CHAPTER TWO

INTRODUCTION TO IMAGE SEQUENCE PROCESSING

2.1 Introduction

Before applying the computer vision operations to the image data, it is usually necessary to process the data in order to assure that it satisfies certain assumptions implied by the particular operations. For example re-sampling the next data to assure that the image coordinate system is correct and noise reduction in order to assure that sensor noise does not introduce false information.

This chapter is dedicated to illustrate the essential preparation work in order to fulfill the requirements of the proposed system. So, it starts by explaining the representation of a digital image followed by defining the video signal. Consequently, this chapter delivers a detailed description of the background subtraction method, binary image manipulations, the constraint tracking approach implemented in this study and conclude by proving a brief introduction about the concepts and the current directions of visual surveillance.

2.2 Digital Image Representation

Digital images is considered as a primary unit in every computer vision system and it defined as a set of discrete, finite values of amplitude mapped in a discrete, finite set of coordinates [8]. From the definition, it can be said that a digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as picture elements or more commonly pixels. The processing of digital images is known as digital image processing by means of using a digital computer to process the images with some form of storage media such as a computer's memory or on a hard disk or CD-ROM. Usually this types of images produced by one or several image sensor such as a various types of light-sensitive cameras. The pixel values typically correspond to light intensity in one or several spectral bands (gray images or color images). The next section presents the the representation of a digital image.

A common way to describe a digital image is a two-dimensional discrete signal. Mathematically, such signals can be represented as functions of two independent variables—for example, a brightness function of two spatial variables. A monochrome digital image $f(x, y)$ is a two-dimensional array of luminance values.

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & ... & f(0,N-1) \\ f(1,0) & f(0,1) & ... & f(0,N-1) \\ . & . & . & . \\ f(M-1,0) & f(M-1,1) & & f(M-1N-1) \end{bmatrix} \tag{2.1}$$

The right side of equation (2.1) is by definition a digital image. Each element of this matrix array is called a picture element, or a pixel [9]. In other hand a color digital image is typically represented by a triplet of values, one for each of the color channels, as in the frequently used RGB color scheme. The individual color values are almost universally 8-bit values, resulting in a total of 3 bytes (or 24 bits) per pixel. This yields a threefold increase in the storage requirements for color versus monochrome images.

$$f(x,y)=\begin{bmatrix} RGB_{0,0} & RGB_{0,1} & ... & RGB_{0,N-1} \\ RGB_{1,0} & RGB_{1,1} & ... & RGB_{1,N-1} \\ . & . & . & . \\ RGB_{M-1,0} & RGB_{M-1,1} & ... & RGB_{M-1,N-1} \end{bmatrix} \tag{2.2}$$

In the color-interleaved format, the color information is separated into three matrices, one for each of the three color channels. The RGB color scheme is just one of many color representation methods used in practice [10, 11]. The letters R, G, and B stand for red, green, and blue, the three primary colors used to synthesize any one of 2^{24} or approximately 16 million colors. Equal quantities of the three color values result in shades of gray in the range of 0 - 255.

2.3 Video Signals

A video is a sequence of images displayed at a certain rate or frequency. National Television System Committee (NTSC) video (the standard for analog video followed in the U.S.) displays 30 images (called frames) per second. Human eye is unable to distinguish between successive images displayed at this frequency, and so a TV broadcast appears to be continuously varying in time [12]. Figure 2.1 illustrates the structure of a video signal captured using 30 frame per second as capturing device frame rate.

