A Project

Motion Detection Sensor
(Study and Design)

By

Arkan Abdul Jabbar Saffer

2022 – 2023

CONTENTS

CONTENTS

LIST OF FIGURES

Abstract

The research project presented to you is to identify and study some of the basic human motion detection algorithms that have been established or developed previously. It will give a presentation of these algorithms to researchers to get a basic idea of the algorithm implementation for human motion detection systems.

The main algorithm discussed here is the one that implements the image subtraction methods and the foreground-background segmentation approaches. It aims to give readers the main idea of the human motion detection system architecture in applications. The report was written to document the design and development of a prototype human motion detection system.

We presented some basic ways to perform the human motion detection algorithm where the attempt was studied for the experiments conducted to evaluate the performance of the prototype system and to record their results.

Chapter One

Introduction

1.1 Introduction

An active electronic motion detector contains an optical, microwave, or acoustic sensor, as well as a transmitter. However, a passive contains only a sensor and only senses a signature from the moving object via emission or reflection. Changes in the optical, microwave, or acoustic field in the device's proximity are interpreted by the electronics based on one of several technologies. Most low-cost motion detectors can detect motion at distances of about 15 feet (4.6 m). Specialized systems are more expensive but have either increased sensitivity or much longer ranges. Tomographic motion detection systems can cover much larger areas because the radio waves it senses are at frequencies that penetrate most walls and obstructions, and are detected in multiple locations.

Motion detectors have found wide use in commercial applications. One common application is activating automatic door openers in businesses and public buildings. Motion sensors are also widely used in lieu of a true occupancy sensor in activating street lights or indoor lights in walkways, such as lobbies and staircases. In such smart lighting systems, energy is conserved by only powering the lights for the duration of a timer, after which the person has presumably left the area. A motion detector may be among the sensors of a burglar alarm that is used to alert the homeowner or security service when it detects the motion of a possible intruder. Such a detector may also trigger a security camera to record the possible intrusion.

Various sizes and shapes are available in the market now. In previous ones, a continuous electricity supply was needed and they were fixed in one place other than rotation. But now, portables are also available at Get-Owl-Home. In small sizes with more features having a chargeable battery. You can put them anywhere you want as you don't need to fix them in any single place.

In this chapter, we would just briefly look into the introduction of the project requirements needed for it and its purpose and aim. Also, a simple development plan for the prototype system which was drafted out is being presented here in this chapter. An overview of the system initially planned to be developed is also being presented here.

1.2 Problem Statement

The system is called Human Motion Detection Sensor System as a model, the project focuses on how the unit detects video motion where we will do research on techniques and methodology for motion detection and how advanced is the technology we prefer to use in this project. And designs recorded this unit the bottom movement and pass it to the next unit that will be on the classification of objects where it classifies the human and non-human being. Thus, this project is to come up with a solution that effectively detects and records motion with one or more objects that move and cause motions to occur.

This project is to help new researchers learn and further research a topic of their interest, which in this case is the Human Motion Detection System.

1.3 Research questions

The question is within the human kinetic detection and sensing system, and this is within the sequence of images:-

1- How do we detect movement?

2- How do we track a moving object?

3- How do we using the Human Motion Detection System?

1.4 Contribute

Through the presented research, we can review how to study and understand the formula and method of the kinetic sensing system in general and the extent of its sensitivity and its applications within the human movement in particular. The utilization and algorithms used in the kinematic system.

1.5 Work environment

This field of research is clearly seen here as being very broad. Only some enhancement or addition needs to be added if the algorithms were to be ported into other applications such as those in control application areas.

Providing inputs to Virtual reality or Augmented reality applications, we need to have extra recognition of the notified human motion returned by the algorithms discussed here. The reason is simply because of cost factors since we only have one camera and for simplicity of the algorithm with the assumption of a stationary camera being used.

Chapter Two

Literature Review

2.1 Introduction for the Motion detection sensor

Definition:-

Detector, sensor, sensor, or sensor is a sensing tool that detects the physical surroundings, some of which measure temperature, some of which measure pressure, some that measure radiation, and some that measure electrons or protons. Where it converts the signals falling on it into electrical impulses that can be measured or counted by a device. This enables us to know the intensity of the influence. There are also types of it that can be linked to computers and through programming; it is possible to form a picture of the distribution of measurements, as is the case in magnetic resonance imaging, which detects tumors in humans [2].

There are many types of sensors according to use, including:

2.1.1 Infrared sensors

Infrared rays are electromagnetic rays that have the same basic properties as light such as reflection, scattering, and interference. It was discovered by the German scientist Frederick William Herschel in the year 1800 when he was able to analyze light into its basic colors through a glass prism, where he noticed an increase in temperature when moving from the violet color field to the red color field. An image or scene is nothing but the result of the emission of electromagnetic waves by surrounding objects and their reflection from them. But the human eye is unable to see all light waves, as the field of view is limited between (0.8-0.4) micrometers, while the infrared field is limited between (3-25) micrometers [4].

Ineffective infrared systems: The principle of their work is to detect weak radiation and amplify it more than 10,000 times, whether these radiations are coming from space or emanating from engines and living bodies. These devices are usually made in the form of scope or in the form of a sighting device in weapons to suit the tasks of observation, monitoring, or shooting, and these devices allow a view of up to 5000 meters. The best-infrared system is based on detecting the infrared rays emitted by the objects to be detected and distinguishing them from the rays emitted by the sun, moon, stars, or those emitted by infrared lamps and then amplifying them.

Active Infrared Systems: Active infrared systems generate these rays by means of ordinary lighting devices with suitable filters to eliminate the waves of light beams located in the visible spectrum and keep only the required invisible beams, which are within the infrared range, so as to illuminate targets and sites at night. It is used to convert physical quantities into a signal that can be read [5].

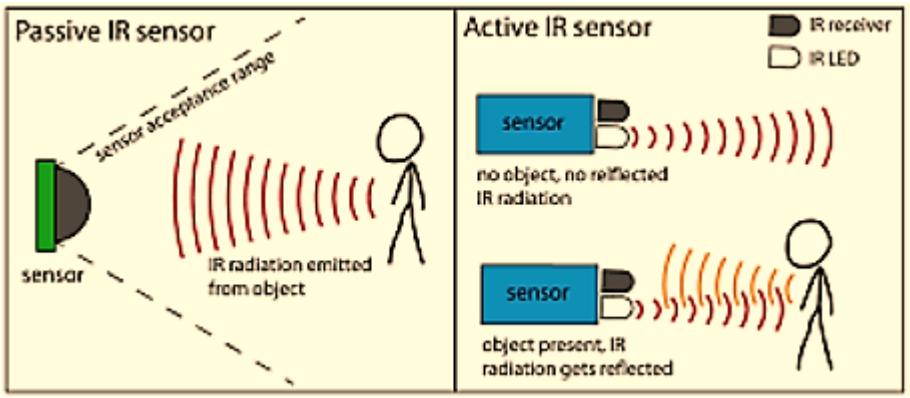

Figure 1: explain the Infrared sensors [5]

2.1.2 Acoustic sensors

Sensors that work on sensing sound and sound are the movement of molecules in general and air in particular. The sound sensor (microphone) is made of several materials, including carbon, which is compressed between two metal strips with an electric potential difference to create a small electric current that causes the vibration of one chip that leads to the movement of carbon, thus creating an electrical signal in the acoustic sensor wire (microphone) [3].

