Abstract

This piece of paper goes through the idea of image processing and pixel - based detection with background subtraction. A combination of the near-infrared technology and background subtraction technique is done, to solve the problem with the irritating projector light, which goes in the presenter's eyes during the presentation process. At first, to achieve the desired results the presenter is detected. Then, with a series of morphological filters a back solid mask, with the shape of the detected object is created. Furthermore, the mask applied to the next slide will be projected covering the presenter's eyes, preventing light to those points. Results shows that the project has managed to accomplish the objectives with a frame rate of approximately 18 frames per second, thus can be characterised as real time application. In addition, the object detection algorithm successfully detected 94% of the tested cases and cover mask has been produced.

Acknowledgments

I would like to express my thanks to my supervisor Dr. Naim Dahnoun for his advice, guidance and feedback throughout this project.

Thanks also to Ross, for his invaluable assist, my friend Yiannis for proofreading my report and experiments participants.

I would like to greatly thank my family for their constant support and encouragement.

Table of Content

Table of Figures

1 Introduction

This piece of work is an effort to examine the topic of ‘Real – time removing of foreground objects from projected scene’. The aim of this project is to detect objects in front of a plane in real-time and to remove the information of their position from a projected scene. My interest in this subject emerged from the essential need of the lecturer to demonstrate the material of the subject having at the same time eye contact with the people that are listening. Another determining factor in my choice was my personal experience with a fellow student who had an Attention Deficit and Hyperactivity Disorder (ADHD). Based on his experience, the teaching method the lecturer used with a projector was distractive to him because he was unable to concentrate to the screen and to the lecturer simultaneously. People with ADHD are influenced from a variety of environmental factors [1]. Consequently, lecturers could reduce the possibility of interruption in several ways. In this specific case, the lecturer should not move far away from the screen, where the students’ attention should focus on. Teaching methods that respond to all students’ needs should be taken in consideration and be applied [2]. Furthermore, an essential factor for an effective teaching is the lecturer, who should have communicability and be able to transfer his interest for the subject to the students. This could be achieved with the teachers’ eye contact towards the classroom [2]. This study is an effort to solve this issue, while a lecturer will be teaching next to the screen and have eye contact with the students individual. Last but not least, this technique will help more the non native language students who suppress more effort to understand because they are trying to focus on the lecturer and on the screen as well.

How does object detection help the lecturer? The theory behind the entire project is that, while the lecturer makes an effort to explain the projected slides, he needs to stand in front of the projector. As a result, the annoying light beam from the projector goes directly to his eyes. Immediately, this results in a reaction to move to a position outside the projected image, disrupting students’ attentiveness to the lecturer while they are trying to read on the screen. A solution to the problem is a series of steps which have to be completed. Firstly, the moving object has to be detected, which is the lecturer in this problem, and secondly, the projected information has to be removed to his place. To remove the light from the object position, black pixels will be used to cover the information of the slides. Removing that part of the slide does not have any influence on the students’ learning process because that piece of information is not readable.

In the process of object detection, which the accuracy and the quality of results are critical factors to the entire project, to identify the background and foreground objects, techniques like background subtraction with pixel intensities and depth measurement could be used.

Background subtraction is a technique with quite simple theory that is widely used for pixel-based object detection applications. Nevertheless, its accuracy is proportional to illumination and motion changes. Different approaches based on the background subtraction technique exist which propose reliability and quality in solution. The main difference, is proposing an alternative way of how the calculation of the background model image or pattern must be and how this model is being updated with time.

Another way to determine whenever a point in an image is either foreground or background is by knowing how far from the viewer that point is. In contrast to background subtraction with pixel intensities, these techniques use the depth information to characterize each pixel and to create the background model. Stereo images captured from two or more cameras are used in these methods to provide the sample. With techniques similar to the way the human brain understands the 3D space, depth information is extracted from 2D images Having the depth information of each point of the image, in foreground the points with less depth are nearer and the points with further depths are the background. This technique is used to eliminate problems that occur due to physical changes of the background, such as great illumination changes or noise created by camera movements. The background model is updated in most cases, like background subtraction with pixel intensities.

Last but not least, all methods and techniques must be able to process on real time and to be implemented on standard existed hardware. Nowadays, technology evolution has lead to very fast computation speeds. Also, the design of hardware devices specializes for image processing applications, allows intensive computations to be performed in real time. Digital signal processors are a great example of high computation efficiency and low power consumption.

The rest of the document is formatted as follows. In the introduction section a more detailed description of the problem is given. The second chapter describes the aim and objectives of the project, while in the theoretical background section more information can be found about the techniques that were used to achieve the solution. Moreover, the implementation of the project is explained in depth in the implementation section. The results are analysed and explained in tests and results chapter. Next can be found an evaluation of the main objectives. Finally, a conclusion section suggests any ways for further improvements.

2 Aim and Objectives

2.1 Aim

The aim of this project is to detect objects in real-time in front of a plane, object position identification and adjustment of the projected slides from a projector in order to remove projected information to object position. As a result, the detected object would not have any light projected to it.

2.2 List of Objectives

The main objectives to achieve this desired result are:

1. Study the appropriate object detection method to fulfill the problem requirements
2. Define Background – Foreground Models
3. Implementation of Background - Foreground models
4. Implementation of object detection algorithm
5. Test algorithm with different conditions
6. Evaluation of the results obtained from objective 5, review methods
7. Compose the real – time object detection system
8. Final tests of the system

The rest of this section briefly explains the main objective of the project and the work that has be done in each one in a few words. More detail evaluation of the objective can be found in *Objectives Evaluation* chapter

2.2.1 Study the appropriate object detection method.

The choice of the appropriate object detection method was critical for further implementation of the project. The hardware that was finally selected to work with was proportional to the method. The method used to detect objects was based on the environment where the project would run. Similar to any other applications the working environment must be studied for any special characteristics before the implementation of any algorithm. Likewise, for this project, the room environment was studied to obtain any helpful information about the light sources, the average distance between the projector and the lecturer and the position of the projector in the room.

Moreover, the projector will operate at the same time with the object detection part of the system; therefore, its effects to the working environment were taken in account. The simultaneous projection of the scene over the object has as a result, morphological and illumination changes of the objects causing problems on correct detection.

In addition, the aim of this project was not to implement a new object detection algorithm better than already existing ones, but to solve the problem with the disturbing light of the projector. The problem that tries to be solved by this project is frequent and until the time the projects began no solution have be found. Thus, the final applications were implement and tested in terms of project aims.

The outcome of this study has helped to understand more clearly the condition of the working environment and also, made the choice of the object detection algorithm much easier. From all the environment study outcomes and strictly the aim of the project, three methods have been selected to be compared. *Basic Background Subtraction* is the method chosen to be used to determine the background and foreground regions of a frame. Practically the new object in the scene does not need to be identified, but only locate its position in the frame. Hence, there is no need for further analyses of foreground region. This method was the most inexpensive and has the less process power in contrary with a 3D camera solution or a stereo vision solution. More on the object detection methods can be found in the *Theoretical Background* section.

2.2.2 Define Background – Foreground Models

The background model has a significant value to the operation of the object detection algorithm. Based on the object detection method, the background – foreground models were defined to meet the project needs. Moreover, in a background subtraction approach, the background model is a critical factor for the system reliability.

Methods and functions that the system will distinguish between background and foreground, as well as how the background model will be updated, was defined in this process step. Additionally, the detected foreground region, which is everything on the frame that has not been classified as background, is the one that the system works with. The foreground is used for the creation of a dynamic mask that covers the object in the modified slide. Morphological filters are applied to the foreground to deal with image noises and are removed before it any other procedure.

2.2.3 Implementation of Object Detection Algorithm

This objective aims to implement the selected object detection algorithm. It requires being in real – time but for the initial implementation, still simple pictures were used.
A high level language was used for the initial implementation of the algorithm, although the final form of the code is written in C using OpenCv library. In addition, in this implementation phase the background model and morphological filters were completed. This was the most time demanding step due to the large amount of function coding.

2.2.4 Test Algorithm with Different Conditions

This was the first testing step; the implemented algorithms were tested in a working environment with frames taken from simple camera and from the camera that was assembled for the purposes of the project. Tests were performed in possible situations that could happen in the room environment such as illumination changes, second object in the vision range and objects with projected pattern on them. The test results were very practical for the further improvement of the algorithm and the whole system.

2.2.5 Compose Real – Time Object Detection System

The combination of all the selected techniques and all the algorithms were done in this part of the progress. Furthermore, the first time that all the components of the system worked together was in this process step. For the purposes of this project, the hardware system that was selected to run the application were personal computers due to the fact that are mostly used in presentations. The use of personal computers for this project has also changed the specifications for the camera. The use of a USB camera was then more suitable, thus, it was more flexible than a camera with a specialized input cable. In addition, a simple portable projector was used. All the components of the system were worked simultaneously. The main elements of the system is the projector, the custom modified camera, the application and the personal computer as the processing unit.

2.2.6 Final Tests of the System

The entire system was tested in order to fulfil the aims of this project. Real room environment was simulated to gather information about the functionality of the system. Testing was performed on people with different skin color to check that the system functions correctly to any skin colour and shade. Moreover, people wearing glasses were involved in testing. Glasses usually change the shape of the head and reflect light; hence the system was checked thoroughly in this kind of situations. Furthermore, tests were performed to show that the system works when the distance between the optical sensor and the screen is altered. Finally, the direction where the person is looking is also checked, to ensure that the system works with any angle or orientation of the person's head.

3 Theoretical Background

3.1 Images

In theory, we have a simple image with two dimensional functions where their values give the brightness of the image at any given point [3]. Digital images differ from a simple photograph in the x, y and f values which are distinct in digital images. A digital image can be considered as a large array of sampled points from the continuous image and each point has a specific quantized brightness. The points are called pixels and the surrounding pixels constitute a neighbourhood. An example of a neighbourhood is given in the figure below. If the number of columns and rows existing in a neighbourhood are of equal number, then it is crucial to provide which pixel in the neighbourhood is the current pixel.

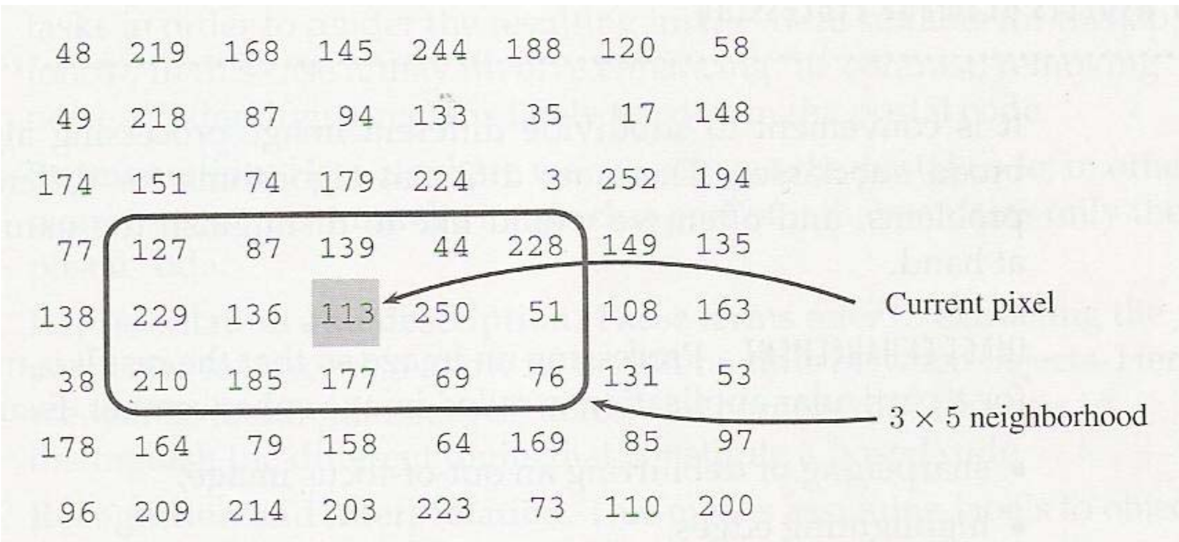


Figure 1: Example of a digital image and neighborhood.

