Doctoral Thesis

Study on Object Tracking Under Outdoor Environments Using Video Cameras

By

Budi Sugandi

08584202

A thesis submitted in partial

fulfillment of the requirements for the degree of

Doctor of Engineering

Associate Professor Hyoungseop Kim

Chairperson of Supervisory Committee

Department of Mechanical and Control Engineering

Kyushu Institute of Technology, Japan

February, 2011

ABSTRACT

In this thesis, we investigated the development of the object tracking system including object detection, object tracking and object identification/recognition and related issues. We explored the related object tracking system methodology and the future development of the tracking system. We have presented various recent methodology of the tracking system including object detection, object tracking and object recognition. On each methodology, we presented detail methodology with the weakness and the strong points. Furthermore, we presented also the future development of the tracking system.

In this thesis, our proposed method is applied into two categories of object tracking system that based on deterministic algorithm and stochastic algorithm. On the first approach, we divide our study on three main categories for building an automated tracking system, which can be listed as object detection, object tracking and object identification. We introduce our proposed method on each category. We propose low resolution image on frame difference to detect the moving object. We proposed a block matching based on PISC image to track the interest object. In addition, to evaluate the method, we proposed an identification method based on the color and spatial features. Our proposed method can achieve satisfactory result with the identification rate of 92.1[%] in average. In order to increase the field of view of camera, we propose also object tracking using multi-camera. We implement the multi-camera system under LAN environment which each camera connected to each PC. We successfully track the interest object using two camera and we obtain the wider view of camera. Our proposed method can achieve the detection rates of 97.23[%] and accuracy of 96.98[%] in average.

On the second approach, we proposed a color-based particle filter for single object and multiple objects tracking. On this approach, we rely on the color likelihood as an image measurement to estimate the state of the moving objects. In addition, to handle the appearance change and background clutter, we proposed also model updating. We analyzed the effect of the number of particles and number of histogram bins to the processing time and tracking accuracy. We obtained that the processing time is related

to the number of particles and number of histogram although the tracking accuracy increase also. The experimental results show the algorithm can successfully track the single moving object based on known and unknown initial position and object appearance. Finally, we proposed to expand the color-based particle filter algorithm to track multiple objects in the presence of occlusion. We proposed to handle occlusion by redefining the resample and model update step. The occlusion itself is predicted based on the estimated position and likelihood measurement. We implemented our proposed algorithm to track two and three moving object which has individual template and independent motion and the satisfactory results are achieved. We found also that the processing time is related to the number of the objects to be tracked.

The tracking system which was proposed in this thesis can be applicable in various applications such as surveillance system, smart room, intelligent transport system, etc. In surveillance system, we can use our proposed system to detect and recognize the anomaly behavior of the people in some area for example airport, supermarket, playground of kinder garden school and so on. Our system can handle the object occlusion and implemented in multi-camera system. So, it makes easier to detect and recognize the anomaly behavior in the crowded environment. For example, we can install our system as surveillance system in the playground of the kinder garden. As our system is based on the color feature and the kinder garden student has different color of hat for different grade, we can detect and track the student based on their hat. If there is anomaly behavior such as, fight between students, the students fall down, etc, the system can detect and recognize it and send the information to the teacher to do an action. In smart room application, for example we can install our system in a hospital in order to recognize and analyze some interactions between the patients and the real environment. We can monitor the actions of the patients using multi-camera system such as sleeping, walking, falling down from bed and other abnormal actions. If the abnormal action happens, the system will send the signal to the operator to make any decision or action. We believe that the application of our developed system can make the society become better.

ACKNOWLEDGMENT

This thesis is a summary of my study in the Department of Mechanical and Control Engineering, Kyushu Institute of Technology. I would like to take this opportunity to thank the people who have contributed in all sorts of ways to this thesis.

First of all, I am very grateful to Assoc. Prof. Hyoungseop Kim for his support, advice and continuous encouragement during my study in his laboratory and in composing this thesis. I wish to thank also the members of my thesis committee: Prof. Seiji Ishikawa, Prof. Yoshihiko Tagawa, and Prof. Seiichi Serikawa for their recommendations and input regarding my work. My sincere thank also to Dr. Joo Kooi Tan for her support and encouragement.