2.1.3 Optical sensors

Image sensors are used to form a digital image of a specific field; where these sensors are affected by the photons falling on them. These photons generate charges where they fall, and these charges are then detected to infer the photons. These sensors contain two layers of imperfect semiconductors:

- **One is of type (P).**
- **Another type (N).**

When photons fall on the semiconductor plate, they cause some of the electrons they hit to be released if they have more than or equal to the energy of removal; Where the removed electron leaves behind a positive charge.

Using some of the properties of imperfect semiconductors, we can make the fall of the photons cause the formation of a charge that can be detected so that we can infer the photons falling.

Falling photons generate charges, and the charge is obviously proportional to the number of photons that fall. If thousands of previous diodes are placed close to each other in small sizes, we will obtain information about light in close points that appear to the human eye to be continuous, but to form the image requires information about colors and not only about the amount of light. Therefore, each color is sensed separately at each pixel, where each sensor is assigned a specific color.

It takes at least four sensors to get enough information on each Pixel. The sensors in the above distribution sense colors (red, green, blue), which is the most common partitioning system where the sum of the colors is white.

Several methods are used to filter colors, including differences in wavelengths between colors, where a single sensor is capable of stimulating a small range of wavelengths, so its charge is formed as a result of the fall of photons of one color. Another method can be used, which is to determine the colors that are allowed to fall on the sensor, by using a membrane that allows the passage of only a certain color (Bayer layer). It is the most popular method due to its low cost and ease of manufacture. Using one of the two methods, the charge formed is due to a specific color, which helps in calculating the colors formed for each point in the image after the three-color components [12].

2.1.4 Inductive sensors

Inductive sensors are used to sense metallic objects and are also commonly used in industrial machine tools.

The inductive sensor consists of four basic components:-

Magnetic flux generator (inductive field generator): to generate magnetic flux, which are two coils as the principle of the transformer? Oscillator circuit: It is a resonant circuit that generates radiofrequency waves to prepare the signal to enter the amplifier (being small). Signal amplifier (Trigger): Amplifies the weak signal from the oscillator. Output: It is a tool to show the status.

Inductive sensors work according to the principle of electromagnetic induction; where the sensor coil consists of two coils. When the body approaches the sensor, the magnetic flux is transmitted from the first coil to the second coil through the conductor, which is the core, i.e., similar to the connection between the primary and secondary coils of a transformer. As a result of this transition, a potential difference is generated at both ends of the second coil, and the signal enters the oscillator circuit for initialization, and then into the amplifier circuit, which amplifies the final output signal. Sensors respond to objects only when they are within certain distances and pass in front of the sensor surface [7].

2.1.5 Tactile sensors:

It is a pointing device that has a special surface that can translate the movement and position of the fingers of the hand into a relative movement that appears on the screen, and it is one of the main features in portable computers, where it replaced the mouse; Its area rarely exceeds 40 cm2. It was first invented by George-E gerpheide in 1988 and the first to take a patent was Apple and used it in its Apple Powerbook computer in 1994.

The sensitive panel consists of different layers: the upper layer is the panel that is touched by hand, and underneath it, there are several layers separated from the other by an insulating layer. Transverse and vertical conductors charged with a constant alternating current.

Touch panel sensors depend on a phenomenon or property (electrical capacitance), and the phenomenon can be summarized by the occurrence of an electric field effect between the two electrical conductors when they come close to each other without contact between them. The field effects interact with each other to form a total electric capacitance that stores charges on the two opposite conductor surfaces. The surface of the touch panel sensors consists of an array of electrodes covered with an insulating protection layer [6].

2.2 A Motion Detection Algorithm

The most popular way to implement motion detection algorithm is by implementing the image segmentation and image subtraction techniques of computer vision [8]. Most of

the algorithm first segments the foreground moving objects from the background image. To do this, they would have to take a sequence of images with no motion by default to initialize the background image. This background image would be updated subsequently to provide a real-time environment where changes to the background are taken into considerations. For example, when a moving object stopped, it would then be a part of the background image. Clearly, the foreground objects can be acquired by using simple arithmetic using image subtraction. The result of the subtraction techniques where pixels belonging to the current image are subtracted by the corresponding pixels in the background image or vice versa would obtain the foreground moving objects. After obtaining the foreground objects, the focus for region of interest is set to these foreground objects instead of the whole image. Therefore further image processing is performed only on these regions of interest. One of the important steps here is to extract important features of the moving objects to recognize the object. However, recognizing the object is beyond this research topic as this module of the project is to provide the objects and the inputs to the object classification engine [8].

2.3 Image Segmentation & Image Subtraction

As mentioned, the first step of a motion detection algorithm would be the foreground-background segmentation step. To segment out the foreground from background there's a lot of techniques available. Ming and Tim [5] presented a way to improve the usual image subtraction techniques which only uses the pixels intensity values by adding also the colour illumination invariance into the algorithm. As a result, their work can also be implemented effectively on motion detection areas where the illumination changes rapidly. As said above, a motion detection algorithm begins with the segmentation part where foreground or moving objects are segmented from the background. The simplest way to implement this is to take an image as background and take the frames obtained at the time *t*, denoted by I(*t*) to compare with the background image denoted by B. Here using simple arithmetic calculations, we can segment out the objects simply by using image subtraction technique of computer vision meaning for each pixels in I(*t*), take the pixel value denoted by P[I(*t*)] and subtract it with the corresponding pixels at the same

position on the background image denoted as P[B]. In mathematical equation it is written as;

$$P[F(t)]=P[I(t)] - P[B]\ldots\ldots\ldots\ldots\ldots (1)$$

Where F refers to the foreground or resulted image after the computation at time *t*. The consideration of this most basic algorithm is based on a static background meaning no changes are made to the background. The background is assumed to be static and never changed even a little bit. The illumination of light and other changes which normally would take place as changes to the background is totally omitted in this algorithm. Thus, a better defined algorithm should be implemented to improve this problem and take these considerations into account. In image subtraction, the term subtraction gives meaning of given two images A and B, either A subtracts B or B subtracts A which may lead to a non-desirable result. Thus, usually we used an absolute subtraction technique where the result of A subtracts B and B subtracts A are basically the same. Ramesh, Rangachar and Brian [7] identify the image subtraction process by calling it difference pictures. The considerations discussed were identified by them as positive difference pictures, negative difference pictures and absolute difference pictures respectively. An important technique was introduced to segment objects using motion. This represents a technique which combines the spatial and temporal Gradients provided by Ramesh, Rangachar and Brian [7]. Also an accumulative difference picture technique was used to give more robust change detection.