There are four types of digital images; binary, greyscale, true colour or red-green-blue and indexed. In *binary* digital images, each pixel is composed of black and white. There are two possible values of pixels; 1(one) bit per pixel is needed. This kind of images is operative in terms of storage. They may also include text (printed or handwriting), fingerprints or architectural plans. *Greyscale* images include pixels that are a shade of grey, normally from 0 which corresponds to black colour, to 255 which is represented as white colour. The range states that each pixel can be represented by 8(eight) bits or exactly 1(one) byte. This is a very common range for image file handling such as image data indexing and processing, especially for applications designed to fit on digital signal processors. Greyscale images are mainly used in medicine (x-rays) and images of printed works as well as in the filed of image processing. Most of the image processing algorithms are using greyscale images. *True colour image* is the image where the pixel has a specific colour. The colour is described by the amount of red, green and blue colour in it. Such kind of image requires 24 bits for each pixel and thus, it is also named 24-bit colour image. Moreover, it may consist of a stack of three matrices, representing the red, green and blue values of a pixel having as a result the correspondence of three values in each pixel. Many of the colour images have a small subset of more than 16 million possible colours. In order to store and to handle them conveniently to the file, the image has a colour map or a colour palette which is a simple way to list the colours that are used in that image. Each pixel has a value but it does not give its colour, something that happens in red-green-blue, but it gives an index to the colour in the map. For this reason, these types of images are called *indexed*.

3.1.1 Image Processing

Image processing is the action through which an image is changed and moderated compared to the original.[3]

Nowadays, image processing is one of the most popular technological actions and there is almost no area of technological endeavour that is not influenced in some way by digital imaging. Some of the fields in which image processing is used are gamma-ray, X-ray, imaging in the Ultraviolet Band, in the Visible and Infrared Band, the Microwave Band and imaging in the Radio Band.

Image processing is divided into several steps through which each one should process the image in order to achieve the desirable result. The first step in image processing is the image acquisition which is considered simple because it is like an image being given in digital form. *Image enhancement* follows and it is the procedure of moderating an image in order to be more appropriate when compared to the first image and for a particular application. Another step is *image restoration*, which is also done for the improvement of an image. Then, we have colour image processing that is of great importance due to the fact that the images which are being processed are used very frequently in the internet. In addition, *Wavelets* are the foundation for representing images in various degrees of resolution. Compression, deals with techniques for reducing the storage required saving an image, or the bandwidth required transmitting it.

Figure 2, shows the wide area of image processing. As an example, zooming in to an image is a result of several algorithms running on the background of an application. Information that was unobservable is readable after the process has been carried out.

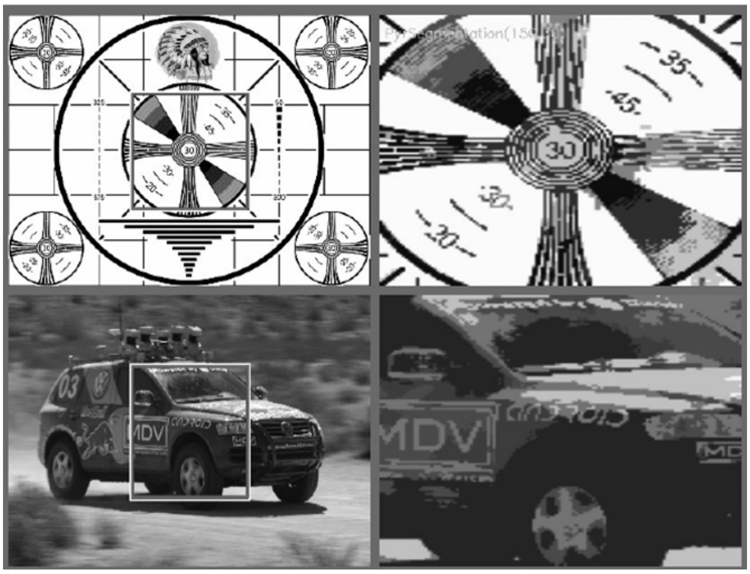


Figure 2: Example of image processing [4].

Morphological processing, on the other hand, has to do with tools which are used to extract image features and are useful to the representation and description of shape. *Segmentation* processes divide an image into component parts and/or objects. These procedures are classified into two different categories; *autonomous segmentation* and *rugged segmentation.* Autonomous segmentation is considered one of the most complicated tasks in the procedure of digital image processing. On the contrary, rugged segmentation is considered easier to be managed and in general makes the processing successful, especially when imaging problems are found.

Digital image processing requires also *representation* and *description*. Both follow the performance of the segmentation. Representation is divided into two classes; firstly *boundary representation* and secondly *regional representation*. Boundary representation is suitable when the attention is on the external shape features like corners and inflections. On the contrary, regional representation handles with the internal properties, for instance, texture or skeletal shape. *Description*, which is also called *feature selection*, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another. Finally, there is recognition which is the procedure which allocates a label to a specific object regarding the express of the descriptors.

3.2 Object Detection

Object detection is considered one of the most essential processes in video analysis. It is associated with image processing and computer vision that aims to detect semantic objects in images and videos such as humans, cars or buildings the viewer is interested to analyse further. In most of the cases, moving objects are interesting and challenging while static parts are not. Thus, object detection can be considered as the detection of motion.

Moreover, sophisticated algorithms were developed for object detection to achieve specific tasks like identifying faces [5], signs, speech or even the people's ethnicity, gender and age, while real – time tracking of moving objects seems to push several applications to the next level. More than a few applications use object detection to maximize the accuracy of their outcome, while at the same time, others rely on their existence, including video surveillance and security systems. Since several algorithms and techniques were developed to match a specific application, frameworks come to put on a test the different aspects of each one [6 7].

This project is directly connected to object detection and thus, the first step of the whole process, is to detect the object. The novelty of the system is that no other application has used object detection to solve problems that are similar to the one the project works on.

3.3 Background Subtraction Techniques

In most of the cases, the normal approach for detecting a moving object from a background scene is background subtraction. The background subtraction techniques and vision systems are linked together with the first being the pre-processing step for object detection and tracking. On one hand, results show that the existing algorithms are satisfied. On the other hand, a big amount of them are influenced by the global and local illumination changes and to be more specific, shadows and highlights are elements which can cause this consequences.

The basic theory of background subtraction is to subtract the current frame or image from a background reference image. The fundamental steps of the algorithm are:

- Background reference image creation.
- Threshold selection, such as to obtain desired results.
- Subtract the two images to find the moving objects.

- Compare points of the new image with threshold to classify its state, background or foreground.

- Apply morphological filtering to eliminate small regions and noise.

After subtraction and thresholding steps, a new image is created containing information about moving or new objects. Therefore, results are taken in account to update the background reference picture and to segment, in most cases, foreground pixels into adjacent regions.

The creation of the reference images is crucial, because the quality of algorithm results depends on that. Noise that is created by the camera movements, high alternations to density due to rapid illumination changes, weather condition and slow-moving or stopping objects can damage the background reference image. To deal with that, most methods computes the background reference image relying on statistical approaches, assuming that each point of the image is a random value and can be estimated using probability distribution based on information taken by previous images [8]. “Person Finder” system use Gaussian distribution to define the background image. Mixture of Gaussian is also used to define the background image [9].

$$| \text{frame}_i - \text{background}_i | > Th \quad (1)$$

Figure 3 shows in a straightforward way how the background subtraction technique aids in object detection. This method assumes that the background is something known. The first part (a) shows the background reference picture. A new frame is shown in (b) and finally in (c) the foreground object is detected.

(a) (b) (c)

Figure 3: Background subtraction in simple manner

3.3.1 Basic Background Subtraction

Basic background subtraction (BBS) is the simplest algorithm that can be used for object detection. By taking the absolute difference of the current frame and the background reference image and by comparing each pixel value with the threshold, it defines the foreground and background point on the frame. A point is classified as foreground if its pixel value is higher than the threshold value and vise versa for background.

$$|I_t(x,y) - B_t(x,y)| > T \tag{2}$$

In formula 2 $I_t(x,y)$ is a 3 x 1 vector showing the pixel value in time t for the current frame. $B_t(x,y)$ is the pixel value from the background reference picture at the same position. T is the threshold value. $B_t(x,y)$ is defined as the mean (μ) computation from a series of N previous frames of pixel values for a certain position and this is the way that the background reference image is created each time. Thus, from equation 2 the new formula is:

$$|I_t(x,y) - \mu_t(x,y)| > T \tag{3}$$

This approach requires the need of huge memory resources. Therefore, a method that defines the background reference image as the running average is introduced.

$$B_{t+1}(x,y) = \alpha * I_t(x,y) + (1 - \alpha) * B_t(x,y) \tag{4}$$

Where α is the learning rate. Thus, from equation 4 the background reference image is computed as the chronological pixel value average.

Typically, α is equal to0.05, so the new background reference image consists from 5% of the new current frame to 95% of the old background reference image.

Theoretically, after the computations, pixels in the image that is associated with the same object would have same values, so are either characterized as foreground or background. Apart from that, a morphological filter is applied on the images to eliminate small regions and separated pixels, such as dilate and erode filters.

3.3.2 Single Gaussian Model

In this technique, the information from each point of the picture is taken as a 3x1 vector, defining by this way the intensity and the colour. Gaussian distribution is used to define the values of each pixel. The background reference picture is therefore computed and updated with the use of $\mu(x,y)$ and covariance $\Sigma(x,y)$.

$$\mu_t(x,y) = (1-\alpha)\ \mu_t(x,y) + \alpha * I_t(x,y) \quad (5)$$

$$\Sigma_t(x,y) = (1-\alpha)\ \Sigma_{t-1}(x,y) + \alpha*(\ I_t(x,y) - \mu_t(x,y)) * (\ I_t(x,y) - \mu_t(x,y))^T \quad (6)$$

In equations 5 and 6, $I(x,y)$ is a 3x1 vector containing information about the pixel colour and intensity. Value α is constant and represents a learning factor. Following the update of the background reference picture, each pixel is classified either as background or foreground. All the foreground pixels are grouped together into blobs. Furthermore, the current frame is compared with the background reference picture using the log likelihood.

$$l(x,y) = -\tfrac{1}{2}*(\ I_t(x,y) - \mu_t(x,y))^T * \Sigma_t^{-1}*(I_t(x,y) - \mu_t(x,y)) \ - \tfrac{1}{2}*\ln|\ \Sigma_t\ | - {}^{m}\!/_{2} *\ln(2\pi) \quad (7)$$

If the result of the likelihood functions is small then the corresponding pixel is classified as foreground or active.

3.3.3 W4 Technique

This technique was used in W4 system [10], which runs in real time and aims to detect and track multiple people, while at the same time monitors their activities. W4 is designed to work outdoors and use grayscale images.

Moreover, W4 makes use of bimodal distribution to observe background model values taken from a statistical analysis of a series of background values during a training period. Each pixel in the background reference picture is characterized by three values; pixel minimum intensity $M(x)$, pixel maximum intensity $N(x)$ and pixel maximum intensity difference between consecutive frames $D(x)$. If for example V is considered to be an array of N frame series that was captured in the training period, then the background reference picture at a given point x would be as follows.

$$M(x) = \min_z\{V^z(x)\}$$

$$N(x) = \max_z\{V^z(x)\}$$

$$D(x) = \max_z\{|\ V^z(x) - V^{z-1}(x)\ |\} \quad (8)$$

$$\text{Where } |\ V^z(x) - \lambda(x)\ | < 2\ \sigma^2$$

In equation 8, $\lambda(x)$ is the median and $\sigma(x)$ is the standard deviation values of intensities for a given pixel at position x for all the frames in V. $V^z(x)$ is defined as background pixels. To familiarize the background reference picture with the environmental changes, W4 uses two updating methods; pixel-based update, to adjust with the background illumination changes and object-based update, to adjust to physical background changes. A car that is static for a long period of time is an example of object-based update, which then has to be appended to the background reference picture.

