette region.

There are numerous of background subtraction algorithms, most of them follow a simple flow diagram, passing through four major steps, which are [2]:

(1) Pre-processing, simple image processing tasks that change the raw input video into a format that can be processed by subsequent steps.

(2) Background modeling, also known as background maintenance.

(3) Foreground detection.

(4) Data validation, also referred to as post-processing, used to eliminate those pixels that do not correspond to actual moving objects

Although the terms background subtraction and background modeling are often used interchangeably, they are separate and distinct processes. Background modeling refers to the process of creating, and subsequently maintaining, a model of the appearance of the background in the field of view of a camera. Background subtraction refers to the process in which an image frame is compared to the background model in order to determine whether individual pixels are part of the background or the foreground. So it is also referred to as foreground detection.

2.1.1.1 Pre-processing

The goal of a moving object detection algorithm is to detect significant changes occurring throughout the video sequence while rejecting unimportant ones. The pre-processing used to filter out common types of unimportant changes before making the object detection decision.

For real-time systems, frame-size and frame-rate reduction are commonly used to reduce the data processing rate. Simple temporal and/or spatial smoothing is often used in the early stage of pre-processing to reduce camera noise and to remove transient environmental noise such as rain and snow captured in outdoor applications.

2.1.1.2 Background Modeling

Background modeling, also referred to as background maintenance, is at the heart of any background subtraction algorithm. The module performing background modeling should not attempt to extract the semantics of foreground objects on its own, since it is not an end by itself; larger systems use it as a component. One can evaluate background modeling by how closely it comes to finding all foreground pixels (as defined by the end task) when a foreground object first appears in the scene, while simultaneously ignoring all others. Backgrounds are not necessarily defined by absence of motion, e.g. waving trees.

2.1.1.3 Foreground Detection

Foreground detection compares the input video frame with the background model, and identifies candidate foreground pixels from the input frame. To obtain this classification, the difference map is usually binaries by thresholding. The correct value of the threshold depends on the scene, on the camera noise, and on the illumination conditions.

2.1.1.4 Data Validation

The output of a foreground detection algorithm where decisions are made independently at each pixel will generally be noisy, with isolated foreground pixels, holes in the middle of connected foreground components, and jagged boundaries. Data validation defined as the process of improving the candidate foreground mask based on information obtained from outside the background model. Data validation phase is sometimes referred to as the post-processing phase of the foreground mask (pixels).

There are various factors that cause the noise in foreground detection such as:

(i) Camera noise

This is the noise caused by the camera's image acquisition components. The intensity of a pixel that corresponds to an edge between two different colored objects in the scene may be set to one of the object's color in one frame and to the other's color in the next frame.

(ii) Reflectance noise

When a source of light, for instance sun, moves it makes some parts in the background scene to reflect light. This phenomenon makes the foreground detection algorithms fail and detect reflectance as foreground regions.

(iii) Background colored object noise

Some parts of the objects may have the same color as the reference background behind them. This resemblance causes some of the algorithms to detect the corresponding pixels as non-foreground and objects to be segmented inaccurately.

(iv) Shadows and sudden illumination change

Shadows cast on objects are detected as foreground by most of the detection algorithms. Also, sudden illumination changes (e.g. turning on lights in a monitored room) makes the algorithms fail to detect actual foreground objects accurately.

Morphological operations, erosion and dilation [2], are applied to the foreground pixel map in order to remove noise that is caused by the first three of the items listed above. Our aim in applying these operations is removing noisy foreground pixels that do not correspond to actual foreground regions and to remove the noisy background pixels near and inside object regions that are actually foreground pixels. Erosion, as its name implies, erodes one-unit thick boundary pixels of foreground regions. Dilation is the reverse of erosion and expands the foreground region boundaries with one-unit thick pixels. The subtle point in applying these morphological filters is deciding on the order and amounts of these operations. The order of these operations affects the quality and the amount affects both the quality and the computational complexity of noise removal.

The results of the existing algorithms are fairly good; in addition, many of them run in real-time.

2.1.2 Frame differencing Technique

Frame differencing is a non-recursive technique, also known as temporal difference, uses the video frame at time t-1 as the background model for the frame at time t. This technique is sensitive to noise and variations in illumination, and does not consider local consistency properties of the change mask. An important disadvantage while tracking people is that when someone is not moving for a very short time (one frame time), the person will become part of the background. This method also fails to segment the non-background objects if they stop moving. Since it uses only a single previous frame, frame differencing may not be able to identify the interior pixels of a large, uniformly-colored moving object. Main advantages are the small computational load, little memory space needed and it’s high adaptively [2].

For the differential method, it analyzes the difference between of two following images in a sequence by using the following formula:

t1, t2: time moments

FD: difference of images

x, y: images coordinate

2.1.3 Comparison between the two algorithms

From the two previous algorithms, it is cleared that both algorithms works well with for the image processing to detect motion. Some significant differences of the two algorithms are noticed in the Table 2.1.

Temporal Differencing Method	Background Subtraction Technique
• Pixel-by-pixel difference of consecutive frames. • Very adaptive to dynamic environments. • Unable to extract feature of object.	• Subtracting current image pixel-by-pixel from a reference image. • Sensitive to dynamic changes • Able to extract feature of object.

Table 2.1: Difference between Background subtraction and Frame differencing algorithms

2.2 IP Cameras

2.2.1 Overview

An IP camera can be described as a camera and computer combined in one unit. The main components of a network camera, as depicted in include a lens, an image sensor, one or several processors, and memory. The processors are used for image processing, compression, video analysis and networking functionalities. The memory is used for storing the network camera‘s firmware (computer program) and for local recording of video sequences [7].

Like a computer, the network camera has its own IP address, is connected directly to a network and can be placed wherever there is a network connection. This differs from a web camera, which can only operate when it is connected to a personal computer (PC) via the USB or IEEE 1394 port, and to use it, software must be installed on the PC. A network camera provides web server, FTP (File Transfer Protocol), and e-mail functionalities, and includes many other IP net-work and security protocols.

IP cameras come with a wide variety of features such as fixed lenses, pan-tilt-zoom control, endless 360 degree panning, indoor or outdoor use, 2-way audio, infrared illumination, intelligent video analytics etc. Yet the main advantage of such cameras is the spectacular resolution, compression schemas capabilities, and the true digital recording to dedicated recording devices or to hard disks on local PCs or across the Internet.

2.2.2 Camera types

Network cameras, often also called IP-based camera can be classified in terms of whether they are designed for indoor use only or for indoor and outdoor use. Outdoor network cameras often have an auto iris lens to regulate the amount of light the image sensor is exposed to. An outdoor camera will also require an external, protective housing unless the camera design already incorporates a protective

enclosure. Housings are also available for indoor cameras that require protection from harsh environments such as dust and humidity, and from damage or tampering. In some camera designs, vandal and tamper-proof features are already built-in and no external housing is required [7].

Network cameras, whether for indoor or outdoor use, can be further categorized into fixed, fixed dome, PTZ, and PTZ dome network cameras.

Figure 2.1: Fixed network cameras

2.2.3 Video Streaming

There are two ways to view media on the internet (such as video, audio, animations, etc): Downloading and Streaming. Progressive downloads are typically cheaper and work with any server, but do not offer content protection nor seeking to undownloaded parts. Streaming offers these functionalities. In recent years, audio/video streaming has become a popular application and significant consumer of network bandwidth. In video streaming, clients request compressed video files that reside on server. This server can be ordinary web server or can be special streaming server tailored for the video streaming application [11].

Streaming works by first compressing a digital file and then breaking it into small packets, which are sent, one after another, over the Internet. When the packets reach their destination (the requesting user), they are decompressed and reassembled

into a form that can be played by the user's system. To maintain the illusion of seamless play, the packets are "buffered" so a number of them are downloaded to the user's machine before playback. As those buffered or preloaded packets play, more packets are being downloaded and queued up for playback. However, when the stream of packets gets too slow (due to network congestion), the client player has nothing to play.

2.3 Applied Algorithms

There are a lot of relevant information and technical papers published on the internet. There will be some discussion over the information discovered.

2.3.1 Background Subtraction Techniques [Alan M. McIvor, 2000]

Background subtraction is a commonly used class of techniques for segmenting out objects of interest in a scene for applications such as surveillance. Background extraction techniques emphasize on three important attributes:

(i) Foreground detection,

(ii) Background maintenance,

(iii) Post-processing.