2.4 Moving Edge Detection

Implementing edge detectors helps in obtaining a clearer picture of the moving objects. Ramesh, Rangachar and Brian [7] & Milan, Vaclav, Roger [16] both stated in their text that using edge detectors combining with the different pictures will give a better result for motion detection by helping to overcome several limitations. There are many edge detectors algorithms being introduced in the image processing field. Mostly all text on image processing does cover the topics on edge detection. Examples of these image processing texts are [7], [8], [16] & [17]. There are also many online references such as [6] which describe convolution techniques which may be applied to edge detection algorithms as well.

Ramesh, Rangachar and Brian [7] identified 3 steps in edge detection algorithms which are:

- ❖ Filtering
- ❖ Enhancement
- ❖ Detection

"Many edge detectors have been developed in the last two decades." quoted from [7]. Some first derivatives operator masks used for edge detectors are Roberts operator, Sobel operator, Prewitt operator. Others use second derivatives operator masks such as Laplacian operator. Some uses Gaussian edge detection methods, a popular known method which uses it is the Canny edge detector.

2.5 Motion Detection Approaches

There are many approaches taken by researchers to perform motion detection. Here we would discuss two among them, the first being the conventional background modeling method which updates the background based on statistical data approach which will be discussed in 2.5.1. The second would be a method identified by IBM research group consists of Ying-Li Tian and Arun Hampapur [13] which will be discussed in 2.5.2

2.5.1 Background modeling

Background does not always stay the same. Thus to improve the most basic algorithm discussed in the previous topic, we make B into a 1D space. For example, let t_2 denotes the time that a background is updated. An updating function denoted by Fn[B(t_2)] refers to the function to update the background image B at time t_2.

Therefore, after performing the initial calculation, the foreground is now extracted from this new background denoted by B(t_2). Thus changing the equation *(1)* into;

$$P[F(t)]=P[I(t)] - P[B(t_2)] \quad \text{..... (2)}$$

Where $B(t_2)=Fn[B]$ meaning $B(t_2)$ is calculated by an updating function that is performed on the previous background. One way is by using data collected from the frames and performs some calculation on the background image. One example is to take or record down all the pixels of all the frames before time t_2.

After that, simply sums them up and calculate the average value and update the background B to get a new background B(t_2) with this average value. However, there have been many ways to update the background.

Ramesh, Rangachar and Brian [7]'s accumulative difference picture is another way to perform background updating algorithms. The advantage of their method is that there function such as Fn[B] discussed above can be done in every frames as they use an accumulative function to determine the result of the subtraction.

Milan, Vaclav, Roger [16] also presented a similar accumulation different picture technique denoted as “cumulative different image” in their text. Ming and Tim [5] used a Gaussian mixture model based on the RGB colour space for maintaining a background for motion detection. Their work is effective when applied to outdoor motion detection systems where it is fast in response to illumination changes for example illumination changes arising from moving clouds. Moore [1] also implemented a Gaussian model for maintaining a background for motion detection. According to his writings, the Gaussian models update can be based on either the K-Means or Expectation Maximization (EM) algorithms.

In Ming and Tim [5] paper, they chose to implement the EM algorithm. After the foreground objects have been segmented, they are separated or identified using some algorithm to detect these regions. One of the ways is by using the region identification algorithm [7], [8] of computer vision where each separated objects are labeled differently to be able to distinguish them. Of course, some image pre-processing [7], [8] would have to be applied before the labelling process is done. Either a threshold function can be used to obtain a binary image to be labelled or edge detection with the convolution [6] operators can be chosen to find these objects in the foreground image and separate them. Some noise filtering functions may also be used to remove noise from the foreground images before proceeding to obtaining a binary image and labelling the objects.

2.5.2 Salient Robust Motion Detection

The claim of Ying-Li Tian and Arun Hampapur [13] is that most motion detection approaches requires hundreds of images without moving objects to learn a background model. Thus, the accumulative difference pictures as suggested by Ramesh, Rangachar and Brian [7] would be an exception. The three major drawbacks of adaptive background subtraction that were identified were:-

- ❖ It makes no allowances for stationary objects in the scene that start to move.
- ❖ It needs hundreds of images to learn the background model.
- ❖ It cannot handle quick image variations and large distracting motion.

Ying-Li Tian and Arun Hampapur [13] had identified previously developed salient motion detection methods by other researchers that were similar to theirs. In their method, they combine temporal difference imaging and temporal filtered optical flow. Their assumption was that the object with salient motion moves approximately consistent in a direction within a given time period. The calculation presented for the temporal difference is somewhat similar to the accumulative difference pictures‟ calculation presented by Ramesh, Rangachar and Brian [7].

They implemented the Lucas and Kanade method for calculating the optical flows. A temporal filter is then applied to the results obtained from the algorithm of optical flows calculations. Combining the y-component and x- component together with the temporal difference image, they obtain the salient object.

- ❖ An object that moves in different direction such as moving in a zigzag cannot be detected because their assumption of the object moving in a consistent direction was contradicted.
- ❖ If the object stops a while, the algorithm would lose track of it, however if it begins to move again, it can be redetected.

2.6 Feature Extraction

After having all the moving objects labelled and segmented, the next step is to understand what the object is or in other words, to recognize them. In some cases, we may want to classify them accordingly. However, the classification engine is dependent on the application where the human motion detection system is applied at.

Here we would describe one of the ways to provide inputs into the classification engine. Many features can be used as inputs into the classification engine depending on how we would like to distinguish the objects that are moving in the area of interest.

In this research in particular, we would like to distinguish human from other moving objects. To achieve this, Song, Goncalves, Feng and Perona [2],[3] have used a probabilistic structure to model a human body.

Using point tracking algorithm, they can pose estimation on the human motion to distinguish them from other moving objects. Their work refers to Johansson‟s [9] stimuli where each joint of the body is shown as a moving dot which is the basis of the point tracking algorithm. Moore [1] stated in his writings that there are many methods of tracking the moving dots or points. According to him, the Kanade-Lucas-Tomasi algorithm stands out for its simplicity but however suffers several flaws. Thus, Moore [1] had implemented a derivation technique based on the work of Birchfield [10]. An improvement work had also been carried out by Zivkovic and van der Heijden [11] where they suggested a way to improve on the standard Kanade-Lucas-Tomasi algorithm. In their papers, they focus on the initial selection of the features points and suggest a method to improve the selection process to reduce detection errors.