To define whenever an object is in foreground or not, four stages are undertaken: Thresholding, cleaning noises by erosion, the removal of small regions with the use of morphological filters and binary analysis of connected components.

A pixel in position x is considered as background when:

$$(|I_t(x) - M(x)|) < kD(x) \quad \text{or} \quad (|I_t(x) - N(x)|) < kD(x) \tag{9}$$

If the conditions are not met, it is being considered as foreground. In equation 9, k is a threshold constant and works better if set to 2, based on experiments performed in [10].

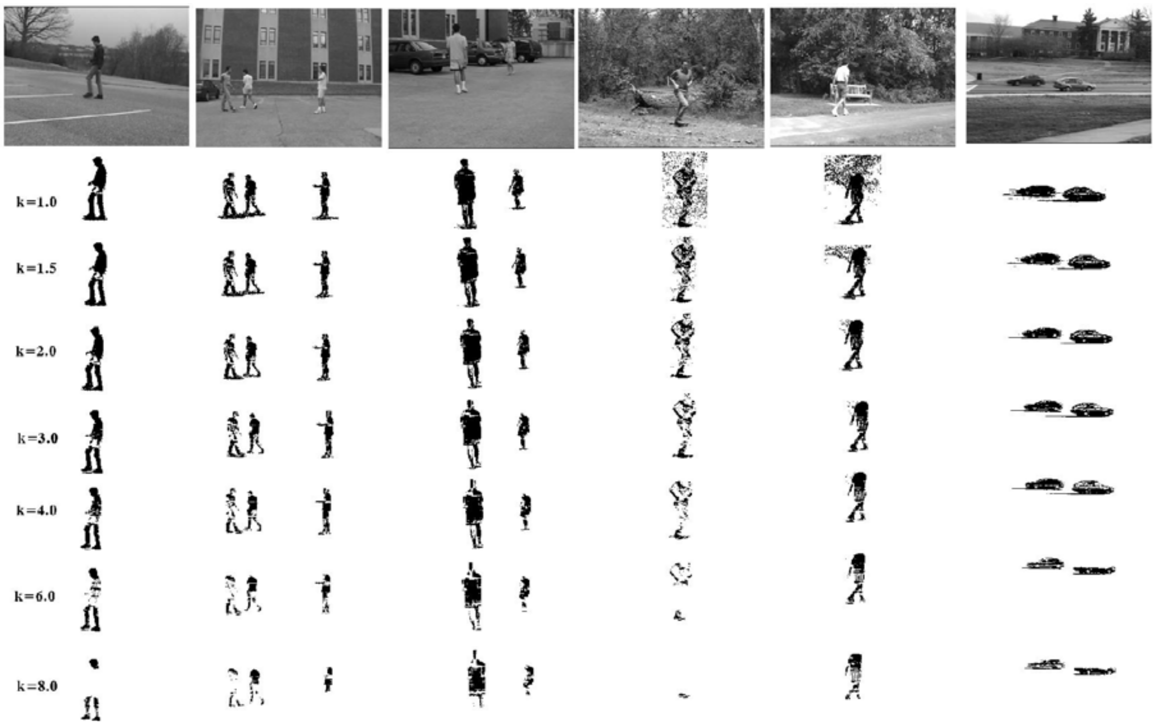


Figure 4: Example of how W4 detects foreground objects for different threshold (k) values.

Figure 4 shows the disparity of a foreground object detection given a different threshold value. At k = 2, all the foreground object regions are clearly shown. For k values that are higher than 2, regions of the objects are considered as background and are subtracted from the image resulting in low detection rate. On the other hand, for k values less than 2, impurities from the background are not filtered properly and are shown as foreground.

3.4 Stereo Vision

Binocular disparity is the difference in location of the same object between two almost identical images, the one of left and the other from the right eye [11].

Both eyes act like a telescope placed together with a small distance among them. If an object is observed which lies opposite to the two telescopes, its position when observed with both telescopes at different times will be different. The object moves slightly to the right or to the left. The object location changes based on the right or left eye and this is because of the eyes' horizontal separation. This is how human eyes function.

The brain uses this disparity to observe depth information from the 2D images. Bear in mind that, the difference in location of the same object in stereo vision refers to the displacement of its coordinates to the X and Y axis.

With stereo analysis, which is the process that is being used to obtain depth distances from stereo images, object detection techniques can create the background image in a more reliable approach [12]. In such a method, it is not necessary to find techniques to decrease errors that result from illuminations changes, shadows and noise due to camera movements. Moreover, objects that are in different depth but in the 2D images seem to be adjacent to each other, can be recognized as different objects and can be processed accordingly.

Figure 5 shows how far and near, in relation to the eye, objects are captured. Near objects are seen with a different angle than far away objects, and so the disparity between right and left eye has greater values. It is in this way that the depth information is taken from a stereo image.

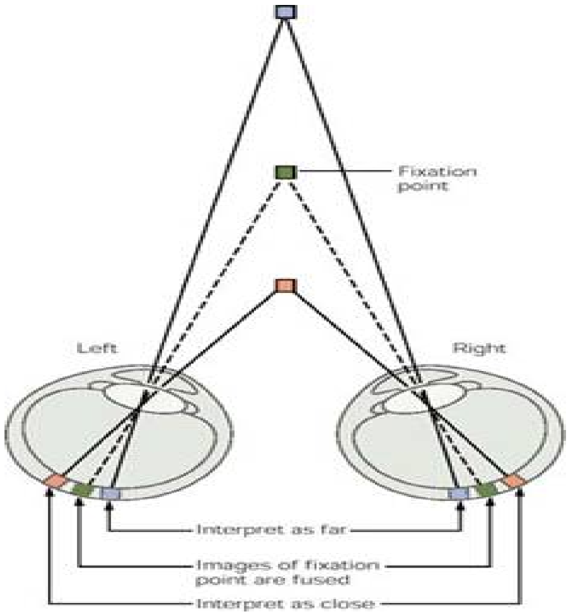
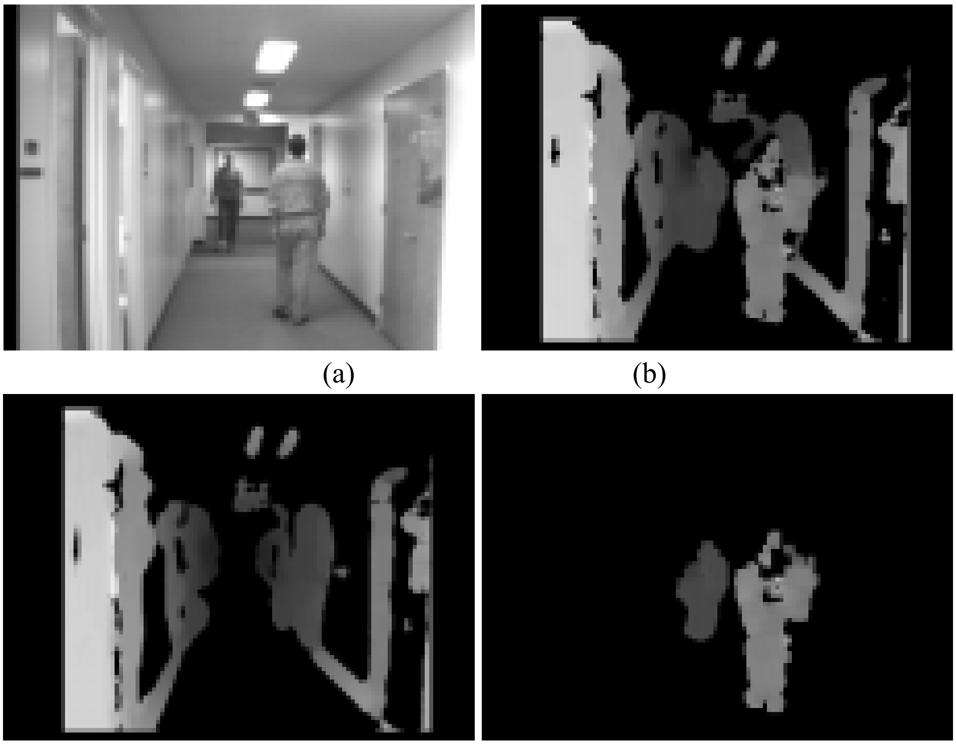


Figure 5: Human eye and depth

3.4.1 Stereo Analysis

Stereo analysis aims to find equivalent elements between the right and the left image. There are two types of matching methods. On one hand, local methods try to match small elements between the two pictures on the basic characteristics of the elements. On the other hand, global methods take in account physical limitations, such as surface continuity. Characteristics of the elements are frequently selected to be the image's corners, because corners remain corners in both images, and therefore are viewpoint-independent [13]. In an area correlation method, there is a negotiation between sizes of the elements that are compared. Small area elements tend to be similar between the stereo images more frequently. Large area elements are more likely to have more noise, hence correlation between the images is difficult. The area correlation method discussed in [13], includes five steps: Geometry correction, Image transform, Area correlation, Extrema extraction and Post-filtering.

Figure 6 illustrates the technique that is used to detect a foreground object in reference [12]. (a) is the initial image while in (b) the stereo disparities are shown. (c) and (d) show how the background and foreground regions are modelled and distinguished respectively.



(a) (b)

(c) (d)

Figure 6: Object detection using disparity images

3.4.2 Detection with Stereo

Detecting an object using the depth information instead intensities information for each pixel, is a way to overcome problems such as global or local illumination changes and shadows which add impurities to the background reference picture. In references [12] and [14], methods are suggested to detect objects based on the depth information taken from stereo pictures. In such a method, the disparity image must be computed for each pair of stereo images. The area correlation method is used to extract the disparity image. In the disparity image, the disparities are inversely related to the depth as shown in the following formula.

$$D = k/x \tag{10}$$

In this formula, d is the disparity, k is a constant associated to camera parameters and x is the distance from the image to the object. Based on that, a background model is constructed and the foreground points of the image are grouped. A series of morphological filters such as dilation and erosion, are used to clear the foreground images from small regions or fracture lines. Afterwards, the background disparity image is updated with the use of the following formulas which give the parameter for the normal distribution.

$$\mu_t(x,y) = \alpha\, \mu_{t-1}(x,y) + (1 - \alpha)\, V_t(x,y) \tag{11}$$

$$\sigma_t^2(x,y) = \alpha\, \sigma^2_{t-1}(x,y) + (1 - \alpha)\, [V_t(x,y) - \mu_t(x,y)]^2 \tag{12}$$

where V_t is the current disparity, μ and σ is the median and standard deviation for the given point.
The constant α is set to a high value so that the amount of data that is taken in account to compute the distribution is sufficient.

Figure 7 shows an example of the multiple person detection method discussed in reference [12]. It also illustrates that two different people at different depths are detected by using two methods; disparity image as the background model and person templates to detect people.

Figure 7: Person detection.

Moreover, these techniques use person templates to detect people. Person templates are binary images in 2D that correspond to the human body. As the foreground region is detected, the next step is to compare it with the person templates. If the result of the comparison is high, usually above 75%, then a new person has been detected. Apart from that, a person can be detected in different depths. As the depth of the foreground region, where the object is detected is known, using the similar triangles method, the width of the person that is expected at that depth is estimated. The width’s corresponding template is selected and is compared with the foreground region. With these techniques, only one person template is used with different scales. In addition, person templates save computation time and therefore are small and do not need a lot of memory resources. Finally, the person region is subtracted from the foreground and the same process is repeated to match with more people.

3.5 Infrared radiation

The English astronomer William Hershel discovered the infrared range in 1800 [15, 16]. He was trying to find out which colour in the visible spectrum transfers heat from the sun. The electromagnetic spectrum area between the visible and microwave regions is called infrared, which corresponds to wavelengths from 0.74μm to 300μm. The infrared region (IR) is divided into three sections: the near-infrared, mid-infrared, and far-infrared.