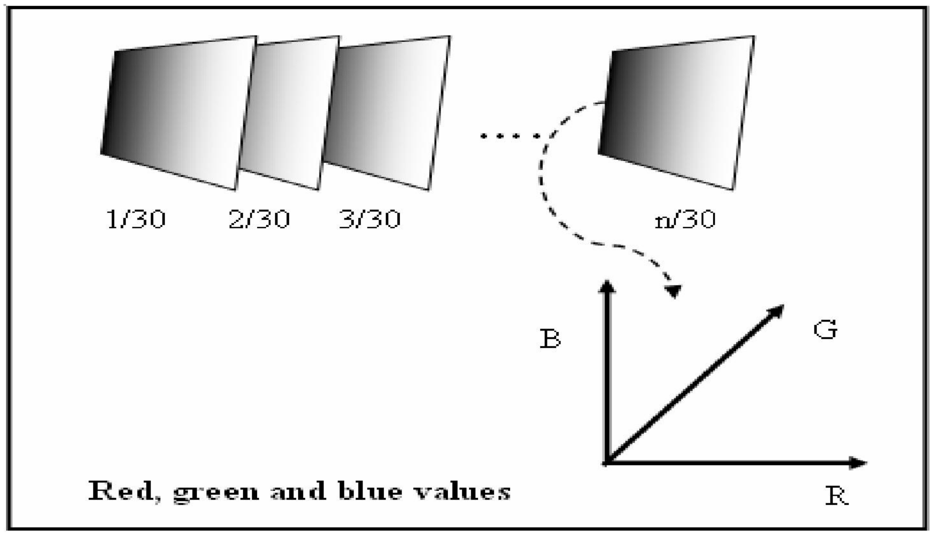


Figure 2.1: Video Signal structure [12].

2.4 Tools and Equipments

This section is devoted to illustrate the tools and equipments implemented in this study. The capturing device used in this work is a fixed camera type MICROSOFT VX1000-1.3 Mega pixel webcam. This camera is used in the process of capturing the video samples, and accompanied by Matlab 7.0.4 as aided software to analyze the simulation scenarios. Matlab software installed in TOSHIBA SATELLITE A100 laptop comes with speed 1.73 GHz and the size of it is memory is 1GB of RAM.

2.5 Background Subtraction

A popular method for online segmentation of moving regions in image sequences involves “background subtraction,” or thresholding the error between an estimate of the image without moving objects and the current image [6]. The numerous approaches to this problem differ in the type of background model used and the procedure used to update the model [7, 13, 14, 15]. Given the assumptions, the most obvious approach is to maintain a background image as a cumulative average of the video stream and to segment moving objects by thresholding a per-pixel distance between the current frame and the background image. This method is the foundation of a collection of techniques generally known as *background subtraction* [14, 16, 17]. Simple background subtraction has the advantage of computational speed but fails in uncontrolled environments. The most common problems involve changing illumination levels and temporal background clutter as often found in outdoor scenes. These two problems are usually addressed by building an adaptive background model, so its parameters can track changing illumination which results more accurately representation for multimodal backgrounds [14, 18, 19].

Two different background subtraction methods is used through out this work, the next section express those methods from a technical point of view and presents a sample of results gained from applying these two methods. The first background subtraction method was implemented in [Altahir A. Altahir et al, 2007], and it based on converting each new next frame into grayscale image in order to minimize the computation requirements. The converting process is based on three steps:

- The first step is converting the red green and blue color values to NTSC coordinates.
- Then setting the hue and saturation components to zero.
- Finally returning the image back to RGB color space.

Figure 2.2, presents four images the first two images represents the static reference frame in RGB color space and in grayscale form; the second two images refer to a sample of an original frame and the result of converting this frame into grayscale.