The concept of background subtraction techniques is the comparison of an observed image with reference image to show an estimate of the image if it contained no objects of interest.

3.2.2 Background Subtraction Techniques [Nikom Suvonvorn, 2007]

There are a few concepts introduced by the paper:

i. To detect foreground objects by taking the difference of current frame and background.

$$|framei - backgroundi| > T$$

ii. To update background image that is not fixed.

$$|framei - frame(i-1)| > T$$

Chapter 3

METHODOLOGY

3.1 Overview

As a result from the analysis shown in chapter two, the algorithm chosen for the project is background subtraction. This due to the project applied on static background scene, so no need to update background reference (initial frame). Additional to that, the feature of the object is essential in this project.

The video capture device in this project is IP camera (150w-ip, 50w). The resolution of the IP camera is 640x480 and it can capture up to 30 frames per second. However the video output resolution is scaled down to 320x240 in order to enhance the processing time of the project.

Figure 3.1: 150w-ip camera

OpenCV is used as the video input/output and image processing tools. OpenCV means Intel® Open Source Computer Vision Library. It is an open source computer vision library. The library is written in C and C++ that implemented for popular algorithms of image processing and computer vision and run under Linux, Windows and Mac OS X.

OpenCV was designed for computational efficiency and with a strong focus on real time applications. OpenCV can take advantage of multi-core processors.

In this project it is implemented as additional library (Lib) for windows and provides a simple application program interface for reading and controlling video stream, processing its frame and rendering the result. The following functions explain the capture of video using OpenCV:

a. Declaration of CvCapture, the structure is used for getting video from camera or AVI file. The structure CvCapture does not have public interface and is used only as parameter for video.

b. Initialize the camera device by cvCreateFileCapture; it allocates the CvCapture structure for reading the video stream from the specified file. Which codecs and file formats are supported depends on the back end library. The camera interface used in the project is video for windows (VFW).

c. In response to the (b) function, the cvReleaseCapture is used to release the CvCapture structure. It can release the CvCapture structure allocated by cvCreateFileCapture.

d. The function cvQueryFrame grabs a frame from a camera or video file, decompresses it and returns it. This function is just a combination of cvGrabFrame, stores the frame internally, and cvRetrieveFrame, get a pointer to the grabbed frame, but in one call.

The operating block diagram of the algorithm is shown in the *figure 3.2*. The system blocks diagram starting from the acquisition of the image from IP camera till the end of the result. *Figure 3.1* shows the sequence of the processing starting from the image acquisition from IP camera to the result of the processed image and decision made.

3.2 System Block Diagram

At the early stage of the block diagram, the CvCapture structure needs to be initialized and using the cvCreateFileCapture to acquire image from the IP camera. Next, it proceeds to the frame grabber to grab the current frame. Once the frame is grabbed, the image frame is converted into grayscale image frame.

The background subtraction algorithm subtracts the current image from a reference image, which is acquired from a static background at beginning.

The absolute intensity value of the result image indicated that whether there is motion detected or not.

Alarm has been setting to indicate a motion or new entry is detected, and then the system trigger to recording.

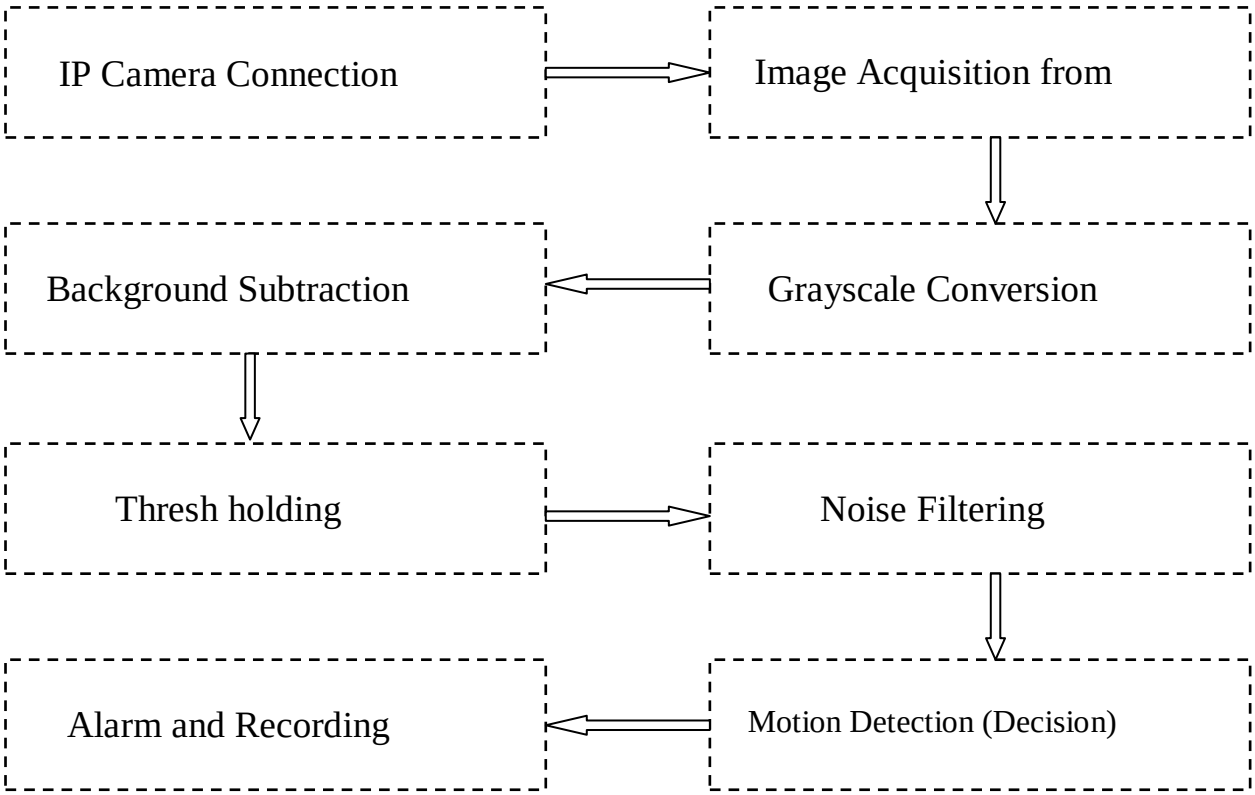


Figure 3.2: Vision system block diagram

3.2.1 IP Camera Connection

The most IP cameras had Ethernet socket which used to connect the camera to PC via Ethernet cable. In this project crossover cable was used to connect the camera (ip50w) with PC. The following IP address "rtsp://192.168.0.77:554/medias1" was been given to the IP camera so that can the PC can see the camera.

3.2.2 Video Acquisition

The First step is to capture the frame; video acquisition from IP camera is done by, OpenCV function, CvCreateFileCapture. The video sequence is then being sampled into multiple frames.

3.2.3 Grayscale Conversion

Once the frame is acquired from IP camera, the frame has to go through the chain of the pre- processing. The first process is gray scaling. In image processing, a grayscale image is an image which the value of each pixel is a single sample, or we called it intensity value. This sort of image is typically composed of shades of gray, varying from black at the weakest intensity to white at the strongest intensity. Grayscale images are distinct from black-and-white images, which in the context of computer imaging are images with only two colors, black and white; grayscale images have many shades of gray in between. The intensity of a pixel is expressed within a given range between a minimum and a maximum, inclusive. This range is represented in an abstract way as a range from 0 (total absence, black) and 1 (total presence, white), with any fractional values in between.

Gray scaling is essential to remove the color values of the image and converts the image into a grayscale image.

The grayscale image simplifies computation drastically, compared to a color RGB image. Conversion of a color image to grayscale is not unique; different weighting of the color channels effectively represents the effect of shooting black-and-white film with different-colored photographic filters on the cameras. A common strategy is to match the luminance of the grayscale image to the luminance of the color image. There are several algorithms which can be used to convert color image to grayscale image. The function cvConvertImage is used to convert input image from one color space to another. Transformation within RGB space to grayscale using luminance model (Y):

$$Y = 0.212671 * R + 0.71516 * G + 0.072169 * B$$

where all the pixel values are changed to that of the luminance model result(Y).