My sincere thank to Ministry of Education, Culture, Sports, Science and Technology (MEXT) Japan for giving me an opportunity to study in Kyushu Institute of Technology (KIT) Japan by giving the Monbukagakusho (MEXT) scholarship.

I would like also to thank all members of KIM Laboratory for their friendly help during my study and for the pleasant and exciting working environment.

Finally, I would like to express my special thank to my lovely wife Wiwin Sarwini and both of my sons Naufal Fadhilah Ardhani and Kenjiro Ramdhan Armani and to my great parents for the continuous support, love, and sacrifices they made that aided me through my education.

Kitakyushu, February 2011

Budi Sugandi

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

CHAPTER I
INTRODUCTION

Visual tracking in dynamic scenes, especially for humans and vehicles, is currently one of the most active research topics in computer vision. As an active research topic, a visual tracking systems attempt to detect, track and recognize certain objects from image sequences, and more generally to understand and describe object behaviors. The aim is to develop intelligent visual tacking to replace the traditional passive video tracking that is proving ineffective as the number of cameras exceeds the capability of human operators to monitor them. In short, the goal of visual tracking is not only to put cameras in the place of human eyes, but also to accomplish the entire tracking task as automatically as possible.

The object tracking system has a wide range of potential applications. We focus in this thesis on applications involving the object tracking for humans, as they are typical object tracking applications in general. The application of object tracking includes surveillance system, real time monitoring, robotic, human machine interaction, etc. This broad range of applications motivates the interests of many researchers worldwide to develop and build the robust and reliable object tracking system. Many researches have been introduced to overcome some difficulties in the object tracking system such as fake motion, illumination change, background clutter and presence of object occlusion. However, the researches on the tracking system still have a challenging task and an open problem need to overcome especially in cases of the presence of fake motion and object occlusions.

The methods of object tracking can be roughly classified into two categories: deterministic methods and stochastic methods. The deterministic methods typically track the object by performing an iterative search for a similarity between the template image and the current one. On the other hand, the stochastic methods use the state space to model the underlying dynamics of the tracking system. This thesis presents a new development of the object tracking system based on both approaches. In the deterministic method, the block matching method based on peripheral increment sign correlation (PISC) image is introduced to track the interest object. Furthermore, the frame difference in low resolution image and object identification using color features is also introduced in this approach. We also implement the multi camera tracking using this approach to increase the field of view (FOV) of the camera. On the other hand, in the stochastic approach, the object tracking is performed based on particle filter to estimate the state of the moving object. The likelihood between the object and particles is verified by image measurements based on the color distributions. In this approach, we propose a color-based particle filter to track the single object and multiple objects in the presence of occlusion and implement in outdoor environment.

1.1 MAIN CONTRIBUTIONS

This thesis investigates the development of the object tracking system including object detection, object tracking and object identification/recognition and related issues. We proposed the methods based on both approach as mentioned above. On the first approach, we proposed a block matching based on PISC image to track the interest object. The proposed method can track the interest object in the presence of other objects. In addition, to evaluate the method, we proposed an identification method based on the color and spatial features. To increase the FOV of camera, we implement also multi-camera tracking using this approach. On the second approach, we proposed a color-based particle filter for single and multiple object tracking. On this approach, we rely on the color likelihood as an image measurement to estimate the state of the moving object. The experimental results demonstrate the effectiveness of the proposed method to track the moving object in different conditions.

The main contributions of this thesis are:

1. *Survey of object tracking system methodology and the future development of the tracking system*

   In this survey, we investigated on various recent methodology of the tracking system including object detection, object tracking and object recognition. On each methodology, we give detail methodology with the weakness and the strong points. At the end of the survey, we investigated the future development of the tracking system.