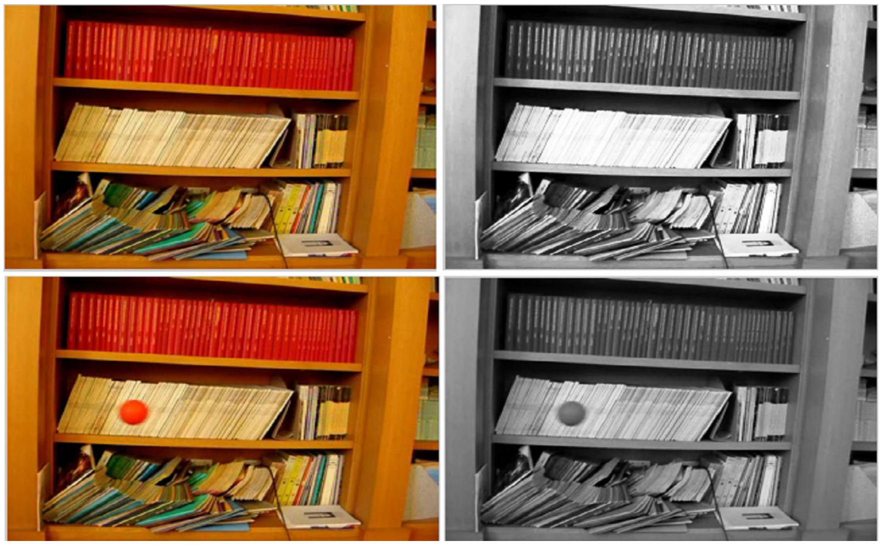


Figure 2.2: Converting the RGB images into grayscale color space.

Then subtracting the results of the grayscale adaptation from a pre defined grayscale static reference frame.

$$I_c(x,y) = I_t(x,y) - B(x,y) \tag{2.3}$$

Where I_t representing the current frame, B refer to a static reference background model and I_c the result of subtraction process. Figure 2.3 shows a sample of 12 successive frames after the subtraction process.

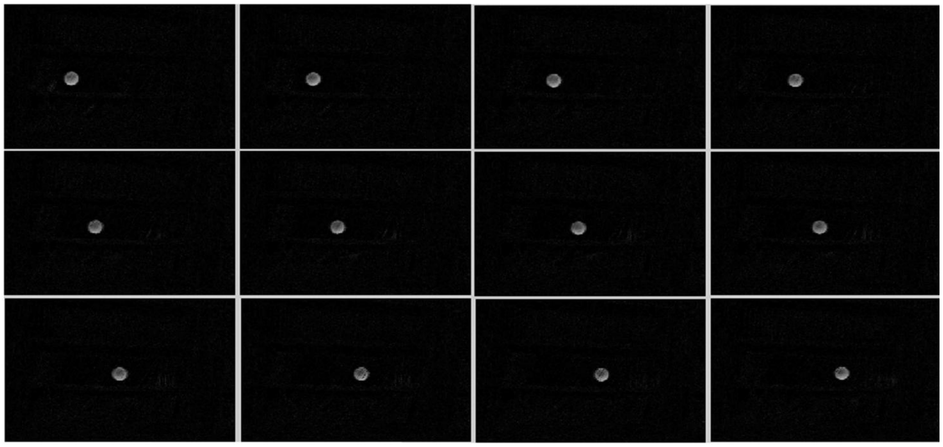


Figure 2.3: A sample of 12 successive frames after the subtraction process.