- Near IR: 700 nm–1400 nm (0.7 μm – 1.4 μm, 215 THz - 430 THz)
- Mid IR: 1400 nm–3000 nm (1.4 μm – 3 μm, 100 THz - 215 THz)
- Far IR: 3000 nm–1 mm (3 μm – 1000 μm, 300 GHz - 100 THz)

Near infrared region acts like visible light and is used for the purposes of this project. Moreover, photography in near infrared range has embedded novel characteristics to art images. Images are also taken in the far infrared range and are called thermal images. Such images are taken from a special camera that sense thermal emission from person and objects.

Images in the infrared region have special characteristics and thus are commonly used into several military projects. In addition, application for research and civilian purposes also exist. Target acquisition, night vision surveillance and tracking are some of the applications that have been implemented for military reasons. On the other hand, environmental monitoring, head analysis, food and material analysis, remote controlling, weather forecasting, and measuring heat from distance are areas where infrared is commonly used.

Figure 8 shows the electromagnetic spectrum from *Gamma ray* until *Radio waves*. It also shows the frequency and temperature that corresponds to each part, while with the use of images the wavelength size of each region is illustrated.

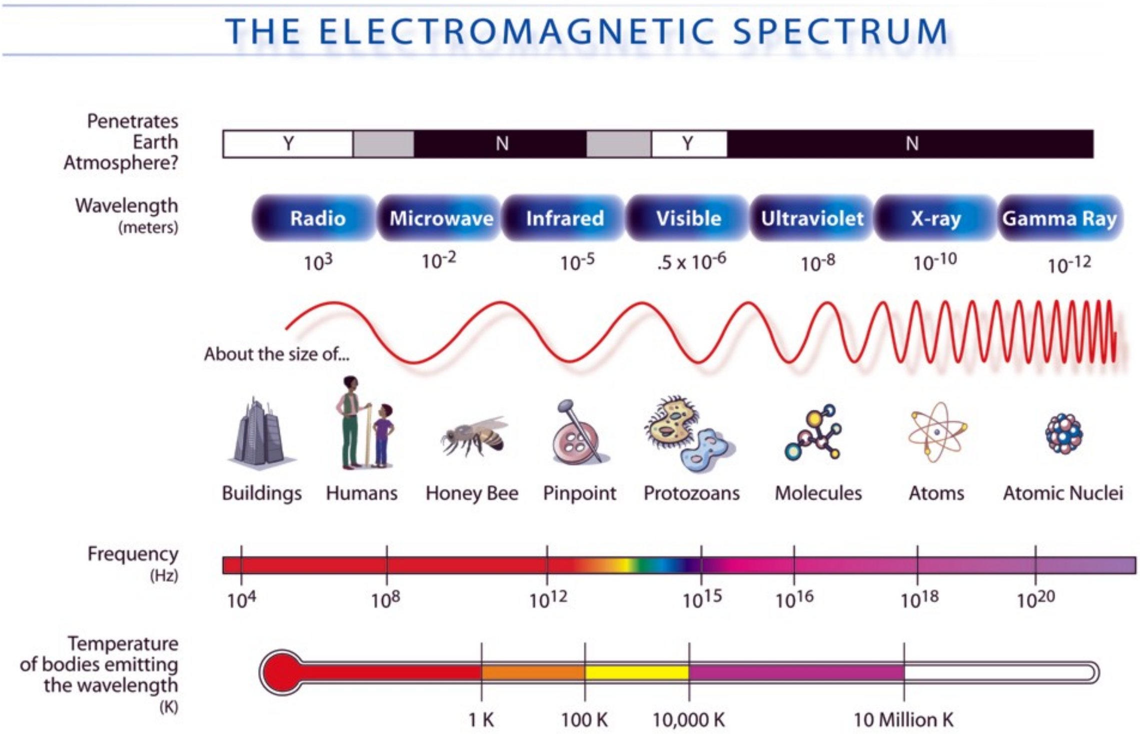


Figure 8: Electromagnetic spectrum [17].

Near infrared spectrum has photographically been measured by Abney and Festing, to be in the range of 700 to 1200 nm in 1881 [15]. Despite the fact that near infrared is the closest in wavelength to visible light, human eye cannot sense it. Thus, the observation of near infrared characteristics is done through silicon sensors. Near infrared radiation is used in quite a lot of fields nowadays. The fact that near infrared has less attenuation losses in glass, has made their usage in fiber optic telecommunication more frequently. Moreover, near infrared is used to analyse materials such as plastic, polymers and fake notes. In addition, dairy and baking industry use near infrared to analyse raw materials for their products. On the other hand, beverage industry use applications based on near infrared to choose ingredients for their alcoholic beverages.

Figure 9 shows visible and infrared range of the electromagnetic spectrum. Infrared radiation is above the red colour of visible light and that is the reason for its name. From the word *infra* that means beyond in Latin. Moreover, in this figure the corresponding wavelengths for each colour in visible spectrum and for each part of the infrared light are shown.

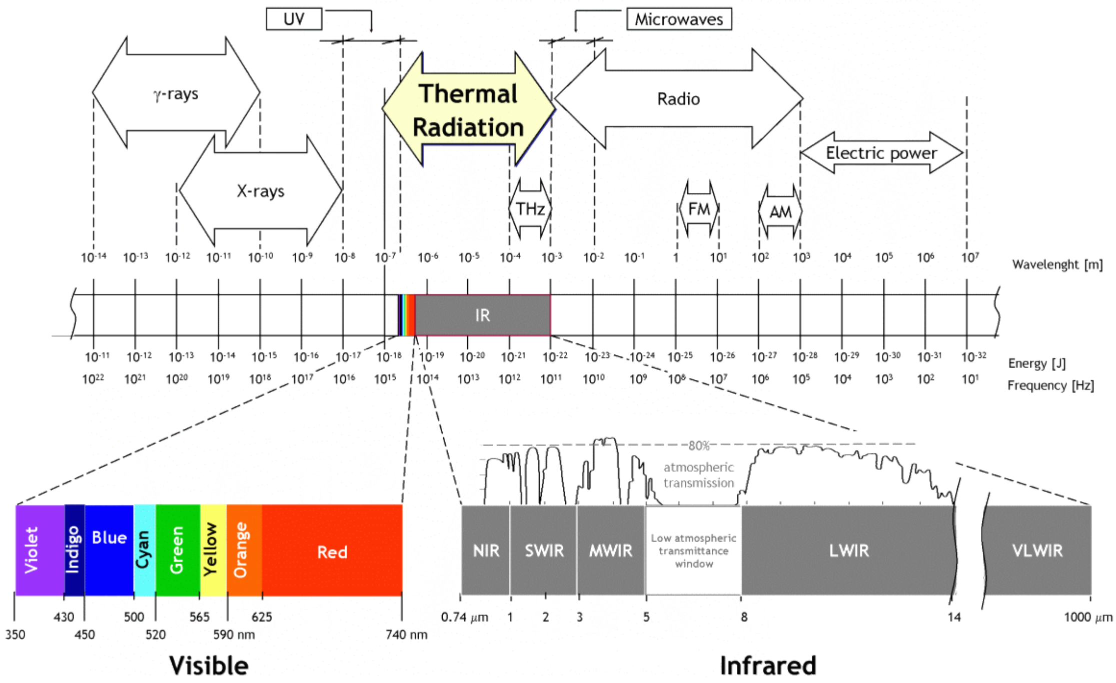


Figure 9: Infrared spectrum [18]

3.6 Infrared Camera

On the contrary, standard cameras use visible light to create an image, while infrared cameras operate at greater wavelengths and create an image using the infrared radiation. Based on body temperature that results to heat being radiated by a body, infrared cameras became a valuable tool for use in a dark or smoke filled environment for rescue operations.

They are related to the project through the fact that images from infrared cameras are likely to have a single color. Moreover, in reality these images are shown with some color to be clearer and more understandable. It is generally uses white color for the warmest parts of the image, red and yellow for medium temperatures and blue for the coolest parts. Due to the fact that this project aims to detect moving objects in a class where objects in the vision range of the camera would have lower temperature than the lecturer, infrared cameras may be a solution to factors that affect the results of object detection methods and techniques.

Night vision cameras on the other hand, are operating in the same way with ordinary cameras. However, they lack of a filter that prevents sensing near infrared light.

Such cameras are used in surveillance systems to monitor areas at night. With out the cut off IR filter, this kind of cameras can sense near infrared radiation and with an infrared illuminator can capture images at total dark conditions.

3.7 OpenCV

OpenCV is a library that was born in Intel, written in C and C++ and is related to computer vision applications [4]. It open source and console free library, meaning that can runs under several operating systems. OpenCV can benefits from multicore processors, thus is specialised in real – time applications with effort to computational efficiency. Stereo vision, product inspections, robotics and medical imaging are some of the fields that cover by functions of OpenCV library.

The project makes use of the OpenCV library to improve the performance of the computations. Moreover, specialised fixtures, such as ROI are used in the way of solving the project problem.

4 Implementation

This section provides information about the method that was used and the steps that were taken to achieve the desired results. In addition, the hardware and software that were used are explained in more depth.

4.1 Hardware

The hardware that was used for this project was a night vision infrared camera with embedded infrared LEDs and an IR-pass filter. At the final stage, the elements were combined to create a solid optical sensor for easier use. The components were selected after experiments in terms of the projects aim and objectives.

4.1.1 Infrared night vision camera

Infrared cameras are nowadays more commonly used by surveillance systems. The main advantage of using this type of cameras is that they provide clear view of the scene in total dark environment conditions. Therefore, they can monitor the area at night, where the most malicious actions take place. Night vision cameras use the infrared part of the electromagnetic spectrum, thus the information that are capturing is the reflection of the infrared light. The fact that the infrared light is not visible to the human eye makes them perfect for their usage. A large amount of night vision cameras are self illuminated and have a ring of infrared LEDs surrounding the image sensor. A web night vision camera was used for the purpose of this project. The camera [19] has the ability to connect to computer thought a USB cable and has built in six infrared Light Emmiting Diodes (LEDs). The video quality is 1.3 Mega pixels at 640 x 480 pixels per frame. Also, a ring in front of the sensor does the focusing function manually. It can work with almost all Windows versions and can be used with most internet communication applications. Figure 10 shows the infrared night vision camera that was used, before any modification.

Figure 10: Night vision web camera with 6 infrared LEDs embedded.

After the first experiment, the infrared light that is created by the LEDs was not strong enough for the project demands. therefore, for the testing experiments the infrared light provided by the environment was used. The reason for that was mainly the distance between the camera and the screen. The camera LEDs have the ability to emit infrared light for the distance of about one meter. However, the average distance between the projector, where the camera is placed, and the screen, is approximately five meters. Hence, tests in total dark conditions are not viable

A critical factor in using an infrared camera was the quality of the images that were captured by a typical and night vision camera. An infrared-pass filter was used in front of both camera lens to let only the infrared light pass through and thus, the image was a result of infrared light reflection only. The typical camera has poor quality images compared to the ones that were captured by the night vision camera. A filter that is built in the lens of most common cameras is the reason for poor quality infrared images. The filter cuts off a big amount of infrared light to ensure better quality in true color images. Therefore, only a night vision camera can be used for the project which allows for the entire available infrared light to pass thought the lens.

Figure 11 shows two images captured from a typical camera with an infrared filter. Small amount of infrared light passes through the lens; hence, it can only capture outdoor images. Indoor images are dark and thus, even when the images are processed the results are inadequate for further analysis.

Figure 11: Images from typical camera with infrared pass filter.

As it is clearly shown in Figure 12, images from night vision camera with the infrared pass filter in front of the camera lens are clear enough to be process. Indoor and outdoors images are containing sufficient light details. Both cameras are true color and any changes on colors are due to the infrared filter.

Figure 12: Images from night vision camera with infrared pass filter.