Thus, to perform a point tracking algorithm, a motion detection algorithm would have to first initialize the feature points on the segmented foreground objects

track their movements using either the Kanade-Lucas-Tomasi algorithm [11] or using a derivation of the Kanade-Lucas-Tomasi algorithm as researched by Birchfield [10]. After tracking the movements of these points, their position and velocity of their movements resulting from the tracking algorithm would be used to map them with the model structure of human body as presented by Song, Goncalves, Feng and Perona [2],[3]. Therefore, the probability of how close a moving object is a human being is returned. A simple classification can therefore be achieved using this point tracking algorithm.

Some research have also been carried out on tracking algorithms such as Andrew and Kentaro [15] had presented a tracking method using probabilistic metric space in their papers.

Their method has several valuable properties such that it provides alternatives to standard learning methods by allowing the use of metrics that are not embedded by vector space. It also uses a noise model that is learnt from training data. Lastly, it needs no assumption of probabilistic pixel wise independence.

Some works have also been carried out on recognizing human motion without using any feature extraction. Pallbo [12] have used a neural model to detect motions by using each pixel in an image as a node input in the neural network.

The recognition and classification part of the system is beyond the scope of work and the partner in this project would survey on that however. Thus, these are techniques that we had managed to come across with. There are however, many techniques available which are not discussed in this paper.

2.7 Motion Detection End Product Applications

Mike and Barry [14] have presented an example of motion detection end product applications configurations. They have identified three modes of configuration for their system. The first discussed was passive motion detection, where motion events and video are recorded into a device and at some point later reviewed by a Control Center. This is suitable for applications whereby no action needs to be taken as a result of a motion detection event but video is required before, during and after an event. Another type was where the Control Center operator Wishes to see what caused the motion detection event immediately.

The third type of configuration discussed was for scenarios where what happens after the event is important, and in this situation, the system is configured so that video is recorded only when a motion detection event had occurred. This reduces the required storage requirements. Motion detection and recorded videos can then be viewed at a later time.

There are also many other applications of human motion detection systems as suggested by Thomas and Erik [18]. Instead of security surveillance systems, it may also be applied in the control areas of application such as providing input to virtual reality engines or augmented reality engines. In those areas, gesture recognition would be an added value to the algorithm of human motion detection and tracking.

Thomas and Erik [18] have also suggested the analysis areas of application where motion detection is used for clinical studies or to help athletes understand their movements and improve their performance.

No matter what areas of application the motion detection is used for, the main architecture of a motion detection system is basically shown as below:

Figure 2: A block diagram of an end product motion detection application [18]

As shown above, an image or image sequence is captured and processed. Then, it would be applied to the motion detection algorithm which returns the moving objects that causes the motion events. Next, the gesture or pose of the moving objects is estimated and tracking is performed. Finally, the recognition is performed and output actions being triggered.

However, this research project only covers the motion detection algorithm combined with a simple recognition engine applied using an artificial neural network. To be specific, this part of the project is only focused on the motion detection algorithm. Thus, in the next chapter, we would give an overview of the project and the techniques and methodology we had implemented after the research had been finished.

Chapter Three

Solution Method

3.1 System Overview

This chapter will look into the design of the methods and techniques implemented in the final prototype system. Diagrams of the architecture of motion detection algorithms are being presented here in this chapter as well. Also, we will look at the development stages of the prototype system including the previous algorithm that was implemented and was discarded or replaced due to certain reasons.

The basic human motion detection would have an alarm system integrated; the development of the prototype system did not include the alarm interfaces. Since there's no hardware for the research team to test, the program would only go as far as recording the video where motion events occurred in a scene.

As shown, there are two outputs given by the motion detection module. This means that there are two algorithms being implemented which are namely spatial update and temporal update + edge detection whose output were respectively denoted in Figures 3 and 4 as algo1 and algo2. We would focus on describing the development of the motion detection algorithm section of the prototype system and also the image acquisition section.

Although the diagram has shown most of the processes for the prototype system, there is another important component that runs in a separated sequence to the processes. The background updating model is an important issue for the motion detection algorithm. Since we've implemented two distinct algorithms for the background updating module.

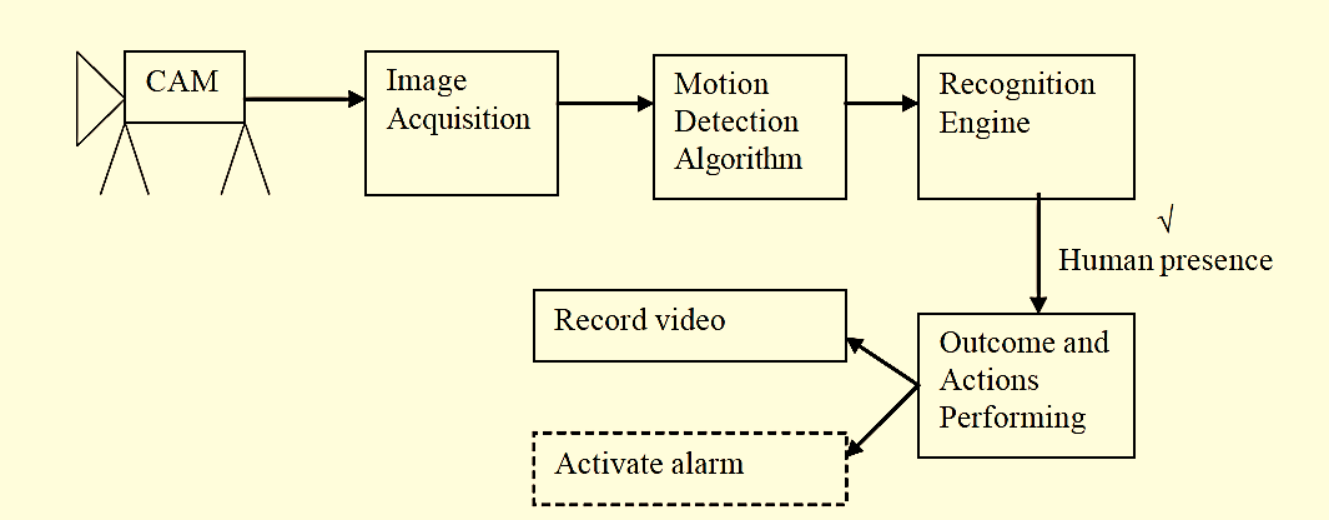

Figure 3: Overview of a motion detection system [18]