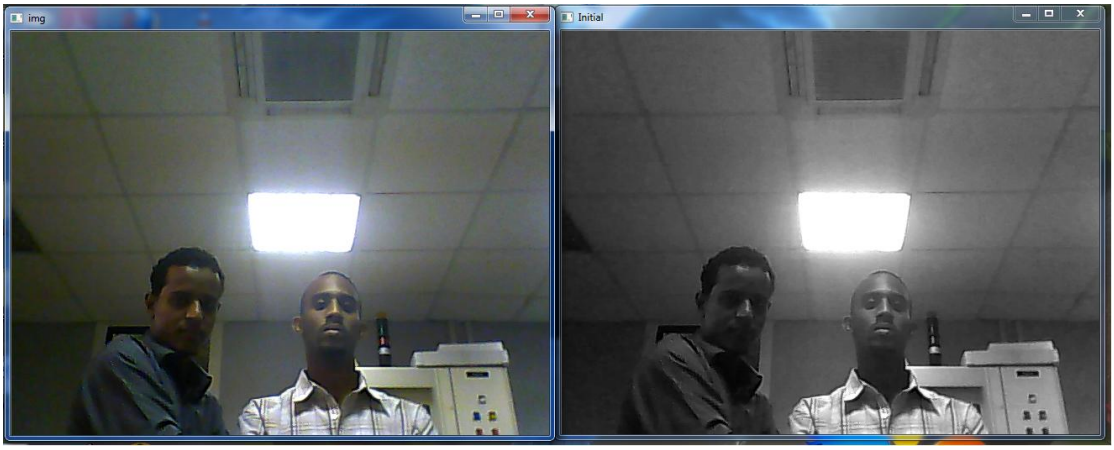


Figure 3.3: Image shows gray-scale conversion

3.2.4 Background Subtraction

Background Subtraction is an image processing technique used to determine changes between changes images. The difference between the reference image and current image is calculated by finding the difference between each pixel in each image, and generating an image based on the result.

The static IP camera of the project meets the requirements of the Background Subtraction method to be use.

The function cvAbsDiff is used to calculate absolute difference between two images. All the images must have the same data type and the same size.

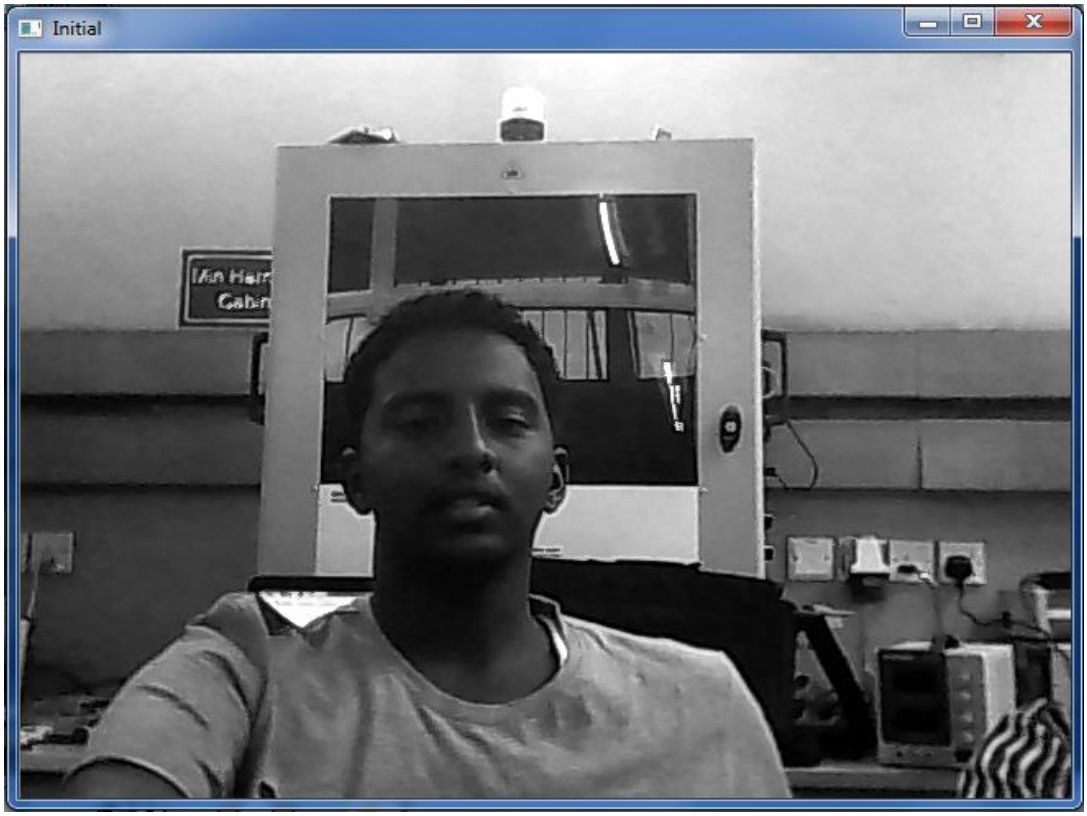


Figure 3.4: Gray-scale image

Figure 3.5: Gray-scale image

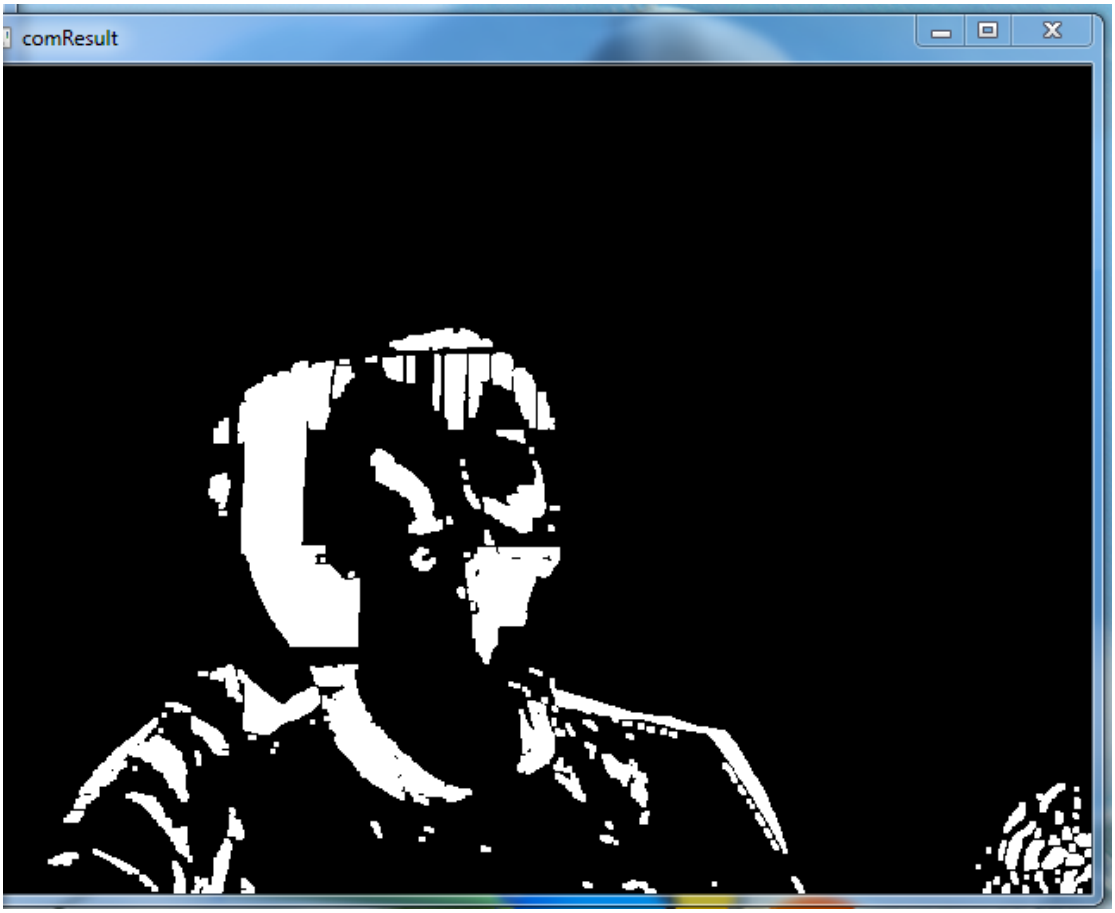


Figure 3.6: Image shows the absolute difference between figures 3.4

And figure 3.5

3.2.5 Thresholding

Thresholding is a fundamental method to convert a gray scale image into a binary mask, so that the objects of interest are separated from the background. In the difference image, the gray levels of pixels belonging to the foreground object should be different from the pixels belonging to the background. Thus, finding an appropriate threshold will solve the localization of the moving object problem.