4.1.2 IR Filter

IR filters are usually called the filters that allow infrared light to pass through while blocking any other wavelength. Typically, camera sensors are sensitive to near-infrared wavelength and infrared cut-off filters are used to block infrared light. This is done to reduce the event of unnatural looking images. In contrast, an IR pass filter is used for this project. The purpose of the filter is to eliminate any visible light from the captured images. A major problem was the alteration of physical characteristics of the object that are exposed to projector light. The light from the projector in most cases changes the color and light characteristics of the objects. As a result, there was faulty detection and no detection of the objects, which also matched the experimental results. Therefore, with the use of the IR-pass filter was achieved to eliminate that problem.

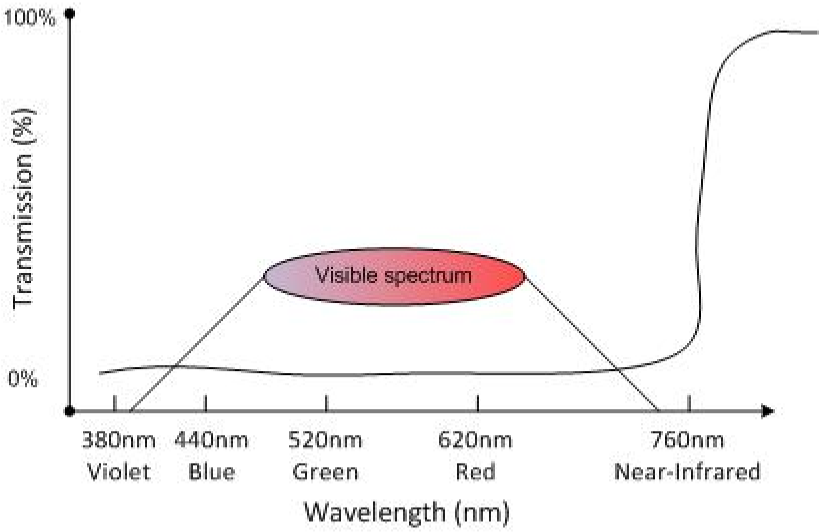


Figure 13: IR-pass Filter 760nm functionality

The IR filter that was used for this project allows any wavelength which is greater than 760nm to pass through. Figure 13 shows the filter functionality with the visible and near infrared wavelengths. The visible spectrum is in the region where no transmission passes through the filter. In the wavelength range above 760nm (394THz), all the light passes through.

Figure 14 illustrates the functionality of the filter over a laptop screen. The screen works with the same principle as the projector. It emits light to show the information to the user. Both pictures were captured with the same night vision camera and without any special effect at the same time period. The left image shows the screen and the IR filter in front of the camera lens. The picture at the right shows the laptop screen without the IR filter. It is clear that the image captured with the IR filter illustrates the screen being turned off. This is a real example of how the filter cuts off the visible spectrum and removes any information lying in that range. The fact that near infrared light performs with the same way as the visible light, results in the image shown at the left hand side. The darker color spots are a result of near infrared light being reflected less and vise versa for lighter spots.

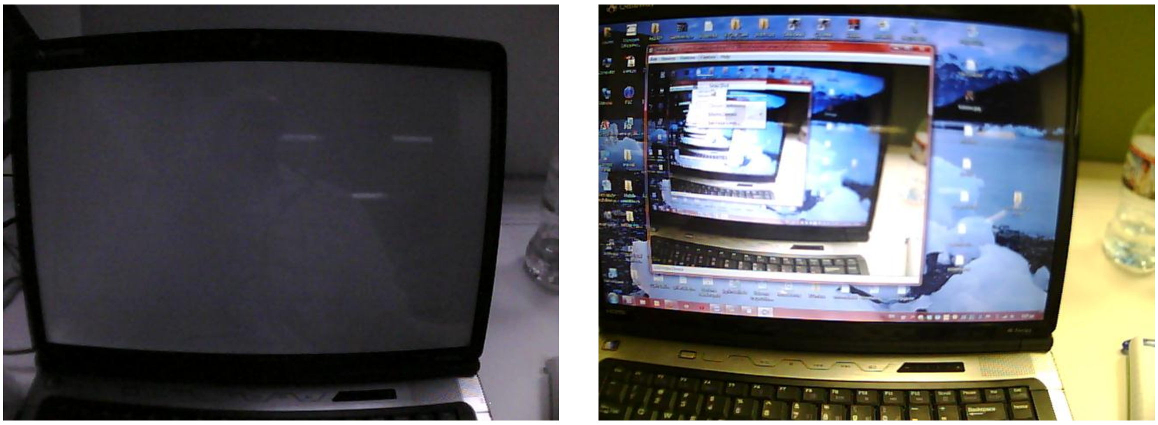


Figure 14: IR Filter functionality over laptop screen

Moreover, a final test undertaken before further implementation was to verify whether the projector emits any near - infrared radiation. Such a finding would be a complete failure for the further development of the project. Fortunately, after experiments on the projectors, it was not found to emit any near infrared radiation. Otherwise, the near infrared radiation from the projector would have the same actions on the object like the visible light, because the spread over the whole screen would not be uniform.

Figure 15, show the filter that was used in the project. The label IR760 indicates that this filter allows any wavelength greater that 760nm to pass through and the label 58mm shows the diameter of the filter.

Figure 15: IR760 infrared pass filter

4.1.3 Optical Sensor

All the hardware elements used in the project were combined together to construct a solid optical sensor. A cylindrical shell was adjusted into the camera stand surrounding the camera. The cylindrical shell has a diameter of 58 mm; hence, the IR filter fits in the edge of the case. In addition to that, the IR filter can be easily removed from the case and vice versa without any difficulty. With this straightforward modification, the testing phase was easier in cases were the filter had to be removed from the case to have a true color image capture. Figure 16 shows the optical sensor in its final state.


Figure 16: Project Optical Sensor

4.2 Software

The software for this project was firstly written in a higher level language and at a second stage was written in C. At the first stage, object detection algorithms were written in MatLab. MatLab has special mechanisms to deal with data and especially images. It is commonly used for simulation and it is easier to work with table equations. Every entry data in MatLab is represented as a table, likewise grayscale images are represented as a two dimensional table. In addition, it provides already made routines for image processing. After several simulations of the algorithms and testing the new idea of using near infrared to solve the problem, the project was processed to the second stage. The second stage of software involves C and OpenCV. With the fist stage, the idea with object detection algorithm and near infrared was tested and confirmed that it worked. Performance and few improvements were added to the second stage to archive real time processing.

4.2.1 MatLab Programming

The initial phase of software programming was written in MatLab. At the beginning, a single camera was used to provide image samples. Due to the fact that near infrared has no color information and it is using a single image channel, grayscale images were firstly used. Firstly, the camera setup is written in the code to connect with MatLab. This connection is done through a mechanism that MatLab provides. Also, the color space and image dimensions were set in this part of the code. The image dimensions that were used was 640 x 480.Furthermore, the background image is created. A series of ten frames are captured and then the average of each pixel value is computed to create the background mode. This is done to ensure that any small alternation to pixel intensity will not affect the background model. Thereafter, a new frame is captured. Each absolute pixel value from the new frame is subtracted from the corresponding pixel in the background model.

Since the representation of black is 0 and of white is 255, all the differences are marked with shades of white. Absolute subtraction is used to show all the possible differences of the two images. With single subtraction, if a brighter spot is subtracted from a darker would result in no difference, because the negative values are counted as zero. A new image is then created to show the differences between the background and current frame. Subsequently the *differences image* is thresholded.

Thresholding is the procedure where the differences image is converted to binary image. In this procedure, each pixel of the differences image is compared to a value. If the pixel value is greater than the threshold value, then the specific pixel is characterised as foreground and is marked in a new image with black color. All the pixels that are characterised as background are marked in the new image with white color. The new image can be identified as *mask* image because that image at the end of the whole process would be the dynamic mask.

After thresholding and the creation of the mask image, noise has to be removed. A series of morphological filters are applied to the image. Firstly, a structure element is created. The structural element will pass through all the mask image and any change to the pixel values, will be applied based on the disk structure element. For noise removal, two filters are used, the *open* filter and the *close* filter. In the case of open filter, an erode operation is done and then a dilation [4]. In the case of the close filter, a dilation is firstly applied and an erosion following that.

Dilation is the operation that is done over an image with the use of a structural element, or otherwise called *kernel*. Often, the structural element can have any shape or size. In most of the cases, the element is a small square or a disk and has a single predefined anchor point, where it is frequently the centered pixel. For the purpose of this project, the structural element is a small disk with size 7 pixels long. The element passes over the image and computes the maximum pixel values contained. The image pixel value is replaced at the position of anchor point with the maximum local value. In other words, it is the computation of the local maximum of the pixels that are overlapped by the structural element. This has as a result the growth of white areas within the image. Erosion is the inverse operation. In the case of erosion, a local minimum is computed and it has as a result to increase the dark areas in an image.

With the open filter performing erosion at first, any small bright areas are removed. The removed white pixels are mainly noise due to camera distortion. Subsequently, dilation is performed to increase the bright area, which is the detected area, and is thus containing the objects. The white pixels are the foreground of the image and increasing them improves the final shape of the mask. A close filter is applied to the mask image to improved and smoothen its shape.

Finally, the mask image is applied on top of the slide that is going to be projected next. In MatLab, the mask image is applied to the frame that has been captured. MatLab was only used to simulate the system functionality and for testing, hence there was no need to create a function that produces the modified slide.

Figure 17 shows the output obtained from MatLab. Starting from the top left corner, the background image is shown, until the bottom right corner, a series of images show all the steps of the system's functionality. In this figure, it is clearly shown how morphological filters are smoothing the detected foreground.

Figure 17: Samples of MatLab results

4.2.2 Algorithm Representation

With the use of a flowchart, tables and images, the system functionality is represented. The following flowchart shows the main loop and the setup step. In addition, the optical sensor has to be placed on top of the projector. Once the system has beet setup, the main loop runs until the ESC key is pressed. In C application, background reading step and the area of interest step can be reset while the loop is running. However, starting values are set to allow the application to start.

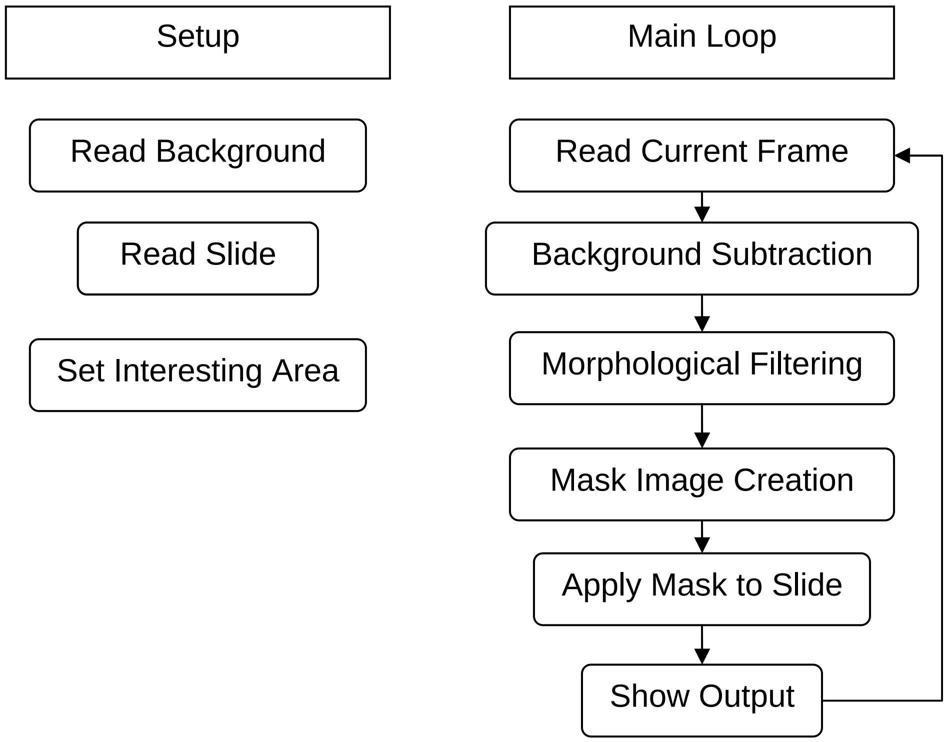


Figure 18 shows the background model. At the left is the data representation and at the right is an actual image captured during testing. They do not have any relation between them. The values inside the table have been randomly selected as an example. Real images are put to show the visual representation of each algorithm step. Each cell of the table represents a pixel in a real image. Values near 255 correspond to a shade of white. As it is common for the projection screen to have a white color, values near to that are selected for the example. In addition, due to the use of the IR filter, highest white values are sensing less.