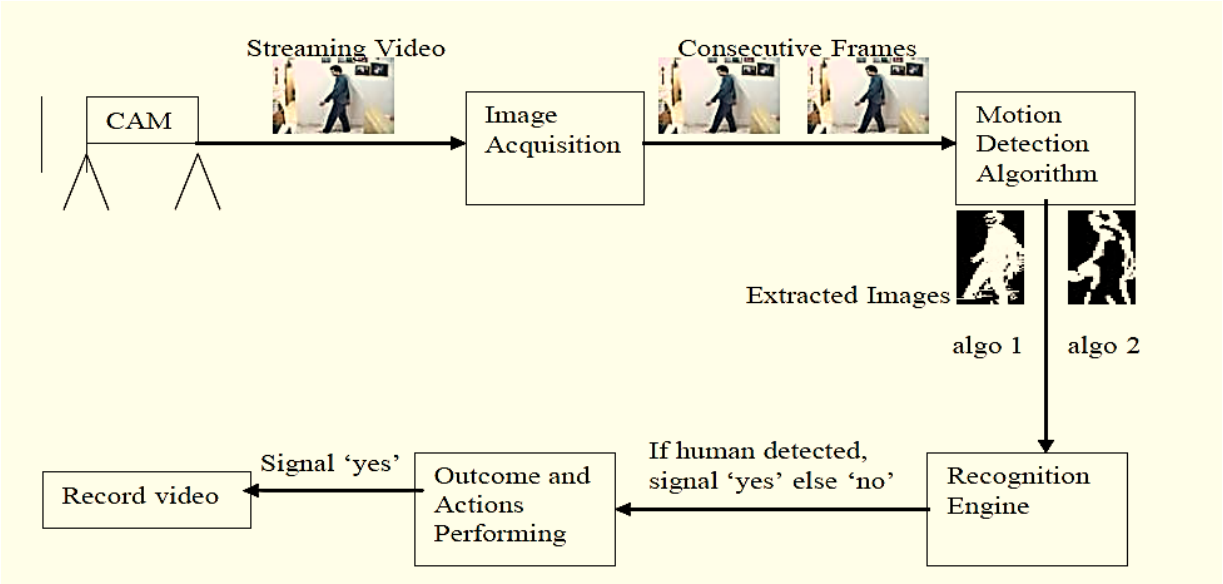

Figure 4: Overview of the prototype human motion detection system [18]

3.2 Image Acquisition

The system developed here can capture a sequence of images both from real-time video from a camera or from a recorded sequence in ".avi" format. This is mostly an initial step for most motion detection algorithms. Images of resolution 300x200 have been set to be captured. No matter what the settings of the camera's property, our prototype system will only use 300x200 resolutions for its calculations. Since we had used a camera capable of displaying more than 300x200 image resolutions then we assume that there should be no problem for other cameras with the same or even better specifications to run our prototype program.

The reason for this is because of the limitation by the rectangle drawing functions provided by Intel Open CV [19] as stated in chapter1. Also, some performance issues on speed were the cause of us developing this project in such resolution specification.

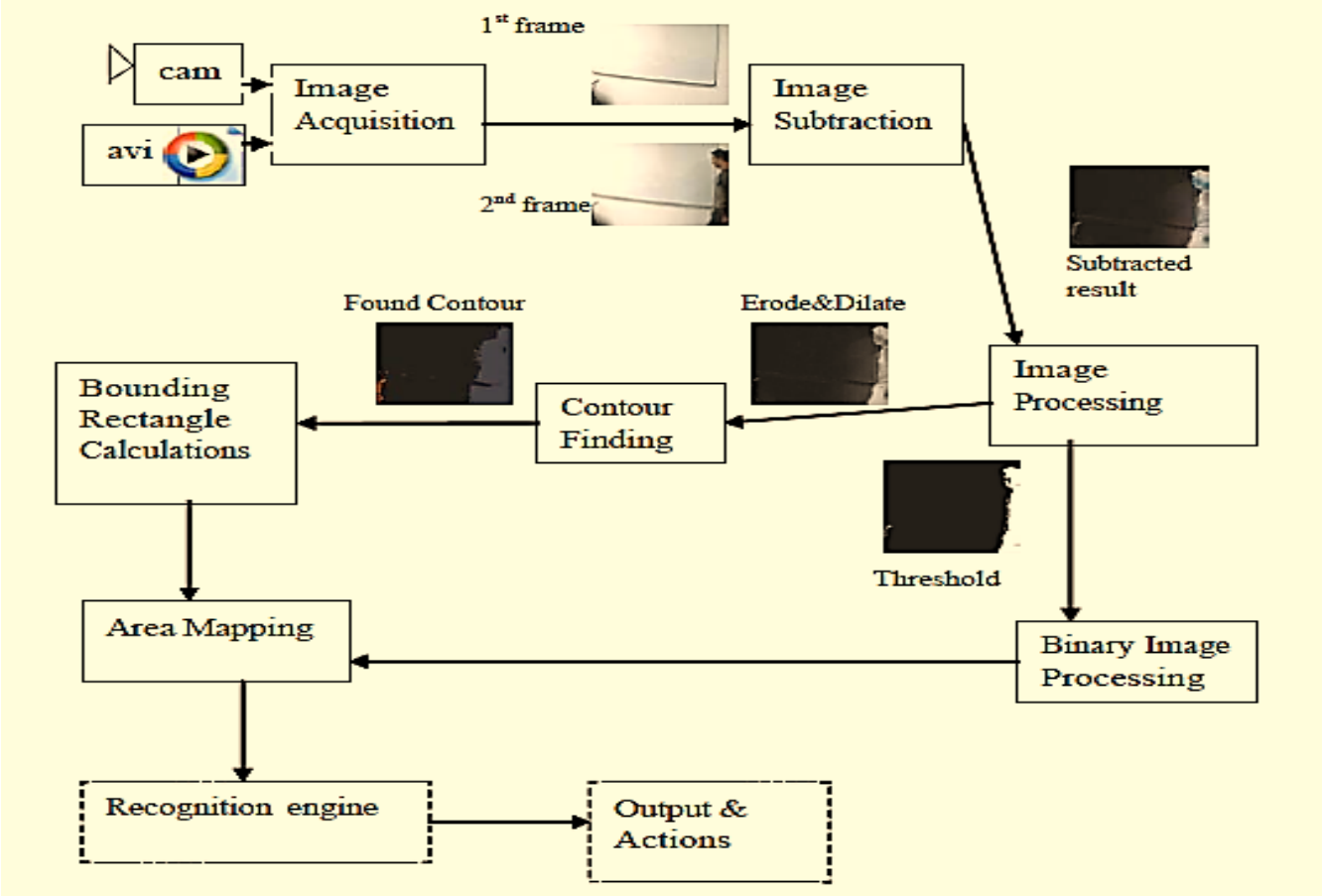

Figure 5: An overview of the motion detection algorithm [19]

3.2.1 Image Segmentation & Image Subtraction

☒ Image subtraction

Here two subsequent images are compared and processed using the arithmetic operation of subtraction to subtract the pixels‟ value in the images. We'd implemented an absolute subtraction technique thus there's no difference in choosing which image to subtract with which, since both would result in the same resulting image. Since usually color images are used, the algorithm implemented considers the color data of the images. The technique first separates the images into three planes or channels. Then it performs the arithmetic subtraction to each plane or channel. After that, the results are combined back again to form a color image. Thus, as shown in figure 5, the result of subtraction is in an inverted color format. The reason for the inversion is simply because subtraction of the pixel's values was performed.