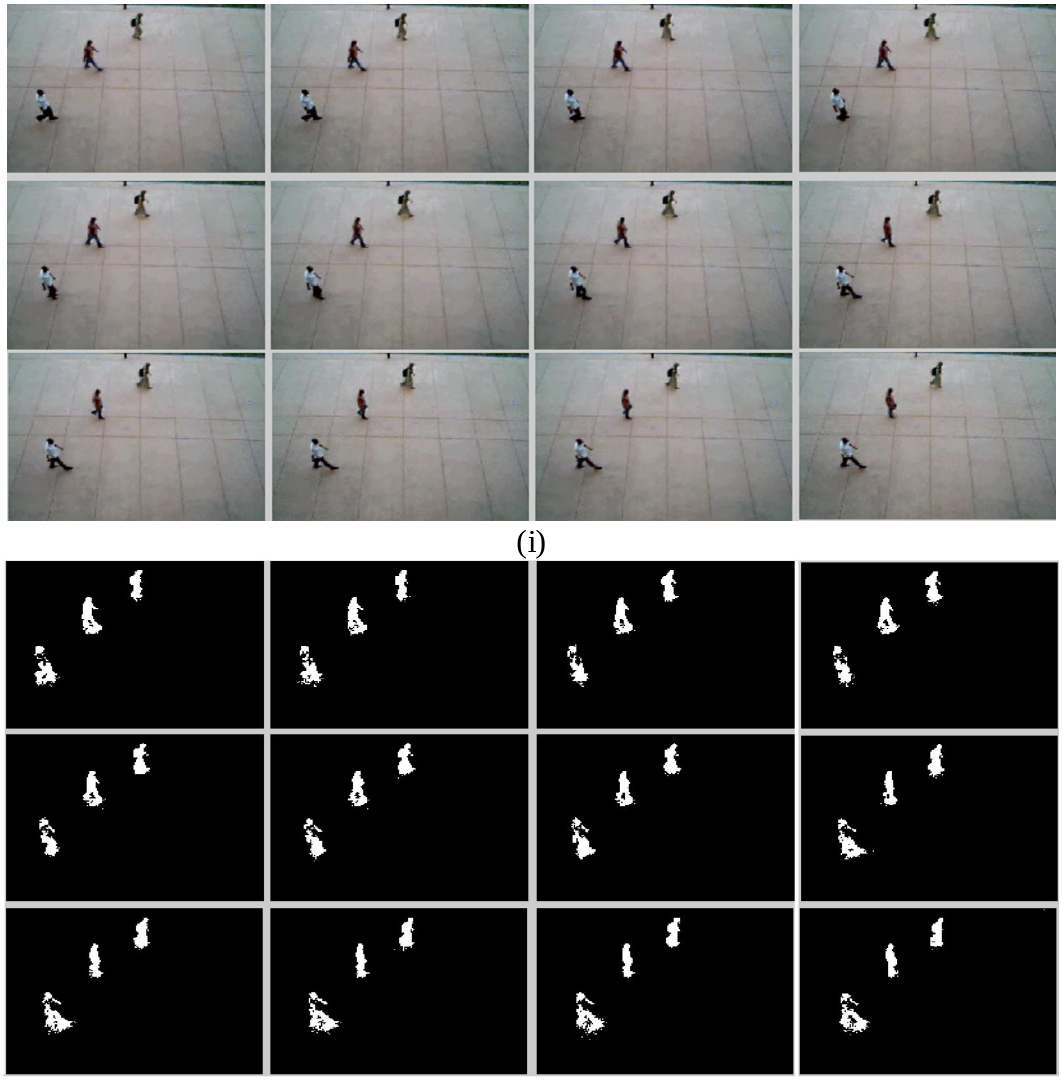
The second background subtraction relied on computing a pixel based absolute difference between each incoming frame I_T and an adaptive background frame B_T after converting them into grayscale color space [6]. The pixels are assumed to contain motion if the absolution difference exceeds a predefined threshold level. As a result, a binary image is formed where active pixels are labeled with “1” and non-active ones with “0”.

$$\left| I_T - B_T \right| > \tau \tag{2.4}$$

Where τ is a pre defined threshold. The thresholding is followed by size filtering and closing with a 3 x 3 kernel in order to discard the small regions, next section will discuss these processes in more details. It is necessary to update the background image frequently in order to guarantee reliable motion detection. The basic idea in background adaptation is to integrate the new incoming information into the current background image using the following first order recursive filter:

$$B(k+1) = \alpha I(k) + (1-\alpha) B(k), \alpha \equiv Adaptation\ Coefficient. \tag{2.5}$$

The larger it is the faster new changes in the scene are updated to the background frame. However, α cannot be too large because it may cause artificial “tails” to be formed behind the moving objects, so α is kept small and the update process based on Equation 3.5 is only indented to adapting the slow changes in overall lighting. The activity of each pixel is monitored during several consecutive frames. The intensity values of those pixels that are active most of the time are directly copied from the latest *I* (*k*) to *B* (*k*). In this way the system can adapt reasonably fast to new static objects appearing on the scene, like stationary cars, or to sudden changes in the illumination level. Figure 2.4 shows the implementation of this method [Altahir A. Altahir et al, 2008a, Altahir A. Altahir et al, 2008d, Altahir A. Altahir et al, 2008e].



(i)

(ii)

Figure 2.4: Background Subtraction (i) Sample of 12 successive RGB frames. (ii) The corresponding binary images after implementing the background subtraction.

The original RGB 12 successive frames are presented beside the binary image version of the same frames after segmenting the moving objects. Although the second method forwards some harmonization between the background and the current frame, however, there are many problems associated with this model. The basic and influential problem in such simple background estimates is tracking the illumination changes, which occur with flashing lights and swaying branches [15]. In this work the background errors is dealt via implementing operations on the output binary image.