Figure 19 illustrates the current input frame. The green color represents the new object in the scene. On the right corners, there are two pixels with different color to represent noise in the current frame. The new object has pixels values less than the previous one at the same position. This is how a new object is sensed in terms of the data. In a real life example, human hair commonly has darker color than the human face and the same applies to human eyes. However, human eyes tend to have a shiny color when observing within the near – infrared range. Eyes, as found in experiments, reflect the near-infrared light more than the human skin. Nevertheless, that shining dot does not affect the system functionality.

210	190	210	200	240	190
230	210	190	230	210	250
200	190	200	220	200	250
190	210	200	210	210	240
200	210	220	240	230	220
220	210	240	200	220	190

Figure 18: Background frame

210	190	210	200	240	160
230	210	190	230	210	250
200	130	140	130	200	250
190	110	120	130	210	240
200	210	220	240	230	220
220	210	240	200	220	100

Figure 19: Current frame

The absolute subtraction step is shown in Figure 20. Each pixel of the current frame is subtracted from the corresponding pixel from the background model. Same pixels after subtraction as expected, have zero value, thus the detection is shown in shade of white in real image example. Darker colour hairs are shown to have the highest differences compared to the background model. In contrary, the shirt is almost the same colour with the background and thus the detection to that area complicated. Fortunately, the skin colour is in the middle and after subtraction, pixel have values high enough to be processed correctly.

0	0	0	0	0	30
0	0	0	0	0	0
0	60	60	90	0	0
0	100	80	80	0	0
0	0	0	0	0	0
0	0	0	0	0	90

Figure 20: Image differences

Figure 21 shows the result of thresholding step. Threshold value for the table examples was assumed to be just over 30, thus the noise pixel in the right top corner was eliminated. Moreover, the resulting image is in binary form and white pixels are represented with the value of one and black pixels with the value of zero. This step does not manage to remove the noise pixel at the right bottom corner, which is shown with different colour, due to high value that the pixel used to have. Nevertheless, the mask image is at the beginning of its final form. Normally, the foreground object has to represented with white colour while the background with black. An inversion has already been performed to the image due to the fact that in mask image the foreground is more useful to be marked as black.

As expected in the real image example, the area under the head, which includes the neck and shirt, is considered as background. The edges of the black objects are rough and there are also some small defects in the inside area.

1	1	1	1	1	1
1	1	1	1	1	1
1	0	0	0	1	1
1	0	0	0	1	1
1	1	1	1	1	1
1	1	1	1	1	0

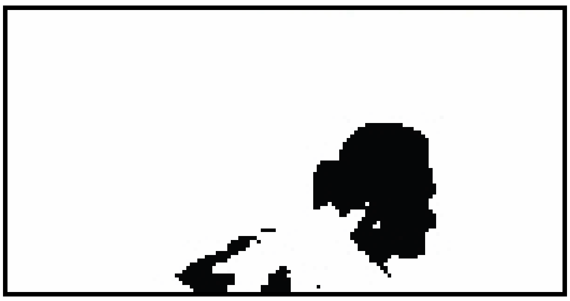


Figure 21: After threshold

The results of the morphological filtering step are also illustrated. A structural element, or kernel, of three pixels in row shapes is used in this example which is shown in Figure 22. With lighter red colour, on the left top corner, the kernel is shown processing the second pixel of the array. The final position of the kernel is shown with red colour at the right bottom corner. At that position, the remaining noise pixel will be processed and due to the filter functionality will be removed. The image at the right which shows the real image example, illustrates the changes of that step in the black object. The rough edges have been smoothed and the small white dots inside the object have been removed. At the end of this step the mask image is in its final form.

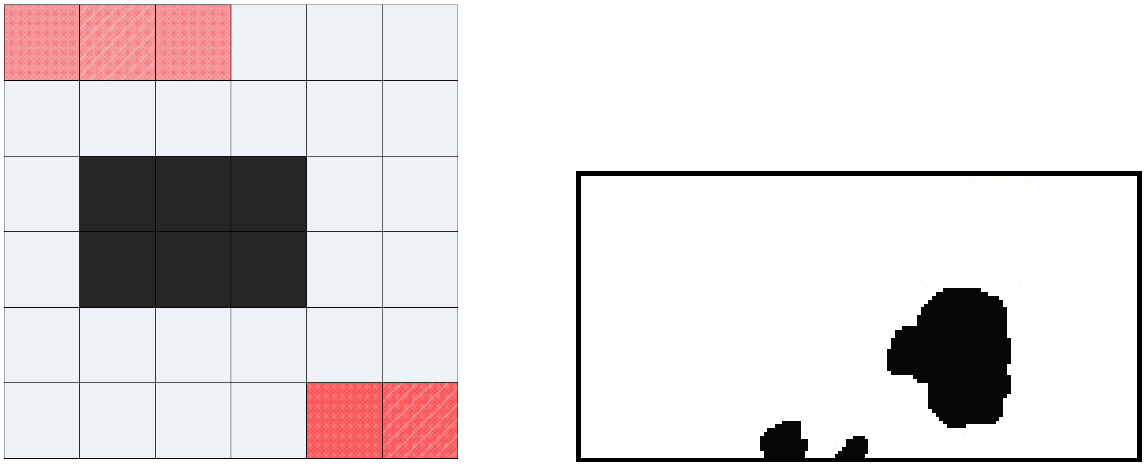


Figure 22: Morphological filtering

Figure 23 shows the final resulting image of the system. Given the mask and a slide image created previously, the system combines the two images and produces a final image which is shown at the left. The mask image is scanned and when a back pixel is found, it is copied in the corresponding position in the slide image. The mask image acts like a dynamic filter which is created with every captured frame and blocks the projection of light within the black area. The right image shows the output of MatLab, which simulates the ground truth after the projection of the final image which the system has created.

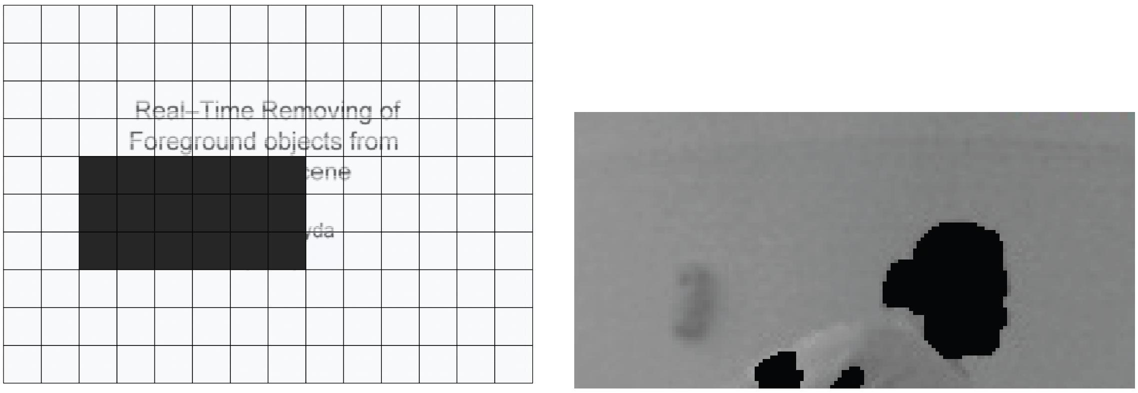


Figure 23: Output

4.2.3 C Programming

After the first implementation in MatLab, the algorithm was converted to C code. In addition OpenCV library was used. The purpose of the conversion was to give performance to the application and make it real time. Comparing the two, C applications perform better with the same quality results. In addition, C code has some extra fixtures to adjust the image mask with the presented slides.

The same methodology is used in C code to read the background image as a first step. Furthermore, in C application a *menu* window is shown to give to the user an interaction with the process. The background reading can be done more than one times, with the use of a button. This addition helps in situations that the camera has moved from its initial position and the background image has to be recreated.

The presentation slides are read by the applications in image form in the setup phase. In addition, at the same phase a small window in the image is created. With the use of the window, the area of the projection in the screen is set. The window shows that only the projected area is important and all image process is done only within the window's margins. This allows less processing time per frame, due to fewer pixels in frame, hence greater performance.

Figure 24 shows the menu window. At the top is the button, in track bar shape for background reading. The next track bar allows the user to adjust the threshold value. As the figure shows, a lot of noise is detected as foreground. The fact that noise is outside the area of interest, it is excluded to be cleaned in the subsequent step. Inside this area, the noise is cleared, improving in this way the performance of the system.

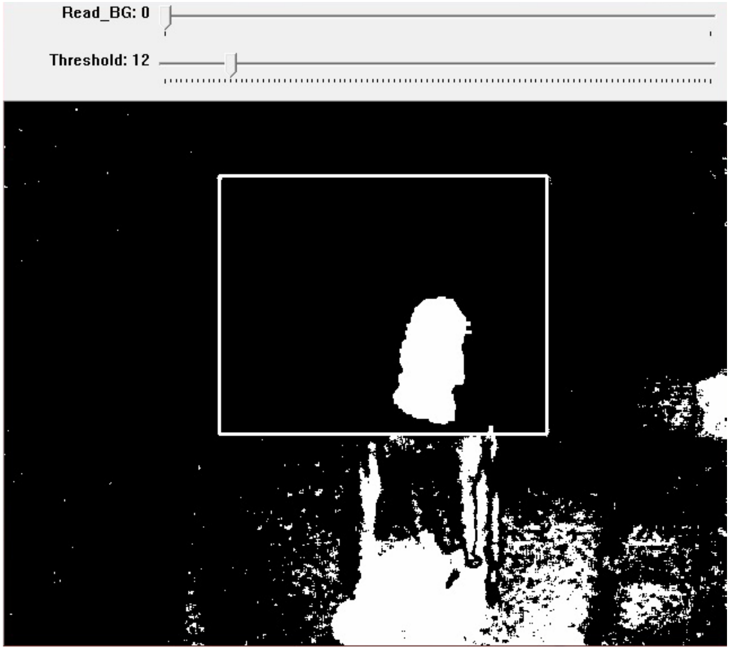


Figure 24: Menu window

In OpenCV, this window is called ‘Region Of Interest (ROI)’ and can be set to any image. In C application, ROI can be set by left clicking on the left top corner and by right clicking on the right bottom corner of the projection screen. Figure 25 shows the ROI window as it is set to enclose the projected area. The left image is grayscale and has been captured without the IR filter to help set the region of interest. On the image at the right, the ROI window is shown as the camera has the IR filter. In the setup phase, all the preview windows and memory space for images are created and connection with camera and slides size adjustments, are performed as well.

Figure 25: Presented area window. (ROI)

Moreover, the algorithm function is called. In that function, the absolute subtractions between the current frame and the background image is done, creating a new image which will be the mask image at a later step in the procedure. A dilation filter is applied over the resulting image to increase the foreground detected area.

The thresholding step comes next to create the binary mask image. Lot of noise pixels are cut out within that step. Afterwards, another function is called to perform more morphological filters and combine the slide image with the mask. The threshold value is adjustable by the user, in order to have better control over the application and to be able to regulate it better to any environment conditions.

The *ApplyMask* function, resizes the mask image to the size of the slide. Again, only the interested area of the mask image is resized and then a dilation filter is applied to smooth any defects that are left over from resizing. Moreover, the mask image is combined with the slide with a procedure that scans the mask image for black pixels and copies them at the same location in the slide image. Then, the slide image contains a solid black shape that refers to the object that has been detected. Finally, the slide is presented and the light is removed from the presenter. Figure 26 shows how the system affects the presentation. At the left image, the system was turned off and the light from the projector goes to presenter's eyes. The right image was captured after turning the system on and shows that no light goes to presenter eyes.