The output of the thresholding operation will be a binary image whose gray level of 0 (black) will indicate a pixel belonging to the background and a gray level of 1 (white) will indicate the object. The efficiency of the foreground detection partially depends on the threshold selection. The threshold can be set empirically or computed adaptively. In the former case, the threshold is fixed for all pixels in the frame and all the frames in the sequence. The value is usually determined experimentally based on a large database.

The function cvThreshold applies fixed-level thresholding to a single-channel array. The function is typically used to get a bi-level (binary) image out of a grayscale image or for removing a noise, i.e. filtering out pixels with too small or too large values.

There are several types of thresholding that the function supports; the most suitable one is the threshold binary.

The types of threshold operations:

(i) CV_THRESH_BINARY val = (val > Thresh? MAX: 0)

(ii) CV_THRESH_BINARY_INV val = (val > Thresh? 0: MAX)

(iii)CV_THRESH_TRUNC val = (val > Thresh? Thresh: val)

(iv)CV_THRESH_TOZERO val = (val > Thresh? val: 0)

(v) CV_THRESH_TOZERO_INV val = (val > Thresh? 0: val)

Table 3.1: Various type of threshold operations

	Value & threshold level
	Threshold binary
	Threshold binary, inverted
	Truncate
	Threshold to zero, inverted
	Threshold to zero

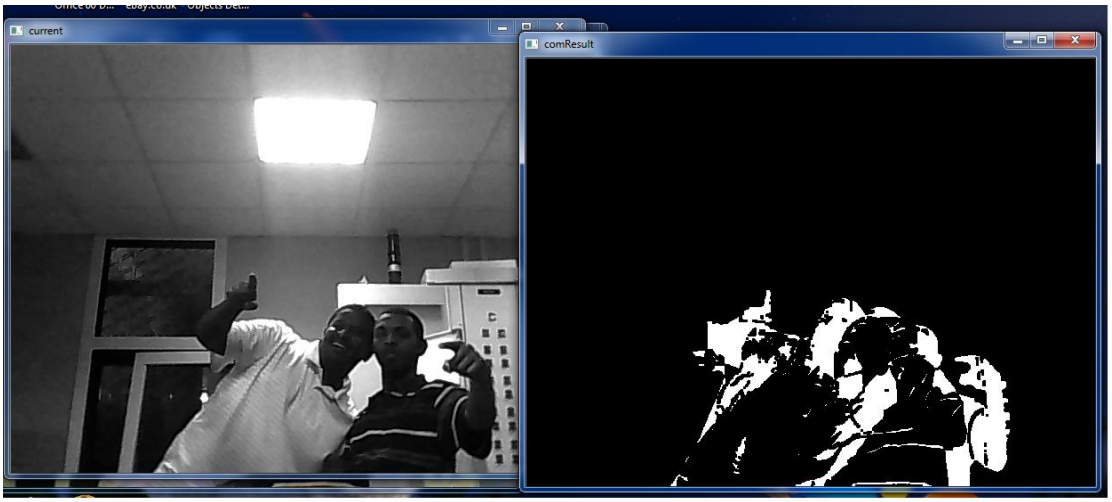


Figure 3.7: Image shows binary threshold type

3.2.6 Noise Filtering

After thresholding, image enhancements are performed using the Processing library's image filters. The image is filtered to eliminate noise, and to develop more contrast in the areas of interest.

Therefore, morphological operations, particular erosion and dilation are applied to the foreground image in order to remove the noise caused. The image is run through an erosion filters, and then a dilation filters. Erosion is used to eliminate the smaller areas of pixels and dilation to reestablish the remaining areas of pixels.

The function cvErode erodes the source image using the specified structuring element that determines the shape of a pixel neighborhood over which the minimum is taken:

$$\min_{(x\prime,y\prime)\ in\ element} src(x' + x, y' + y)$$

The function cvDilate dilates the source image using the specified structuring element that determines the shape of a pixel neighborhood over which the maximum is taken:

$$\max_{(x\prime,y\prime)\ in\ element} src(x' + x, y' + y)$$

These two functions support the in-place mode. Erosion and dilation can be applied several times. For color images, each channel is processed independently.

The sequence of the operation is one level of erosion followed by one level of dilation. There is a need initialize the structuring element by creating a mask. The steps of the morphological operation are shown in *figure 3.8*.

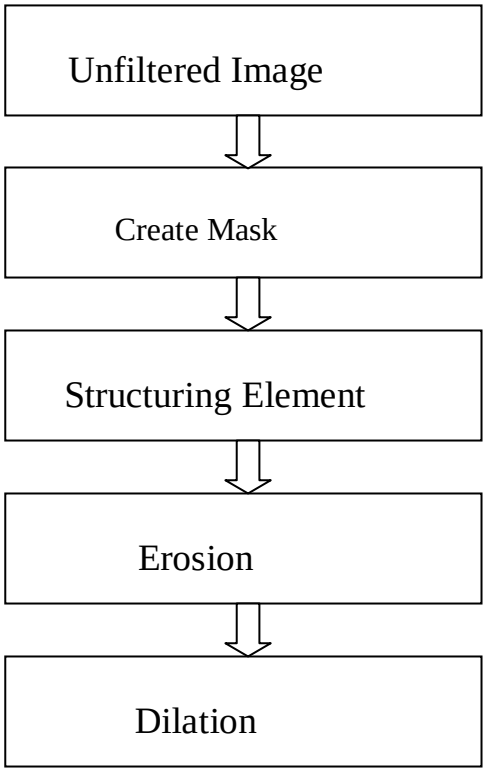


Figure 3.8: The flow of noise filtering

3.2.6 Motion Detection (Decision Making)

This part of the processing is responsible to detect the motion on the scene. After the image processing blocks, the information of the result image can be used to indicate whether motion present on the scene.

The function cvCountNonZero returns the number of non-zero elements in matrix to indicate there is white pixel present on the scene. White pixels present indicate that there is motion on the scene.

$$\sum_{I} mtx(I) \neq 0$$

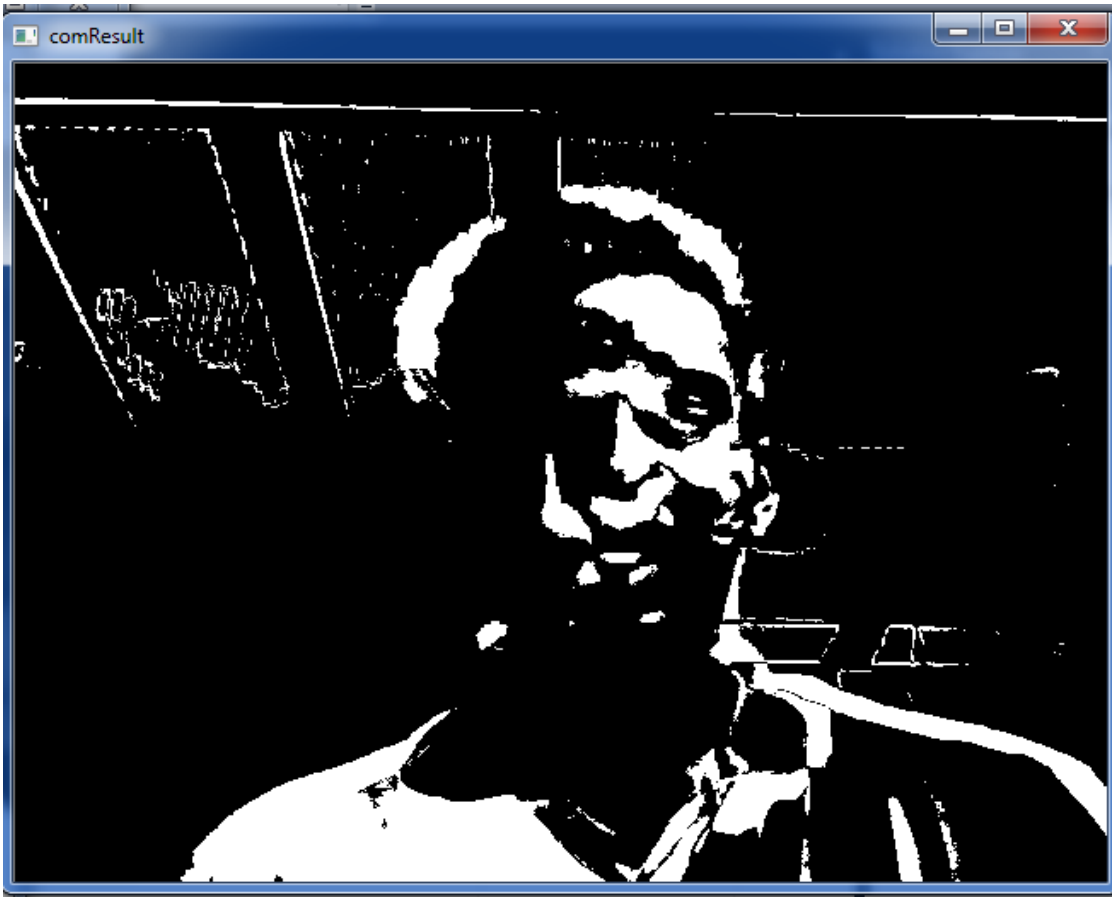


Figure 3.9: Image shows the result of background subtraction without noise filtering