2.6 Obtaining and Manipulating the Binary Image

Binary images are also called bi-level or two-level. (The names black-and-white, B&W, monochrome or monochromatic are often used for this concept). It defined as a digital image that has only two possible values for each pixel. Binary images often arise in digital image processing as masks or as the result of certain operations such as segmentation, thresholding, and dithering. Binary images are used in many applications since they are the simplest to process, however they are such an abbreviated representation of the image information. So, the binary images are useful where all the information you need can be provided by the silhouette of the object and when you can obtain the silhouette of that object easily. Some sample application domains include:

- Identifying objects.
- Identifying orientations of objects.
- Interpreting text.

Sometimes the output of other image processing techniques is represented in the form of a binary image, for example, the output of edge detection can be a binary image (edge points and non-edge points). Binary image processing techniques can be useful for subsequent processing of these output images. Implementing the binary image in images analysis has several advantages; the next part discusses the advantage of using binary images [10]:

- Easy to acquire: simple digital cameras can be used together with very simple frame stores, or low-cost scanners, or thresholding may be applied to grey-level images.
- Low storage: no more than 1 bit/pixel, often this can be reduced as such images are very amenable to compression (e.g. run-length coding).
- Simple processing: the algorithms are in most cases much simpler than those applied to grey-level images.

In other hand, the disadvantages of implementing binary images in image analysis are listed as follows [10]:

- Limited application: as the representation is only a silhouette, application is restricted to tasks where internal detail is not required as a distinguishing characteristic.
- Does not extend to 3D: the 3D nature of objects can rarely be represented by silhouettes. (The 3D equivalent of binary processing uses voxels, spatial occupancy of small cubes in 3D space).
- Specialized lighting is required for silhouettes: it is difficult to obtain reliable binary images without restricting the environment. The simplest example is an overhead projector or light box.

2.6.1 Obtaining the Binary Images

Binary images are typically obtained by thresholding a grey level image. Pixels with a grey level above the threshold are set to 1 (equivalently 255), whilst the rest are set to 0. This produces a white object on a black background (or vice versa, depending on the relative grey values of the object and the background) [8, 10, 11]. Of course, the `negative' of a binary image is also a binary image, simply one in which the pixel values have been reversed. So, the simple global thresholding can be described as:

$$f(x,y) = 1 \quad if \quad f > T \tag{2.6}$$

$$f(x,y) = 0 \quad if \quad f < T \tag{2.7}$$

So, the binary image defining the characteristic function of the object in an image to be:

$$f(x,y) = 1 \qquad f(x,y) \in \ An\ object \tag{2.8}$$

$$f(x,y) = 0 \qquad f(x,y) \in \ \mathrm{Im}age\ background \tag{2.9}$$

2.6.2 Manipulating the Binary Image

The noise and background subtraction drawbacks influence, appears in the output images in form of small objects and holes in the object of interest. To overcome these problems there is a need for binary images manipulation techniques and these techniques are:

(a) Labeling the Connected Components

One of the most common operations in machine vision is finding the connected components in an image. The term connected components refer to a set of pixels in which each pixel is connected to all other pixels in the same region. The points in a connected component form a candidate region for representing an object [10]. A component labeling finds all connected components in an image and assigns a unique label to all points in the same component. In many applications, it is desirable to compute characteristics (such as size, position, and bounding box) of the components while labeling these components. In this research work the labeling process is achieved to all white pixels no matter the size is big or small. However to consider the effective regions there is a need of applying size filtering to remove the noise influence. The next section introduces and discusses the size filtering.

(b) The Morphological Operators

Morphological image processing is a type of processing in which the spatial form or structures of objects within an image are modified. Dilation, erosion and skeletonization are three fundamental morphological operations [9, 10]. With dilation, an object grows uniformly in spatial extent, whereas with erosion an object shrinks uniformly. Skeletonization results in a stick figure representation of an object [9, 10].