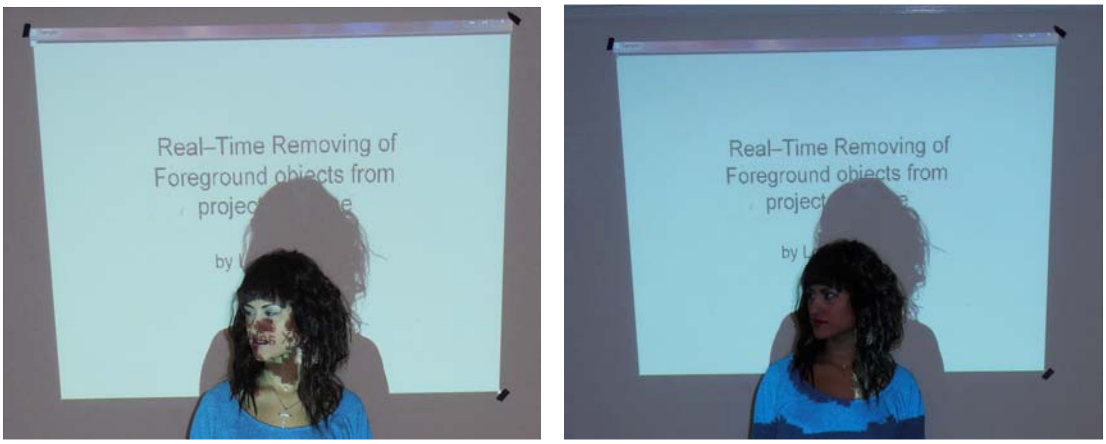


Figure 26: Presentation with and without system running

The execution speed time of the C code is approximately 55ms per frame or 18 frames per second, which is fast enough to be characterised as real time application. The speed has been measured after running the code on a laptop, with two core CPU running at 2GHz. The images from the camera were 640 x 480 pixels.

C application can be seen in Figure 27. At the left is the *menu* window, which has the background reading and threshold track bars. Also, in *menu* window the results of the background subtraction algorithm are shown. With white color are the points in the current frame that are different from the background image. Inside the rectangle, which is the region of interest, is the mask image after noise removal and smoothing. The area outside the rectangle shows the detected foreground without performing any process on it. The current frame captured by the optical sensor, is illustrated in the *CrayScale* window, which is the image in the middle of the figure. *CrayScale* is the window that ROI can be set by left clicking on the left top corner and by right clicking on the right bottom corner. The *Sample* window is located on the right which shows the output of the system. The combination of the slide image, which has been loaded at the setup phase, and mask images is illustrated at that window. Furthermore, the projector has to project the sample window to have the desired results.

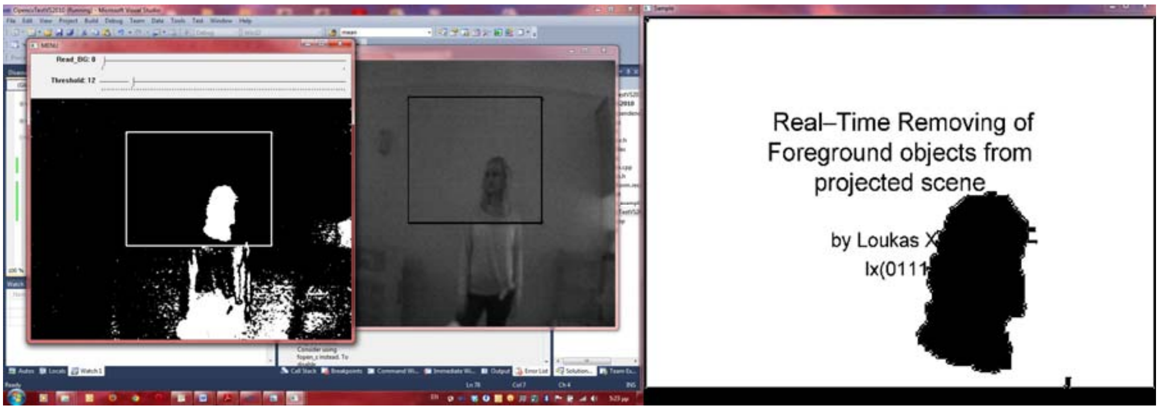


Figure 27: Application in C

5 Tests and Results Analysis

The system was tested in terms of detection and covering with different people. The aim of those tests was to check if the system functions correctly with different people in terms of their skin color. In addition, people wearing glasses were also used in test, checking if this alternation affects the functionality of the system. The system was also tested with placing the camera in different distances from the screen. The results of those tests are analysed and presented in the rest of the chapter.

5.1 Room Environment

Tests were performed in an environment similar to the lecture room. The background screen was white to represent a real projection screen and the light was at the same level as in the presentation. Environment's infrared light was used. All the test were taken under the same room environment.

5.2 Distance Test

The distance tests were performed to check the system's functionality within different distances between the optical sensor and the objects. The testing distances cover the range of two to four meters. These distances are chosen to be tested because are common distances between the projector and the screen. Three images are captured and processed per checkpoint. The system passed successfully the tests, managing to correctly detect and cover the person's eyes in all situations. Moreover, in all the tests the system managed to cover the head of the person, hence it functions perfectly for the distances between two and four meter. The results for that test were as expected

Figure 28,Figure **29** and Figure **30** shows the captured data from that test. For each distance test, one of the three captures is shown. As it is clearly shown, the system covers the head of the person in all the images. The threshold value was set to 15 and the structure element size was six. In addition, the steps of the procedure are shown with the usage of six images. In the left upper corner the background frame is illustrated, which corresponds to the background model.

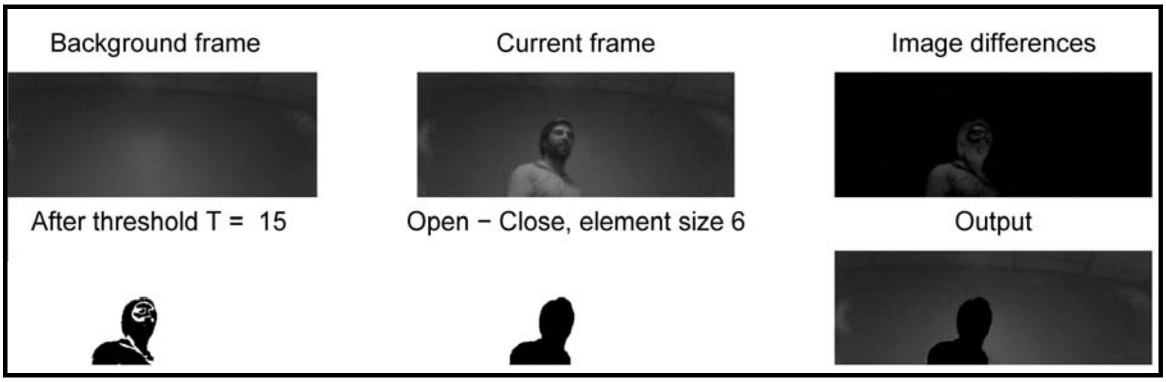


Figure 28: Two meters test

The upper middle image shows the current captured frame and at the right of the first row is the image that contains the result of absolute subtraction. Then, the mask image is initiated. The bottom left image shows the resulting image after the threshold step.

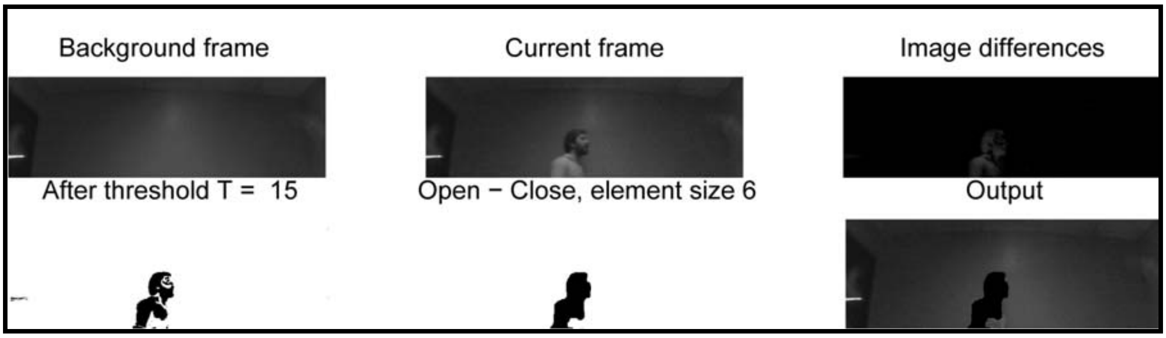


Figure 29: Three meters test

The image in the middle of the second row shows the mask image after being processed by morphological filters. The foreground object has a solid shape with smoother edges. The last image at the right shows the output of the system. The system manages to detect and cover the whole head of the person.

Furthermore, in Figure 30, the system's reaction to noise is illustrated. As it is shown in the threshold image, which is located at the left in the second row, a square shape object is seen above the person. That noise is removed in the next step, while the only thing remaining is the person's mask.

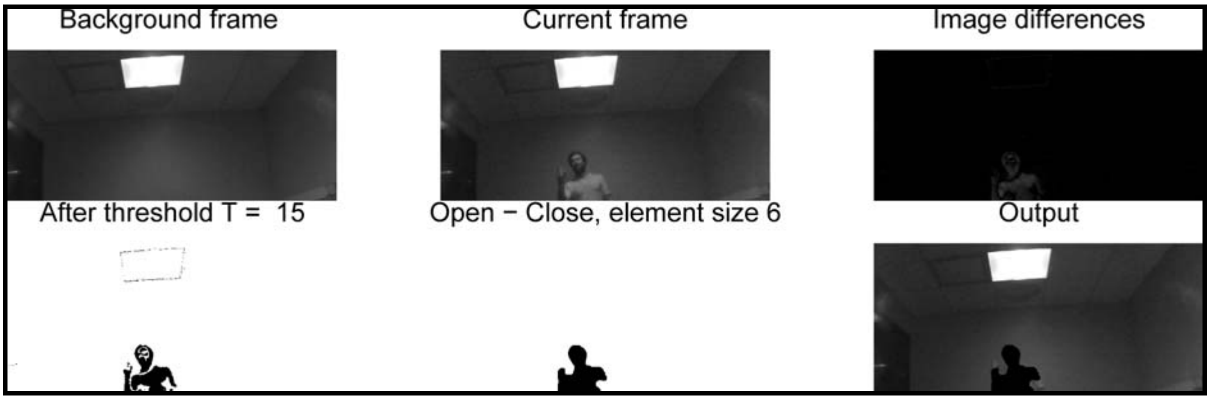


Figure 30: Four meters test

5.3 Front Face Test

For the purpose of that test, participants were asked to look at the audience direction simulating the most common position of the presenter. The room environment was the same as previously taken tests. Moreover, the test was subdivided into two major parts. The first part was about marking the object from two different distances and the second part, aims to examine people's skin colour. For the distance test part, the distances of two and four meters, which referred as near and far, are selected to be observed. Participants are classified based on their skin colour from one to three, in which one corresponds to brightest, and three to darkest colour. Furthermore, two regions of the head are set to be the interested area; eyes and the whole head, with eyes to be the most important spot. A sample to be calculated as successful has to firstly cover the eyes area and then the whole head area. After analysing the result, system is shown to function perfectly to near distances with 100% detection in eyes' area. As it is also shown in Table 1, the detection for the head area at the same distance was 80%. However, the detection ratio for the far distance was lower, limiting to 71% to eyes area and just 43% to the head area. A possible reason for the low detection ratio can be the low amount of infrared light at the tests time. In addition, given that at near distances the ratio is high enough, a camera with greater resolution, from the one that was used, is possible to have higher detection ratio.