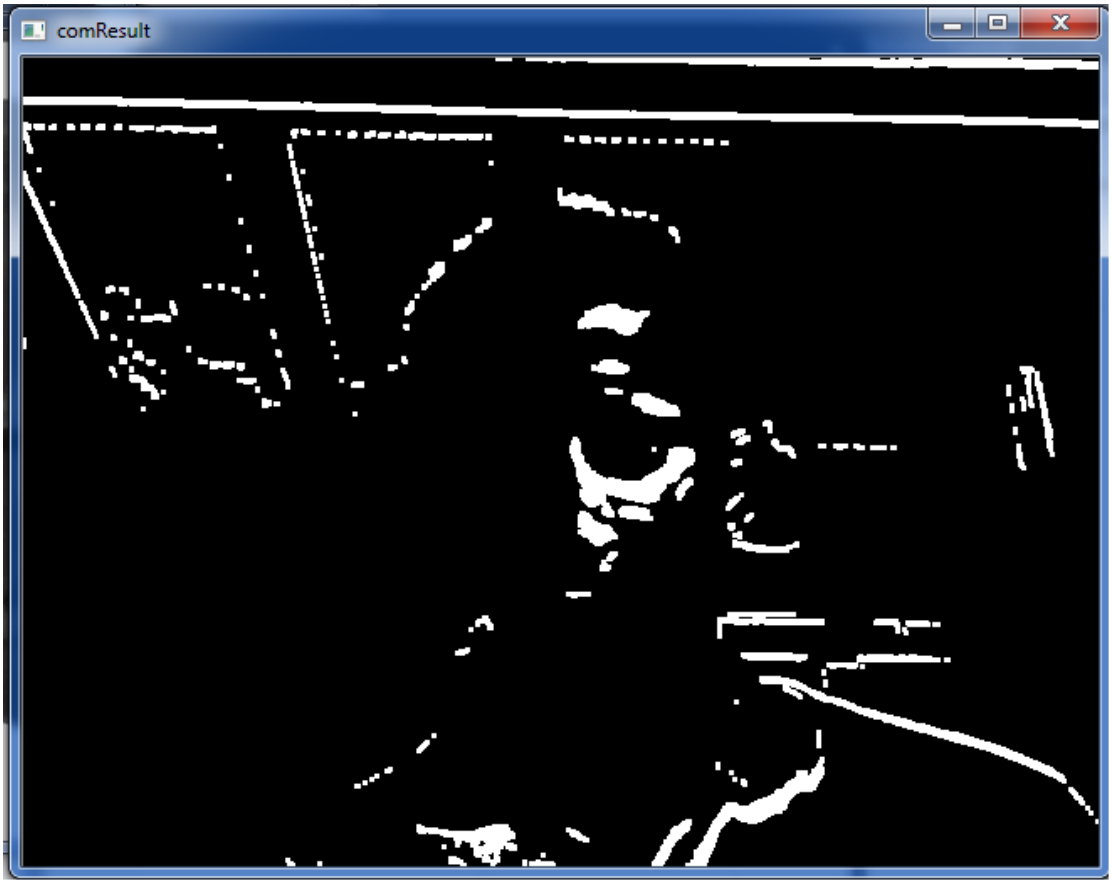


Figure 3.10: Image shows figure 3.8 after filtering

3.2.7 Alarm and Recording

Whenever an unauthorized entry or significant motion is detected, in motion detection decision making, the system automatically trigger to recording this motion at fixed location in memory that operator can accesses and alarm also is setting to this reason.

The function cvCreateVideoWriter creates the video writer structure.

In response cvReleaseVideoWriter function finishes writing to the video file and releases the structure.

Chapter 4

RESULTS AND DISCUSSIONS

This chapter shows the results and discussions of algorithm tested with the environment and circumstances to evaluate its performance and limitations. The major focus here is to achieve an algorithm which is robust and intelligent to detect objects motion.

4.1 Results from Motion Detection Algorithm

The vision system developed in the project works well in the indoor scene. The motion detection algorithm used to indicate the existence of motion in the scene and then record this motion.

The tab (Motion Detection) is clicked to show result of motion detection algorithm. Next figures show the results:

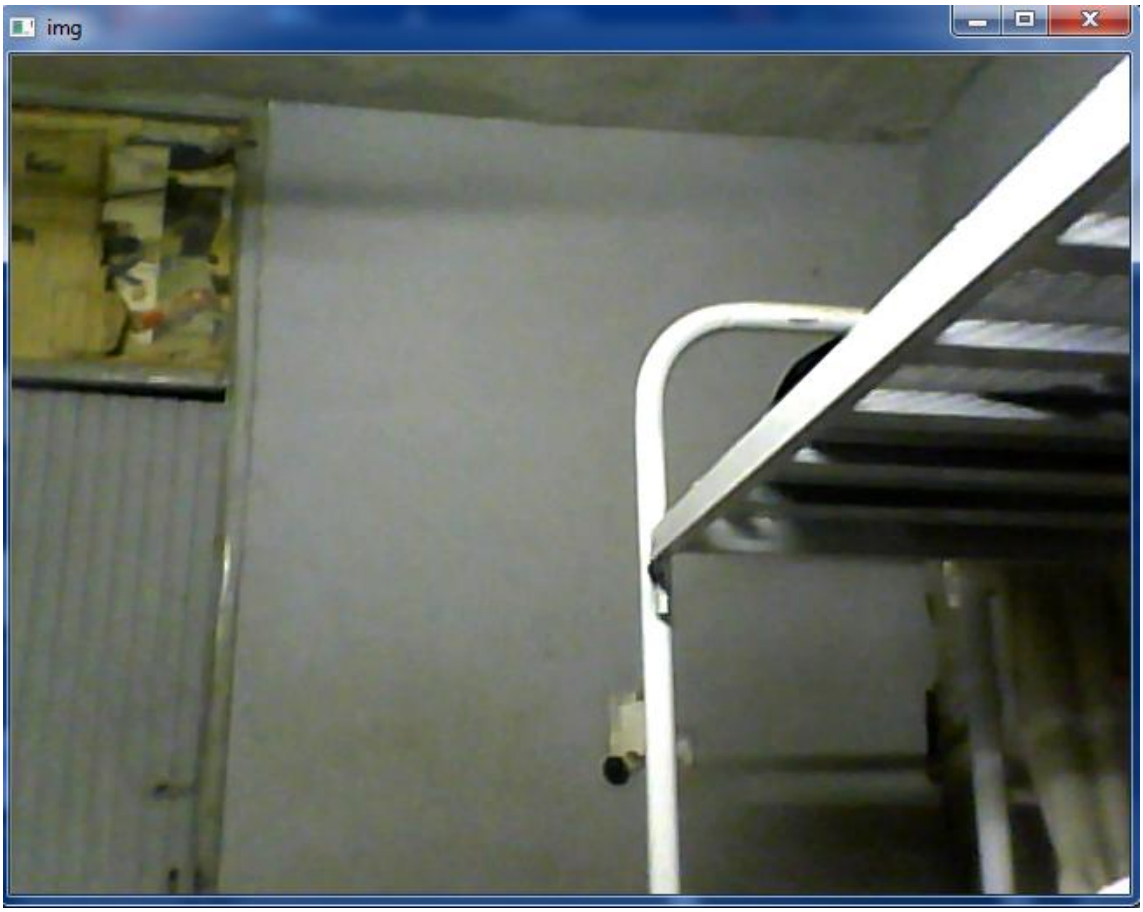


Figure 4.1: Colour image show the initial frame taken by the camera, i.e. background model