This research work adopts and uses closing operation and basically it can be derived from the fundamental operations of erosion and dilation using the same structuring element for both operations. Figure 2.5 illustrates an original RGB frame with background subtraction algorithm and the binary image after applying closing operation.

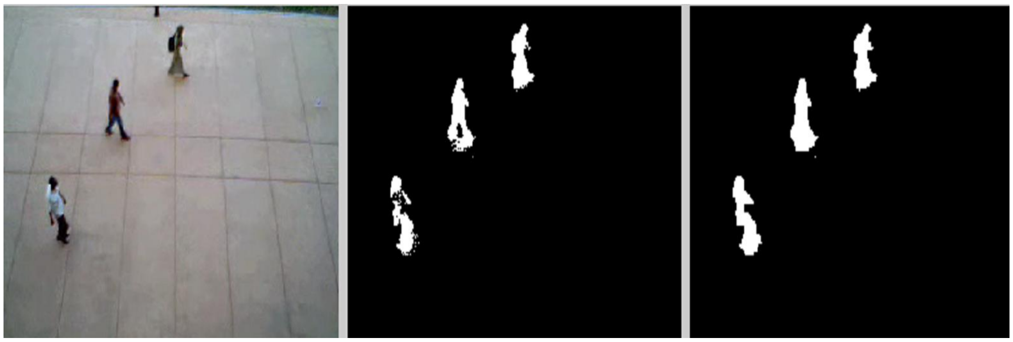


Figure 2.5: Applying binary image closing on a sample frame.

(c) Size Filtering

It's very common to use thresholding for finding a binary image. In most cases, there are some regions in an image that are due to noise, usually, such regions are small. In many applications, it is known that objects of interest are of size greater than T_0 pixels [10]. In such cases one may use a size filter to remove noise after component labeling. Figure 2.6 shows an original RGB frame with the initial result on background subtraction algorithm and the result after applying size filtering.

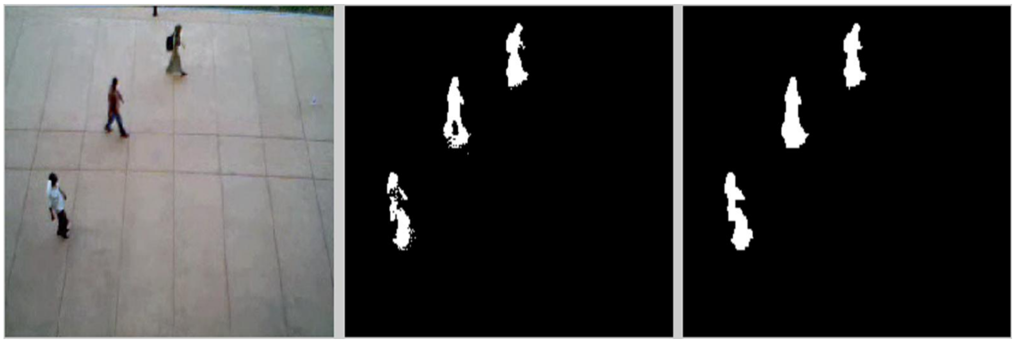


Figure 2.6: Applying binary image size filtering on a sample frame.

All components below T_0 size are removed via changing the corresponding pixels intensity to 0. This simple filtering mechanism is very effective in removing the influence of noise.

2.7 The Constraint Tracking Method

Due to the presence of multiple concurrent objects in the scene, it is important to have consistent labeling of these objects throughout the video sequence. The problem becomes even more complex when these objects occlude each other in the scene. Thus, in the surveillance domain, a system must detect and track objects as well as handle entries and exits in the scene. Many different types of tracking algorithms have been proposed to cover the surveillance systems requirements. According to [20], these algorithms can be listed under the following categories:

Point tracking method defined as the correspondence of detected objects represented by points across the frames. Point correspondence is a complicated problem-specially in the presence of occlusions. Overall, point correspondence methods can be divided into two broad categories, namely, deterministic [21, 22, 23, 24, 25, 26] and statistical methods [27, 28, 29].