☒ Image processing

The next step would be to perform some image processing operations on the result obtained from the previous step. The two branches coming out from this module as shown in figure 5 is because two images were produced from different image processing techniques for different purposes from this stage of the process. The first is the result obtained from a threshold function. This result is further used for recognition purposes as it filters the human body shapes better than the other output. An adaptive threshold function is implemented here. The method is known as the Otsu threshold. The other output from this stage is eroded and dilated image. This function simply removes a small piece of noise that may be caused by camera signal noise or small pixel changes. The function increases the filled section of an image and then decreases it leading to small portions being removed. The reason for using this instead of threshold for contour is because to draw the bounding box we want a bigger region so that we do not lose out on any body parts on the rectangle drawn. This is only implemented for the 2nd algorithm; the temporal update + edge detection background update algorithm. The reason is simply that the 2nd algorithm doesn't return enough information to be threshold by Otsu threshold function thus, instead of using Otsu threshold, dilate and erode is being implemented.

☒ Contour finding

The next step is performed on the eroded and dilated image for algorithm 2 and the threshold image for algorithm 1 to identify individual contours that are present in the image. These contours are displayed with different colors for contours considered as separated from one another. Those that share the same color may be considered connected to each other.

☒ Bounding Rectangle Calculations

Some operations are done in order to remove overlaying boxes or rectangles when drawn to the source image. Basically, rectangles or boxes that are near and almost cross

one another's edges would be joined up to form a larger rectangle or box. This would help eliminate the problem of only a portion of human being returned to be recognized.

☒ Binary Image Processing

On the other path of the process as shown in figure 5, the threshold images obtained are then further enhanced to fill in the empty spaces inside the binary region which represents the moving object. Basically, an algorithm that scans through the vertical lines and filling up each vertical line's first and last white pixels‟ region in between is implemented.

For example, if in vertical line x=1, let the first white pixel be in x=9, and the last being in x=13. Then this algorithm would fill up the pixels I(1,10), I(1,11), and I(1,12) where I represent the image's coordinates. Not only the vertical lines are performed but also the horizontal lines are performed with a similar task too. After we obtain both the resulted images from these two algorithms, we use the AND operator to combine both the result to form a better representation of the moving object's shape.

☒ Area Mapping

After the bounding rectangles are identified, the position is then mapped to the source image, and the rectangle being drawn there. Mapping is done for the area of the bounding boxes drawn in the source and also the corresponding area in the binary processed image. The area from these processed binary images is the one to be used for the recognition engines. This project does not cover the recognition engine and the output component.

3.3 Background Updating Algorithms

Here we've implemented two algorithms to update the background image which is to be used in the image subtraction stage of the motion detection algorithm is shown in figure 5. Now we discuss the two algorithms in the following section:

3.3.1 Spatial Update

The first algorithm implemented was designed in such a way where the background model is updated based on spatial data instead of temporal data. By temporal data here, we refer to the time or frame number of the images in the sequence. Thus, since this method uses only spatial data which means the background is updated based on some percentage of the pixel change in the subtraction result.

For example, let A be an image in a sequence or being the current frame processed and let B represent the previous background or the initial background image. After processing the image subtraction stage of the motion detection algorithm, the resulted subtraction image is then considered. The percentage of pixel change is then calculated based on the number of nonzero pixels belonging to the resulted image from the subtraction process. To be exact:-

$$\boldsymbol{Percentage} = \frac{\textbf{number of nonzero pixels in subtraction result image } (|\mathbf{A}-\mathbf{B}|)}{\textbf{total resolution of the images A or B or even the result image of } (|\mathbf{A}-\mathbf{B}|)}$$

Taking advantage of similar studies setting the minimum refresh process to 80% which means that the background will be set to a completely new image (the current frame image) without any calculations when the percentage value calculated by the above formula is greater than or equal to 80. With this algorithm, the increment of Gradual acceleration. The reason is simply that the calculation is done less often. There is no need to collect statistical data for temporal information. The assumption being made here is that the background will not change unless the camera is repositioned or reconfigured. Thus, giving spatial data that satisfies the pixel shift condition is above 80%. This method also helps eliminate the need for a tracking algorithm where any moving objects that were not in the initial background image will not be detected. However, it also has several drawbacks like giving more false alarms which cannot be removed and can only be removed by reformatting a new background image.

3.3.2 Temporal Update + Edge Detection

As for the second algorithm, we had implemented here a usual motion detection approach for updating the background image. Compared to the first algorithm, this method uses much more computation steps and thus giving slower response or performance. The information that is obtained for the computations are from subsequent frames and not only depend on a single subtraction result in fact it doesn't concern any subtraction process results. Basically, it uses a running average method written readily by Intel Open CV [19] group. The methods described as the accumulative difference are similar to the method that we present here for the second algorithm on the background update model.

Basically, all frames are used for computation here for a running average image to be produced. A basic running average method is defined as the sum of the previous value of a pixel at a certain location with the new value taken by the corresponding pixel at the next frame in the image sequence with a certain factor of degrading the old pixel value.

For example, let A represents an image in a sequence or being the current frame processed, and let B represents the next image or frame in the image sequence. The factor of degrading the old pixel is denoted as W. The computation of the running average image R is computed as:

$$\mathbf{R[x,y] = (1-W) \bullet A[x,y] + W \bullet B[x,y]}$$

As the computation continues, instead of using A[x,y] pixels values, the function will replace it with R[x,y] values to form a recurrence equation. And thus making the equation formed as:

$$\mathbf{R[x,y] = (1-W) \bullet R[x,y] + W \bullet B[x,y]}$$

Notice that x and y denote the location of the certain pixels we're referring to on the computation.

Thus all pixels belonging to the images are computed to give a resulting running average image of the same size which is used as the background image.

The background image is therefore formed after a certain period of time and is constantly updated based on the frequency set for that period of update.

Here we implemented a double-layered running average meaning we implemented the same formula on a number of resulting images from the running average formula to form a new result that is used for our background instead of directly using the resulting images to form our background.

Therefore, there are two timer macros needed that defines the frequency of updating the temporal data obtained from the first layer of the running average function that would be used as inputs to the second layer of the running average function and also to define the frequency of updating the background using the second layer function.

In algorithm 2, the moving edge concept is also implemented. The reason is simply that the result returned is completely different compared to the result obtained by algorithm 1. The perfect shape of a human obtained by algorithm 1 is not obtained here. Thus to help recognition of a human being detected, using edge would help a lot in distinguishing human objects and non-human ones.

Instead of using the AND operator to combine the edge obtained from the current frame and the result of the subtraction, the AND operator was not used in fact. The implemented moving edge technique only performs a Canny edge operator [6], [7], [8], [16] & [17] directly on the resulted image from the subtraction process.