Distance Test	Eyes detection ratio	Head detection ratio
Near	100%	80%
Far	71%	43%
Total	86%	62%

Table 1 : Front face test, Distance test detection rates

Results from the front face detection test are shown is Figure 31. On the left hand side image, the captured frame can be seen. Next, at the right of each row the mask and the final output of the system is shown. The last row of images is an example of successful detection in eyes area but unsuccessful detection on the head. Overall, for this part of the test, the system managed to gain 86% detecting eyes area and 62% for head's area detection.

Figure 31: Results from front face test.

As already mentioned, the people that participated on the test have been labeled with a value from one to three that corresponds to the skin tone. Once the results had been analysed, the following table was produced. Table **2** presents the results taken from the skin color test. As it is clearly shown, detection ratio for people with darker color is higher. The same applies to people within the middle skin tone. For brighter skin colour the detection ratio is lower as they match more with the screen colour

Skin Colour Test	Eyes detection ratio	Head detection ratio
Colour 1 (brighter)	71%	42%
Colour 2	100%	60%
Colour 3 (darker)	100%	100%
Total	90%	67%

Table 2: Front face test, Colour skin test detection rates

At the head detection column, the results were 42%, 60% and 100% from bright to darker skin tone respectively. In general, the detection ratios for this part of the test were 90% for the eyes area and 67% for the head area.

5.4 Side Face Test

For the side detection test the participants were asked to look sideways, in order to simulate the situation where the presenter tries to show something on the screen. Moreover, the final head position that the project needs to operate is completed successful and therefore at any angle beyond the final operating angle, the detecting procedure does not need to be processed correctly by the system. Following the same procedure as previous tests, the working environment conditions were the same and the side face test was divided into two major subparts; distance test and skin colour test. Participants maintain the same label for skin color as the checkpoint at two and four meter distance from screen is also observed. The same evaluation criteria were followed as the interested regions were continued to be the eyes and head. As it is shown in Table **3**, the distance is seen not to affect the detection ratio. On the other hand, detection ratio in the head area observed to be low. This does not affect the system functionality, thus the main reason for low ratio is commonly small holes or defects in mask head shape. The important eyes area was fully detected in this test at the same time as in total for head's area the ratio was 60%.

Distance Test	Eyes detection ratio	Head detection ratio
Near	100%	57%
Far	100%	63%
Total	100%	60%

Table 3: Side face test, Distance test detection rates

Furthermore, in the skin color test, as shown in Table 4, eyes area has fully detected correctly. Once again, the head's area detection ratio was observed to be low, resulting in 40% for brighter skin colours and 57% for middle skin tone. The system's total ratio at the head's area detection has increased after detecting correctly all the samples from people with darker skin colour.

Skin Colour Test	Eyes detection ratio	Head detection ratio
Colour 1 (brighter)	100%	40%
Colour 2	100%	57%
Colour 3 (darker)	100%	100%
Total	100%	66%

Table 4: Side face test, Colour skin test detection rates

In Figure 32, results are illustrated for the side face test. It is clearly shown why the head’s area detection has inadequate results. Four of the six output results observed to have defects in the shape of the mask. In addition, the significant value of morphological filtering in improving cover mark can be seen in the second and third column.

Figure 32: Results from side face test

5.5 Two People in Screen Test

In this test, the performance of the system was measured when two people are appearing in the screen area. The distance where the frames were captured was fixed at three meters. The participants differ in skin colour, with the one having brighter skin tone than the other. Again, the environment was the same as the other tests and the same evaluation method was applied. The participants were free to look in any direction thus, to simulate actual head movements which take place in a presentation. Table 5 shows that the system managed to get an 80% detection ratio overall in the eyes area and 70% in the head area. The detection in head’s area is slightly better for darker skin colour than brighter ones.

For brighter skin color an average of 60% detection ratio was achieved where in darker skin colours the ratio was 80%

Skin Colour Test	Eyes detection ratio	Head detection ratio
Colour 1 (brighter)	80%	60%
Colour 3 (darker)	80%	80%
Total	80%	70%

Table 5: Results from two people test

5.6 Free Moving Test

The last test was performed to evaluate the system functionality when participants move freely in the room and act naturally like in a real presentation. The distance between the screen and the camera was set to four meter. For this test, the different skin colors were not taken in account. Moreover, the same room environment and assessment method as for all other tests was carried out. The system is shown to perform well with 90% detection ratio in the eyes area, while in the head's area the detection ratio was 67%.

5.7 Overall Detection Ratio

Table 6, illustrates the overall arithmetic results that were measured after the system had successfully completed several tests. The overall marking for the system detection ratio is 94%, with the most important being the eyes area.

Test	Eyes detection ratio	Head detection ratio
Front face test	89%	78%
Side face test	100%	60%
Near distance	100%	100%
Far distance	87%	53%
Free moving	90%	67%
Two person	80%	70%
Overall samples	94%	69%

Table 6: Overall detection ratio results

The resulting tests were as expected. Higher detection rate to side face tests than to front face test was anticipated. The eyes area is greater when a person is facing the camera, thus possibilities for fault in detection are higher. Therefore, detection ratio on side face test achieved the maximum possible.

Moreover, the darker skin colors seem to have an advantage over brighter in terms of system performance. The lack of colour, due to IR filter, turns to complex detection when the person's skin colour is brighter. Commonly distinguishing between background and foreground in such situations are difficult and thresholding with a proper value is crucial.

On the other hand, the lower ratio at far distances detection appears to be an effect of low amount of infrared light. Furthermore, the standard camera's resolution shows to be another factor that affects the quality of the results.

For greater distances, image pixel tents to cover more area on the person body than in shorter distances. Thus, images captured from a distance, contains distorted information about shapes and colours. Decreasing the size of structural element will partly improve the detection ratio. A camera with greater resolution will significantly improve the performance of the system, not only at long distances, but also short. However, the cost was set to the lowest possible, thus to have a solution available to everyone.

6 Objectives Evaluation

The project has achieved its objectives successfully. Experiments have shown that the algorithm successfully detected 94% of the tested situations and the presenter's eyes have been covered. Moreover, with the appropriate working environment study, the implementation of such a complex object detection algorithm which will decrease the performance of the systems was prevented. Furthermore, with the addition of near infrared imaging fixtures, the simplification of definition and implementation of the background model has been enabled, maintaining the quality of the results.

The application has been tested on a laptop machine with two cores, running at 2 MHz, two Giga Bytes ram memory with 32-bit windows operating system. The result has been measured to be 55ms per frame. This speed provides approximately 18 frames per second and thus the applications can been characterised as real time. Hence, the previously mentioned system specifications set the minimum system requirements.

7 Conclusion

The system manages to achieve its aim and works in real time with quality results. It produces images to project with the rate of 18 frames per second.

However, improvements in the algorithm and to the entire system can be done. Experiments show that the system performs better if the infrared light is opposite to the object. With a self illuminator infrared source, the capabilities of the system can expand and increase the possibility of working in total dark conditions. Furthermore, the use of the IR filter, introduces a new way of background image creation and sensing for such applications. Hence, a review can be examined in running systems.

Moreover, an intermediate device which will act as the computer can be introduced. The new device will release the computer from the effort to process all the data and thus will increase the outcome rate of the system. Such a device is a signal processor unit. In addition, the portability of the system will increase, making it possible to run on more operating systems. The addition of the new *processing* device can be combined with the appropriate code to fit on the machine.

What's more, the solution approach can be altered in terms of the algorithms that were used. In a more powerful system, a face detection algorithm would prove to be a better methodology to follow.

8 Reference

[1] P. Cooper and K. Ideus, “Attention Deficit / Hyperactivity Disorder, A practical Guide for Teachers”, David Fulton Publishers, London, 1996.

[2] Robert J. Marzano and Jana S. Marzano. “The key to classroom management, Educational leadership,” 2003, Volume 61 | Number 1 Building Classroom Relationships Pages 6-13

[3] Gonzalez, C.R and Woods, E. R, “*Digital Image Processing*”, Pearson Education Inc, New Jersey, USA, 2008

[4] Bradski, Gary, and Adrian Kaehler. "Image processing." Learning OpenCV: computer vision with the OpenCV library. Sebastopol, CA: O'Reilly, 109 – 144, 2008.

[5] P. Viola and M. Jones, “Robust Real-time Object Detection” Second International Workshop on Statistical and Computational Theories of Vision – Modeling, Learning, Computing, and Sampling, Vancouver, Canada, July 13, 2001

[6] Nascimento, J.C. Marques, J.S. "Performance evaluation of object detection algorithms for video surveillance," Multimedia, IEEE Transactions on , vol.8, no.4, pp.761-774, Aug. 2006

[7] Mariano, V. Y. Junghye Min Jin-Hyeong Park Kasturi, R. Mihalcik, D. Huiping Li Doermann, D. Drayer, T. "Performance evaluation of object detection algorithms," Pattern Recognition, 2002. Proceedings. 16th International Conference on , vol.3, no., pp. 965- 969 vol.3, 2002

[8] Wren, C.R. Azarbayejani, A. Darrell, T. Pentland, A.P. , "Pfinder: real-time tracking of the human body," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol.19, no.7, pp.780-785, Jul 1997

[9] Stauffer, C. Grimson, W.E.L. , "Adaptive background mixture models for real-time tracking," *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.* , vol.2, no., pp.2 vol. (xxiii+637+663), 1999

[10] Haritaoglu, I. Harwood, D. Davis, L.S. "W4: real-time surveillance of people and their activities," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.22, no.8, pp.809-830, Aug 2000

[11] Bradski, Gary, and Adrian Kaehler. "Projection and 3D Vision ." Learning OpenCV: computer vision with the OpenCV library. Sebastopol, CA: O'Reilly, 405 – 459, 2008.

[12] Beymer D. and Konolige K.;, "Real-Time Tracking of Multiple People Using Continuous Detection" Artificial Intelligence center SRI international

[13] Kurt Konolige. "Small vision systems: hardware and implementation." In Eighth International Symposium on Robotics Research, pp. 111-116, Hayama, Japan, 1997

[14] Eveland, C. Konolige, K. Bolles, R.C. "Background modeling for segmentation of video-rate stereo sequences" Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on, vol., no., pp.266-271, 23-25 Jun 1998

[15] Eric C. Newgard. "Near-Infrared Spectroscopy for Analysis of Agricultural Material" Illinois University, 2005 <http://online.physics.uiuc.edu/courses/phys598OS /fall05/FinalReportFiles/EricNewgardTermPaper.pdf>

[16] "Infrared." Wikipedia, the free encyclopedia. <http://en.wikipedia.org/wiki/Infrared#cite_note-0>.

[17] "My NASA Data Electromagnetic Spectrum." My NASA Data Home Page. <http://mynasadata.larc.nasa.gov/ElectroMag.html>.

[18] "Infrared spectrum image" Wikipedia, the free encyclopedia. <http://en.wikipedia.org/wiki/File:Infrared_spectrum.gif#file>

[19] " Item detail page." Welcome to langtoninfo.co.uk: Online retailer. <http://www.langtoninfo.co.uk/showitem.aspx?isbn=0188218000842>.

[20] Vincent, E. Laganiere, R. "Detecting planar homographies in an image pair", Image and Signal Processing and Analysis, pp 182 - 187 2001.

[21] Kanade, T. Yoshida, A. Oda, K. Kano, H. Tanaka, M. "A stereo machine for video-rate dense depth mapping and its new applications," Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on, vol., no., pp.196-202, 18-20 Jun 1996

[22] Frantz, G. "Digital signal processor trends," Micro, IEEE , vol.20, no.6, pp.52-59, Nov/Dec 2000

[23] Labayrade, R. Aubert, D. Tarel, J.-P. "Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation," Intelligent Vehicle Symposium, 2002. IEEE , vol.2, no., pp. 646- 651 vol.2, 17-21 June 2002

[24] Ying-Li Tian Lu, M. Hampapur, A. "Robust and efficient foreground analysis for real-time video surveillance," Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol.1, no., pp. 1182- 1187 vol. 1, 20-25 June 2005