Kernel tracking is typically performed by computing the motion of the object, which is represented by a primitive object region, from one frame to the next. The object motion is generally in the form of parametric motion (translation, conformal, affine, etc.) or the dense flow field computed in subsequent frames. Where, [20] divides these tracking methods into two subcategories based on the appearance representation used, namely, templates and density-based appearance models [30, 31], and multi view appearance models [32, 33].

Silhouette based methods provide an accurate shape description for these objects. The goal of a silhouette-based object tracker is to find the object region in each frame by means of an object model generated using the previous frames. This model can be in the form of a color histogram, object edges or the object contour. The system will divide silhouette trackers into two categories, namely, shape matching and contour tracking. Shape matching approaches search for the object silhouette in the current frame [34]. Contour tracking approaches, on the other hand, evolve an initial contour to its new position in the current frame by either using the state space models or direct minimization of some energy functional [35, 36].

This study implemented a point based labeling method defined by Matlab to correspond the objects over the frame sequence. The series of the points over the frame sequence is obtained from extracting the center of mass location for the moving objects. The correspondence operation is built on evaluating the distance between the new position of the object of interest and a reference point, the previous step followed by comparing the obtained distance with the measured distance between the previous position of the object of interest and the same reference point. The evaluation considers each two close measured distances refers to the same trajectory of a particular object.

The advantages of this method are the simplicity and low computational power required which will assist in creating real time surveillance systems, while the obvious drawback for this method it fails in handling occlusion which is not look into in this research work and should be considered as one of the future improvements for this research.

2.8 Visual Surveillance Concepts and Directions

Visual surveillance in dynamic scenes attempts to detect, recognize and track certain objects from image sequences, and more generally to understand and describe object behaviors. The aim is to develop intelligent visual surveillance to replace the traditional passive video surveillance that is proving ineffective as the number of cameras exceeds the capability of human operators to monitor them. Visual surveillance in dynamic scenes has a wide range of potential applications, such as:

- Access control in special areas. In some security-sensitive, only people with a special identity are allowed to enter. Biometric feature databases including legal visitors are built beforehand using biometric techniques, and then decide whether the visitor can be cleared for entry .
- Person-specific identification in certain scenes. Personal identification at a distance by a smart surveillance system can help the police to catch suspects. The police may build a biometric feature database of suspects, and place visual

surveillance systems at locations where the suspects usually appear. The systems automatically recognize and judge whether or not the people in view are suspects.

- Crowd flux statistics and congestion analysis. Using techniques for human detection, visual surveillance systems can automatically compute the flux of people at important public areas and implements the results for traffic management.
- Anomaly detection and alarming. In some circumstances, it is necessary to analyze the behaviors of people and vehicles and determine whether these behaviors are normal or abnormal and then make a decision based on learned behavior.
- Interactive surveillance using multiple cameras. For social security, cooperative surveillance using multiple cameras could be used for traffic management.

There have been a number of famous visual surveillance systems. [37] employs a combination of shape analysis and tracking, and constructs models of people's appearances in order to detect and track groups of people as well as monitor their behaviors even in the presence of occlusion and in outdoor environments. This system uses the single camera and grayscale sensor. The VIEWS system [38] at the University of Reading is a three-dimensional (3-D) model based vehicle tracking system. The Pfinder system developed by [13] is used to recover a 3-D description of a person in a large room. It tracks a single non occluded person in complex scenes, and has been used in many applications. As a single-person tracking system, TI, developed by [39], detects moving objects in indoor scenes using motion detection, tracks them using first-order prediction, and recognizes behaviors by applying predicates to a graph formed by linking corresponding objects in successive frames. This system cannot handle small motions of background objects. The system at CMU [40] can monitor activities over a large area using multiple cameras that are connected into a network. It can detect and track multiple persons and vehicles within cluttered scenes and monitor their activities over long periods of time [41].

2.9 Summary

In order to ensure proper functionality of a particular computer vision system, there are critical requirements must be fulfilled. Commonly this is known as a preprocessing level and it usually to process the data to assure that it satisfies certain assumptions implied. The main preprocessing operations in this work are segmenting the foreground moving objects, manipulating the binary images and tracking the objects over the frame sequence.