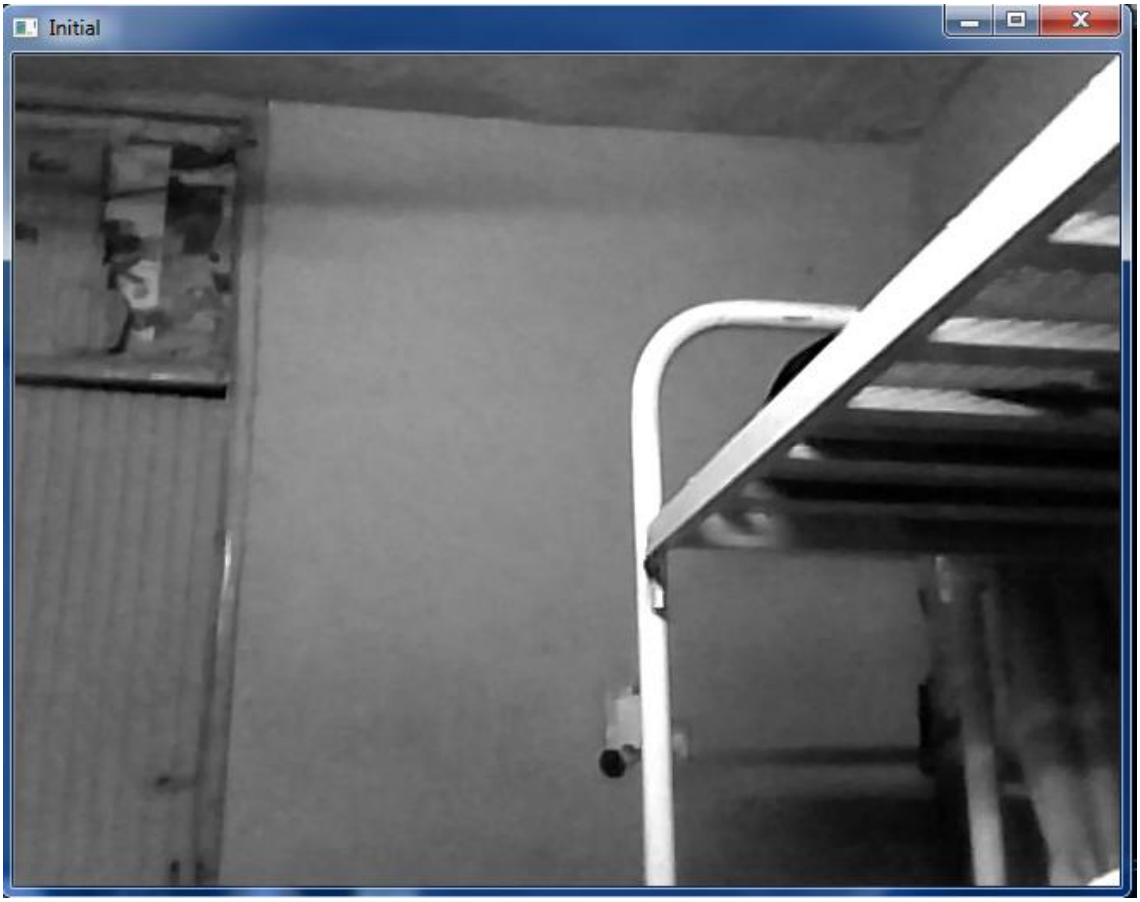


Figure 4.2: Gray-scale image show the initial frame taken by the camera, i.e. background model

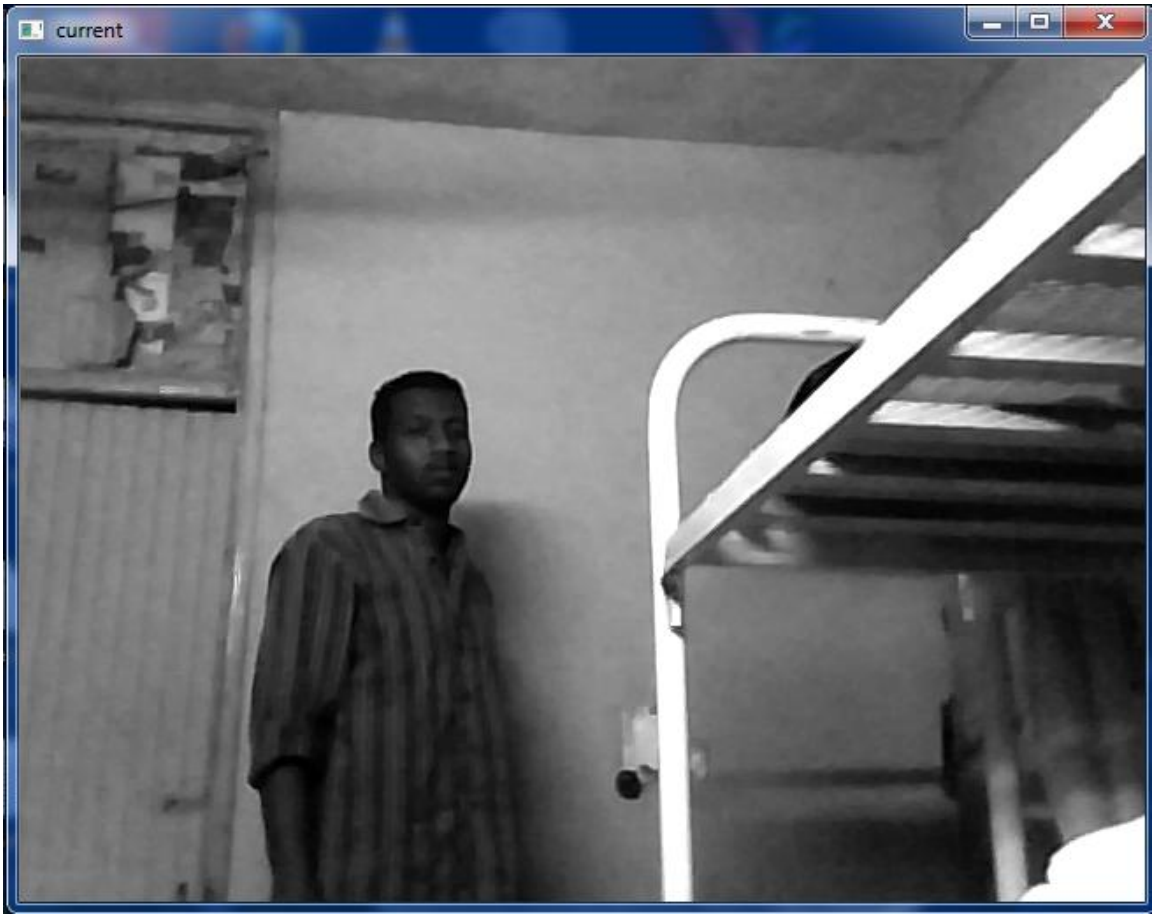


Figure 4.3: Gray-scale video (stream) shows the current frames taken by the camera

As mention before according to value of threshold the image in *figure 4.3* converted to binary image (white and black pixels) as shown in *figure 4.4*.