3.4 Design how to implement a prototype using the Intel Open Resume Libraries

In building a prototype, it can be used to implement libraries MFC (Microsoft Foundation Classes) for some basic Windows functionality using some of the win32 APIs (application programming interfaces) provided by Microsoft. As for the aspect of image processing and its operations [19].As for the recognition engine, the partner in this project can use some Torch libraries to program the artificial neural network [20].

Chapter Four

Designing

4.1 Image Subtraction

As was previously mentioned, the implementation and design are within the approach of absolute subtraction of color.

As shown above, the process is separated into 3 parts, separating the channels, performing subtraction for each of them, and finally, combining back the results. The source codes provided here are edited to help make it readable. Functions such as get Pixels() and set pixels() needs to be further expanded in actual coding.

4.2 Image Processing

As described previously, this section has two distinct kinds of image processing techniques being implemented. The adaptive threshold function was performed using a function provided by the partner in this project since the purpose was to give a better result for the recognition engine and helps improve his feature extraction algorithm. To apply a threshold value returned by the function we can simply use the CV Threshold (some parameters) function provided by the Open CV library [19].

For the other processes, libraries function by Open CV can be implemented directly since they provided Dilate & Erode function in cv Dilate(some parameters) & cv Erode(some parameters) for algorithm 2. To get the actual parameters for these functions one may read the Open CV's documentation or manual provided once one installed the libraries. In the case of algorithm 2, functions such as CV Canny (some parameters) and CV Dilate (some parameters) are also being implemented for the moving edge concept described in the 2nd and 3rd chapters.

However, here the moving edges method being implemented is different from what was described in the 2nd chapter, thus, the function CV and (some parameters) were not implemented at all as described in the 3rd chapter. Thus, the CV Canny (some parameters) function is performed directly on the resulted image of the subtraction after it was being threshold. In this case, it uses the image as a result of the adaptive threshold method implemented instead of the dilated and eroded image.

4.3 Contour Finding

Similar to those performed in the previous stage, this section can be implemented by using functions provided by Open CV libraries [19] directly. Any way to show how it was implemented on the prototype system, the source codes are shown below:

As shown above, the source codes prepare data for the next stage by storing information of the rectangle in a vector which is implemented by the Standard Template Library provided by the C++ language include headers. This information is then used for the next stage where the rectangles are bounded and the overlaying ones are being removed. The following section describes the implementation of the calculations done to do that.

4.4 Bounding Rectangle Calculations

An algorithm was developed to remove overlaying rectangles caused by disconnected contours components.

Condition 1 and condition 2 shown above are two lengthy conditions for cross-over to happen between the overlaying bounding rectangles. One of the conditions is used to check whether the right side of one rectangle crosses over or is nearby to another rectangle's left side. The other condition is used to check if it is the case of the rectangles being on opposite sides, therefore the condition is almost the same except that it considers the rectangles on the opposite sides. Both the condition checks also the vertical cross-over that occurs. This algorithm helps reduces the problem posted by the 2nd algorithm, Temporal update + edge detection of the background updating model where only a portion of a human is moving. This algorithm helps return the correct bounding box in a more accurate manner in case of such events where not only the portion is returned but the whole object is returned. However, it does not guarantee to solve this problem 100%.

4.5 Binary Image Processing

We developed a way to better obtain a shape of a moving object from the threshold result of subtraction.

Notice that only for algorithm 1, Spatial Update of the background updating model is using this algorithm. The result obtained by algorithm 2, Temporal Update + Edge Detection is not returning a complete shape of the moving object instead it returns the edge only which is not suitable to use this algorithm since the edge does not bound the exact objects shape. Also this is not the whole story, a AND operator is used to combine the temp & temp src image data used in the algorithm above in order to get the shape of a moving object without noise. It is achieve thus by using cv And(some parameters) as described previously on implementing moving edge concept in algorithm 2.

4.6 Area Mapping

This section is basically implemented by using the cvSetImageROI(some parameters) and the cvResetImageROI(some parameters) functions as provided by Open CV [19].

The equation for the running average method used to replace the initially described formula in phase 1 report now is as described previously to be as follows: (applied directly from Open CV's documentation)

$$\mathbf{R(x,y)=(1-\alpha)\bullet R(x,y) + \alpha \bullet I(x,y)}$$

Where I is the current image frame, R being the running average of frame sequence and α (alpha) is the weight that regulates update speed (how fast the accumulator forgets about previous frames).

However, here, we used two running average images for updating the background model. Two timer's macros have been defined to set the frequency of how often or seldom these running average images being updated. In this algorithm, the timer macros are used as the time interval which defines the time period for the background to be updated periodically. Thus, the algorithm consists of dual-layered running average images.

Therefore, tried to combine both algorithms into one prototype so that only 1 executable needs to be presented.

Comparing our previous method of calculation to using the running average, we find there's not much difference. Thus, we had chosen to omit the previously developed prototype and only continue on the new prototype as the new one have both algorithms implemented on it.

Finally, the developed prototype had implemented both algorithms. The prototype is separated into two modules. The first is using direct capture of images from the camera. The latter accepts ".avi" files. When a source for the image sequence to be obtained is set, a dialog will be prompted. Details instruction is given on the options to be selected. Users are expected to click on two buttons, each button to prompt out 3 windows namely "Output", "BG" and "Segments". The two buttons are to initialize the algorithm whether to initialize algorithm 1 or algorithm 2. After the windows have been prompted out, users are instructed to press on the keyboard as responses. By pressing 1 – a real-time video with motions area marked will be displayed at the "Output" window. Respectively, pressing "2" for displaying only the moving regions. Also, pressing 3 for displaying the moving object's contour. The same button which started the algorithm is to be pressed to stop the algorithm.

Chapter Five

Summery and Conclusion

5.1 Description

This chapter provides a description of how the applied motion detection algorithm is tested for performance and what pre-earned results are returned in the actual implementation and also expected and theoretically assumed results that will be obtained given the resulting images that were to be passed to the recognition engine. And discuss the several advantages and drawbacks of the algorithms implemented. We will also look into future works that can be carried out to further research in this area or this topic.

5.2 Experiments Description

There are many approaches to test the performance of a motion detection system. Here in this chapter, we will present one of the ways to test the prototype system developed. Since in this report we focus on the motion detection module, we will present a method to test the performance of the motion detection algorithm as a start, after that, we will take into consideration the performance of the prototype implemented with the recognition engine as well.

The first would test the performance of how many motion events caused by humans are correctly detected with the algorithm and the latter would test the performance of how accurately these events are recognized as human.

To test the performance of how many motion events caused by humans are correctly detected, we count the number of frames that a rectangle is correctly drawn to return a region of interest where the whole body of the human is included in that region. We consider the ratio of this number that we counted with the total number of frames included in the sequence of images passed in which has a human causing the motion events.