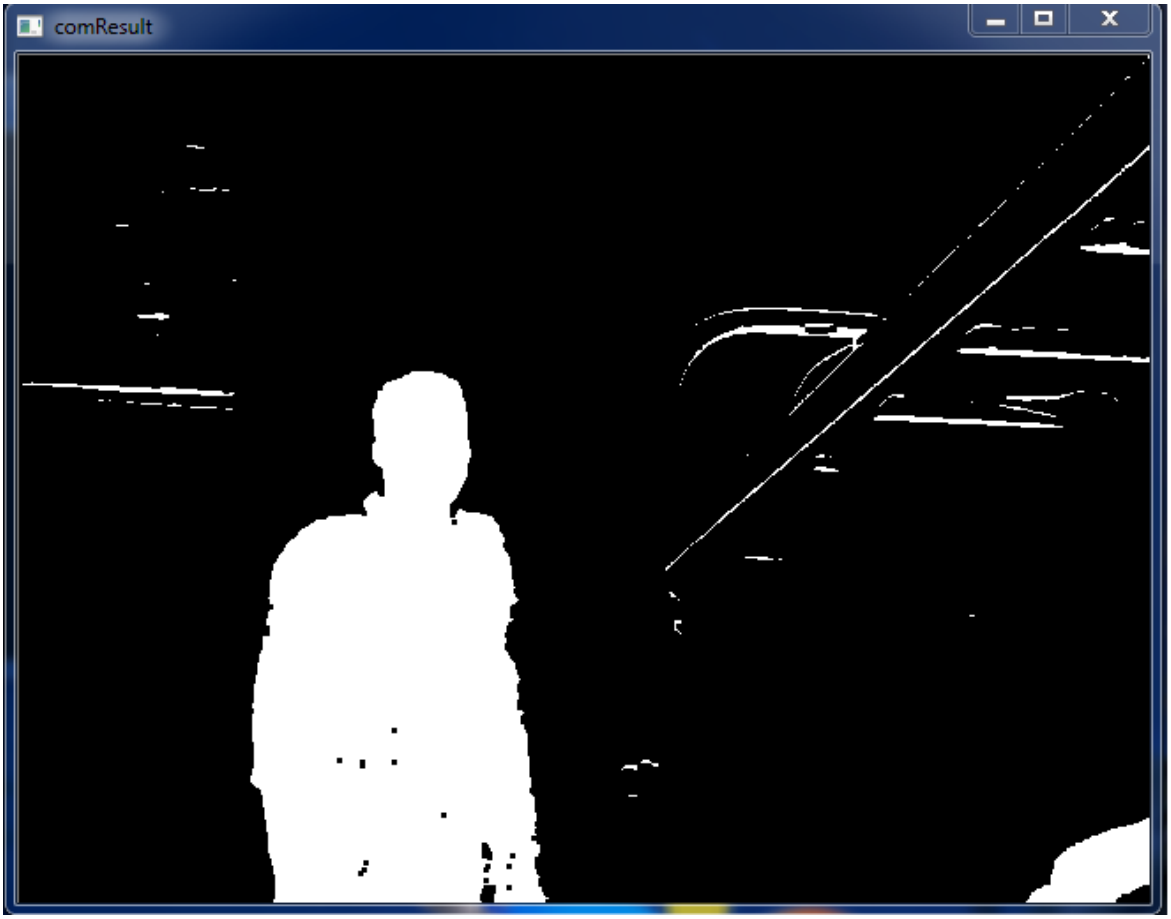


Figure 4.4: Binary image shows the result of background subtraction process

If the number of white pixels (foreground) in binary image reach a certain value (or greater than it) the camera start to record a video (motion detection case), otherwise no recording happen.

4.2 Discussion of the Results

The last results of motion detection algorithm chosen for this project is considered to be more robust and reliable compared to frame difference algorithm for static background model.

However, the achieved finding met the objectives of the project.

4.3 System validation

After the vision system has been tested in several sets of environments, the performance and limitation of the system can be determined. The system basically is limited to the limitation as follows:

1. Sudden illumination changes
2. Unable to detect minor motion
3. Unable to detect motion at far distance

4.3.1 Sudden Illumination Changes

Sudden changes in the illumination due to turning lights off in the indoor environment or due to sun being covered by clouds in outdoor environment make the brightness of the scene change hugely. This make the vision system failed to detect the moving object accurately. *Figure 4.5* shows the effect of sudden illumination changes where there is no motion occur.

Figure 4.5: shows the effect of sudden illumination changes where there is no motion occur

4.3.2 Unable to Detect Minor Motion

The system unable to detect the minor motion, due to the noise filtering algorithm had filtered out the small motion. *Figure 5.6* shows that the significant motion is interested only in the program.

Figure 4.6: shows that the significant motion is interested only in the program

4.3.3 Unable to detect motion at far distances

As mention before Number of white pixels in binary image (foreground) is used to determine if the motion happen. The algorithm is limited to far moving objects from camera because the object detection does not produce much significant information.

The resolution of 320x240 of the IP camera is further reduce during the image resize process which use for faster processing time.

The resolution effects on number of foreground pixels can be detected by camera. (I.e. number of pixels for moving object is less than the number of pixels for

same object but at closer distance to the camera), therefore detection at far distance can't handle because the algorithm treat it as minor motion.

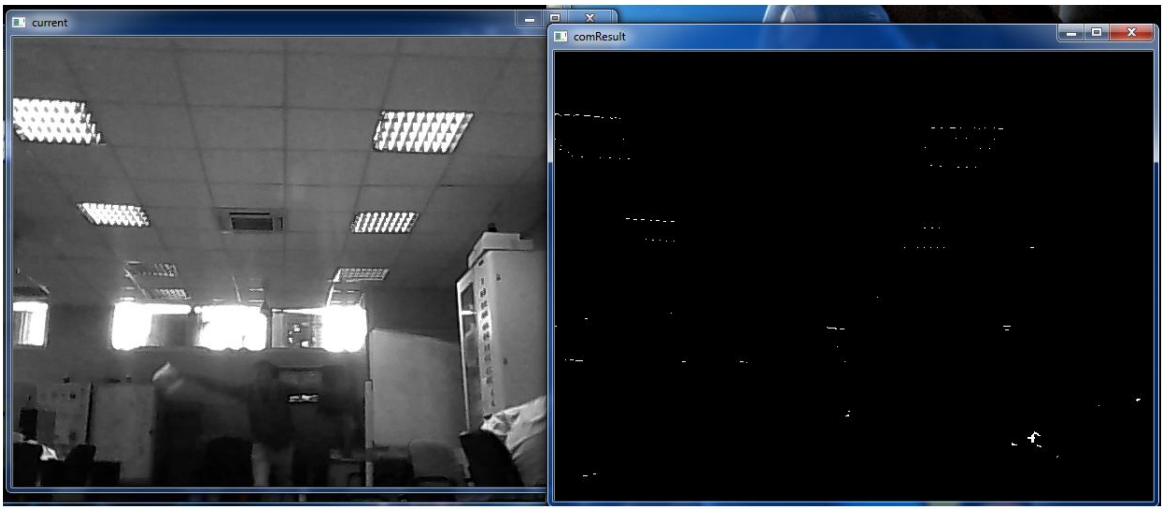


Figure 4.7: Image shows the failure of detecting objects at far distance

Chapter 5

Summary, Conclusion, and Recommendations

5.1 Summary

In this project, we presented a novel background subtraction algorithm for detecting moving objects from a static background scene. The method is shown to be very accurate, robust, reliable and efficiently computed. Experimental results and real-time applications were also shown. This method was designed under an assumption that the background scene is static, thus it may suffer from dynamic scene change such as an extraneous event in which there are new objects deposited into the scene and become part of the background scene. However, this problem can be coped with by allowing the system to adaptively update the background.

5.2 Conclusion

The project implements motion detection to successfully develop a video monitoring and detection system. The system mainly provides an efficient method for surveillance purpose and is aimed to be highly beneficial.

Overall, the project successfully fulfills all the proposed objectives to develop a vision based system to motion detection.

5.3 Obstacles of the research project

The first task of this project is to conceptualize the theory and concepts of image processing. Although is time consuming and difficult to understand, but is rather attractive and interesting.

The major problem of the project is to activate the video acquisition from IP camera which took considerable time to be realized. After some evaluation and considerations, the rtsp protocol used to handle the video acquisition from the IP camera.

5.4 Recommendations

We believe that, no perfect system exists. Background subtraction in itself is applications oriented. Background subtraction in itself is only a preprocessing, absolutely not the ultimate task. A perfect system should solve many problems, such as bootstrapping, moving objects, gradually and suddenly change of illumination, and tree waving, and so on. But some of these cannot be solved very well simultaneously because differentiating of them needs semantic understanding of motion of foreground and of background, and it is impossible if you have no information from the ultimate purpose.

Furthermore, in a particular application, not all the problems will be encountered. A good system should use the knowledge derived from its purpose as possible as enough to solve the problems encountered. No perfect system exists, but a good framework will give background modeling and subtraction much help. When there is an expected semantic interpretation of the foreground pixels in a given application (e.g. high-risk areas like banks), the algorithm designer should incorporate higher-level constraints that reflect the ability of a given algorithm to detect the most important changes, instead of treating every pixel equally.