This method however is only suitable to be used to test algorithm 2, since algorithm 1 would never lose track of the human even though they remain static or not moving in other words. Even though algorithm 1 guarantees the region of the rectangle drawn covers the human, however, it is very much dependent on the threshold result of the

subtracted image. Thus, we may use the same technique to test the performance as well. Therefore, for testing purposes, we had used a few video clips in the form of ".avi" file format system. We consider the effectiveness of the bounding rectangles in three forms, Full Bound-which means the whole human body must be wrapped, Sufficient Bound-which means the bounded area is good enough to let the recognition engine recognize meaning it, covers the necessary areas.

5.3 Conclusion about the project

In conclusion, both algorithms have their pros and cons. The final decision on which algorithm will be chosen to be implemented in the application software system depends on the requirements of the application system. If it is for example a security monitoring system or the like, the presence of a human there is much more important compared to detecting movement or movements? If this is the case, the algorithm of the recent human motion detection system that has been developed is more suitable for implementation. However, on the other hand, if hand gesture recognition software or the like is developed, the importance of movement or movement goes beyond the presence of a hand or an object in the video sequence. Next, it is preferable to use the prototype system algorithm to detect the previous and original human motion that has been developed.

Here are the disadvantages of Algorithm 1, Spatial refresh which uses spatial data to refresh the background:

- ☒ Small movements or changes in the background such as a jerking screen are not sufficient to update the background however are causing a lot of noise to be detected and spoiling the recognition performance. In this case, many false alarms of rectangles are being drawn.
- ☒ When a non-perfect subtraction result is obtained, the algorithm of binary image processing to enhance the image will result in a worse representation of a human shape.
- ☒ The aspect of the percentage value to be set for the background update is difficult since the ratio of having high updates rates and low updates rates can be both

causing problems to the algorithm. When high update rates, the human will have a higher possibility of being in the background image thus giving an extra-human as a movement since he had left his initial position. On the contrary, having low update rates gives many noises to the result caused by small changes in the real-world background scene.

☒ It requires very high illumination to be able to detect movements in the subtraction stage since information is being threshold by the Otsu function.

However, there are several advantages for this algorithm also which are:

☒ It eliminates the tracking algorithm since even human presence which is non-moving can also be detected.

☒ A perfect shape of the object is obtained if conditions are satisfied where the subtraction would return a great result.

Now we discussed the drawbacks of algorithm 2, temporal update + edge detection where the background is updated periodically,

☒ A non-moving human is not being detected.

☒ Only a portion of moving body parts is returned. (partially solved by using bounding rectangle calculations)

☒ Fails get a perfect shape of the object since the subtraction results give both background and current frame's difference pixels, resulting in an object with shadows. The distance of the object and its shadow cast by the background depends on the speed of the computer or processor implementing the system and the speed of the camera's capture measured in frames per second.

☒ The region may return human shapes being focused on the left side, center, or right side simply because the obtained result of the region is through different processes from the result of obtaining the bounding region.

The advantages of this algorithm are:

☒ Using moving edges, more detailed information of the object is gained.

☒ Solves the problem of the jerking camera in algorithm 1.

☒ Makes sure an updated background is being used unlike algorithm 1 which may be possible to be using an outdated background as its subtraction process‟ base.

5.4 Future works

Here, we had presented research on some image processing techniques implemented for motion detection algorithms and also some of the methodology and approaches of implementing a motion detection algorithm itself. The review includes research on a basic end-product implementation of a motion detection application.

We had presented some ways for implementing the approaches researched and also shared some ideas for alternatives ways to implement the motion detection algorithm.

An implementation and system design of a prototype system developed for testing purposes is reviewed in this report as well.

There are still many areas that can be further researched from this point onwards.

For example, the techniques introduced may be enhanced to suit some problem-specific applications or some domain-specific applications.

Also, the techniques may be further enhanced by implementing more useful methods and algorithms such as those involving tracking the object, which causes the motion events such as those using optical flows or image flows.

Here, human motions are being detected. However, future works may also want to recognize the pose or gesture of the human body registered by the algorithm implemented here in the prototype system.

References

[1] David Moore, “A real-world system for human motion detection and tracking”, California Institute of Technology, June 2003.

[2] Yang Song, Luis Goncalves, Pietro Perona, “Learning Probabilistic Structure for Human Motion Detection”, California Institute of Technology.

[3] Yang Song, Xiaolin Feng, Pietro Perona, “Towards Detection of Human Motion”, California Institute of Technology.

[4] Randal C. Nelson, “Qualitative Detection of Motion by a Moving Observer”, University of Rochester.

[5] Ming Xu, Tim Ellis, “Illumination-Invariant Motion Detection Using Colour Mixture Models”, City University, London.

[6] Lina J. Karam, David Rice, “Image Convolution Concepts and Applications online tutorial”, http://www.eas.asu.edu/~karam/2dconvolution/, Arizona State University.

[7] Jain, R., Kasturi R., Schunck G., “Machine Vision”, McGraw Hill, 1995

[8] Forsyth,D.A., Ponce,J., “Computer Vision: A Modern Approach”, Pearson Education, Upper Saddle River, NJ, 2003

[9] G.Johansson, “Visual perception of biological motion and a model for its analysis.”, Perception and Psychophysics, 14:201-211, 1973.

[10] S. Birchfield, “Derivation of Kanade-Lucas-Tomasi tracking equation”, http://robotics.stanford.edu/~birch/klt/derivation.ps , 1997.

[11] Zoran Zivkovic, Ferdinand van der Heijden, “Improving the selection of feature points for tracking”, Pattern Analysis and Recognition vol 7, no.2, 2004.

[12] Robert Pallbo, “Motion Detection: A Neural Model and its Implementation”, Lund University Cognitive Science.

[13] Ying-Li Tian, Arun Hampapur, “Robust Salient Motion Detection with Complex Background for Real-time Video Surveillance”, IBM T.J Watson Research Centre.

[14] Mike Smart, Barry Keepence, “Understanding Motion Detection”, Indigo Vision VB8000 motion detection enabled transmitter documentation.

[15] Andrew Blake, Kentaro Toyama, “Probabilistic Tracking in a Metric Space”, Microsoft Research Ltd. Cambridge U.K & Microsoft Research Redmond, WA, U.S.A.

[16] Milan Sonka, Vaclav Hlavac, Roger Boyle, “Image Processing, Analysis, and Machine Vision”, PWS Publishing, 1999.

[17] Linda G. Shapiro, George C. Stockman, “Computer Vision”, Prentice Hall, 2001.

[18] Thomas B. Moesland, Erik Granum, “A Survey of Computer Vision-Based Human Motion Capture”, Computer Vision and Image Understanding, 81:231-268, 2001.

[19] Bob Davies, Rainer Lainhart, etc. , “Intel Open Source Computer Vision Libraries”, http://www.intel.com/research/mrl/research/opencv/ , Intel Corporation.

[20] Ronan Collobert, “Torch Tutorial”, Institut Dalle Molle d'Intelligence Artificielle Perceptive Institute, 2 October 2002