The IP camera selected for project had low resolution; in order to enhancement the result from system, the IP camera resolution must be good enough.

References

[1] Intel® Open Source Computer Vision Library Reference Manual.

[2] Shireen Y. Elhabian, Khalid M. El-Sayed and Sumaya H. Ahmed. Moving Object Detection in Spatial Domain using Background Removal Techniques. Cairo University. July 2000.

[3] Ang Kah Wei. Human Motion Detection Using Image Processing Technique. University Technology Malaysia. May 2009.

[4] Jonathan Glumac. Motion Detection and Object Tracking with Interactive Surface. Worcester Polytechnic Institute. April, 2009.

[5] Thanarat Horprasert, David Harwood, and Larry S. Davis. A Robust Background Subtraction and Shadow Detection. University of Maryland.

[6] Mech R, Wollborn M. A noise robust method for 2D shape estimation of moving objects in video sequences considering a moving camera. Signal Process 1998; 66(2): 203-217.

[7] Azza Elgaili Abdelkarim. Network Surveillance System of Engineering Faculty, University Of Khartoum. July 2010.

[8] *Álvaro Bayona, Juan C. SanMiguel, José M. Martínez*. Stationary Foreground Detection Using Background Subtraction and Temporal Difference in Video Surveillance. Universidad Autónoma de Madrid, Spain. September 26-29, 2010.

[9] Javed O., Shafique K., Shah M. A hierarchical approach to robust background subtraction using color and gradient information. motion.*Workshop on Motion and Video Computing (MOTION' 02)*2002; 22-27.

[10] Nascimento J., Marques JS. Performance evaluation of object detection algorithms for video surveillance. IEEE Transactions on Multimedia 2005.

[11] DeskvShare. Understanding video streaming. [Online] [Cited: Nov 30, 2009.] http://www.deskshare.com/Resources/articles/vc_StreamingMediaFormats.aspx.

Appendix A

Program Source Code

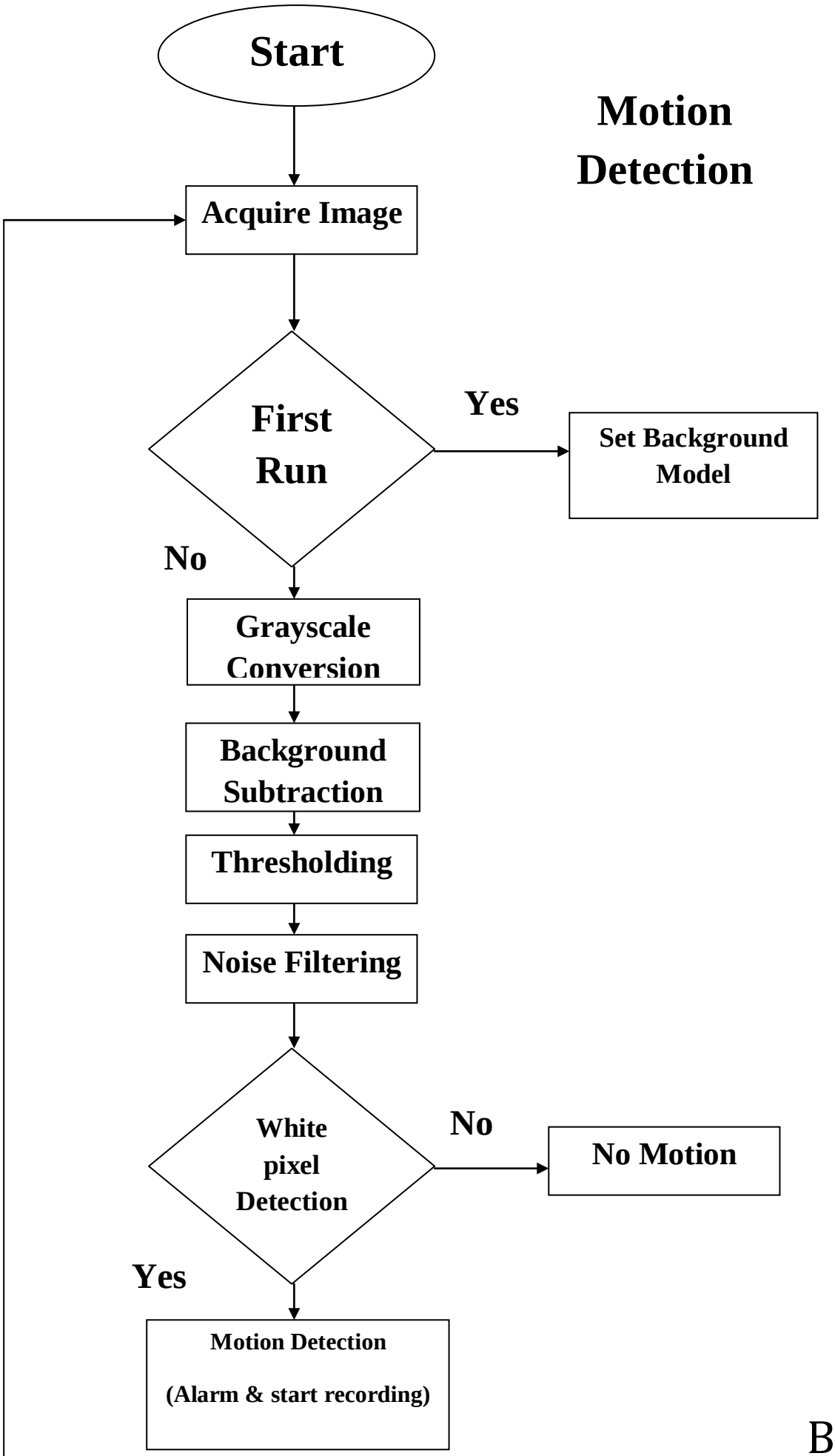
```
#include <cv.h>
#include <cxcore.h>
#include <highgui.h>
#include <cvaux.h>
#include <windows.h>
#include <iostream>
#include <stdio.h>

int main()
{
        //CvCapture* cam = cvCreateCameraCapture(0);     // For webcam access
        CvCapture* cam = cvCreateFileCapture("rtsp://192.168.0.77:554/medias1");
        IplImage* img;
        img = cvQueryFrame(cam);

        IplImage* current = cvCreateImage(cvGetSize(img),IPL_DEPTH_8U,1);
        IplImage* comResult = cvCreateImage(cvGetSize(img),IPL_DEPTH_8U,1);
```

```
        double cam_w = cvGetCaptureProperty(cam, CV_CAP_PROP_FRAME_WIDTH);

        double cam_h = cvGetCaptureProperty(cam, CV_CAP_PROP_FRAME_HEIGHT);

        double fps = 30;


CvVideoWriter* writer =
cvCreateVideoWriter("G:\\Project\\out.avi",0,fps,cvSize((int)cam_w, (int)cam_h),1);


          for (int i=0;i<30;i++)
            {

               img = cvQueryFrame(cam);

               cvWaitKey(20);
            }



        cvShowImage("img",img);

        IplImage* initial = cvCreateImage(cvGetSize(img),IPL_DEPTH_8U,1);

        cvConvertImage(img,initial, CV_BGR2GRAY);

        cvShowImage("Initial",initial);

                     while(1)

        {

               img = cvQueryFrame(cam);

               cvConvertImage(img,current, CV_BGR2GRAY);

               cvShowImage("current",current);


               cvAbsDiff(current,initial,comResult);

               cvThreshold ( comResult,comResult,50,255,CV_THRESH_BINARY);

               cvErode(comResult,comResult,0,1);

               cvDilate(comResult,comResult,0,1);

               cvShowImage("comResult",comResult);

               if (cvCountNonZero (comResult)>50000)


               {
                      Beep(1000,100);
```

```
                cvWriteFrame( writer, img );
        }

        if (cvWaitKey(30) == 27)

        {

                break;

        }
    }

    cvReleaseCapture(&cam);

    cvReleaseVideoWriter(&writer);
};
```

Appendix B

Flow Chart


Start
Motion Detection
Acquire Image
First Run
Yes
Set Background Model
No
Grayscale Conversion
Background Subtraction
Thresholding
Noise Filtering
White pixel Detection
No
No Motion
Yes
Motion Detection
(Alarm & start recording)