2. *Interest object tracking based on PISC block matching*

   The object tracking in outdoor environment has more challenging problem than indoor environment. One of the challenges is the noise due to the fake motion of background such as waving leaves, waving flag, etc. Many tracking algorithms have better performance under static background but sometimes miss-tracking results are obtained under background with complex motions. We propose to use low resolution image to remove this noise. Since low resolution image has a nice property that it can remove the small size pixels, it is adopted to solve this problem due to the fact that most of the fake motions in the background have small region. To handle the occlusion problem, we propose a block matching method based on PISC image to track the interest object in occlusion condition. The PISC image is an image matching method with robust performance, high accuracy, and high computational efficiency. The PISC image is constructed from the trend of the brightness changes in the neighborhood of the pixel under consideration. We propose also the identification method to evaluate our method based on color and spatial information of the tracked object.

3. *Multi-camera tracking*

   We propose multi camera tracking in overlapping view to cover large environments of tracking. The multi-camera system is able to switch each other when the tracking object moves from FOV of one camera to the other one. Therefore, the system is possible to track the moving object continuously. Each camera is connected to each PC under local area network (LAN) connection. The communication between PCs is performed using network programming, called socket programming. By specifying IP address of the server from the clients' side, the information of the object can be exchanged between server and clients.

4. *Color-based particle filter tracking*

   The limitation of the previous proposed method is overcome using particle filter approach. The proposed color-based particle filter can track the moving object in noisy environment. We rely on the state space model to model the dynamic of the moving object. The object state is defined using position, speed, size, and size scale of the object model. The state of the object is predicted based on the system model. We used the color distribution measurement between the particles and the object model to update the likelihood of each particle/sample. To ensure the proper tracking object, target-model is update based on certain condition. The estimation of the object state is performed based on the mean of samples likelihood.

5. *Occlusion handling based on particle filter approach*

   The problem of occlusions can cause a failure in the tracking process. They may cause inaccurate in estimation and updating of the state of the tracked object since the likelihood of the occluded target would be meaningless. The resampling phase would propagate the wrong random samples and quickly cause the loosing of the object. We propose to handle occlusion by static and dynamic object by redefining the resample and target update of the samples. The occlusion is predicted based on the dynamical model and likelihood measurement. We divide the occlusion in to two parts: full occlusion and partial occlusion. In partial occlusion, we proposed to measure the likelihood in the region of each un-occluded object based on the predicted state. In this condition, the samples are resampled and the target model is updated. In case of full occlusion, we do not measure the likelihood of occluded object because it will be meaningless. In this condition, there is no resample and model updating; the samples are just propagated based on the system model. However, the likelihood of the samples of un-occluded object is still measured.

1.2 OUTLINE OF THE THESIS

The structure of this thesis is as follows.

Chapter 2 describes a variety of tracking methodologies including object detection, object tracking and object recognition/identification method.

Chapter 3 describes the development of object tracking methodology based on deterministic approach. In this chapter, we introduce frame difference method on low resolution image that can remove the fake motion background on object detection step. We propose also the block matching method based on PISC method to track the interest object and identification method based on color and spatial features of the tracked object to evaluate the effectiveness of our method.

Chapter 4 proposes a multi-camera tracking in outdoor environment. The object detection is performed based on PISC and the object tracking is performed based on the block matching technique. The communication between cameras is performed using the network programming under the LAN environment.

Chapter 5 describes a probabilistic approach for object tracking based on particle filter. This chapter proposes to overcome the limitation of the previous proposed method using a color-based particle filter tracking including the target model and likelihood update.

Chapter 6 describes the implementation of particle filter to track multiple objects in the presence of object occlusions. The object occlusions are predicted and handled using the history of the likelihood measurement and the distance between the samples. The proposed method can solve the occlusion problem properly.

Finally, the conclusions are presented in Chapter 7 along with some future works.

CHAPTER II

OBJECT TRACKING METHODOLOGY

CHAPTER II
OBJECT TRACKING METHODOLOGY

2.1 INTRODUCTION

Object tracking is required by many vision applications such as human-computer interfaces [1], video communication / compression [2] or surveillance system [3-4] and so on. The increase of high-powered computers, the availability of high quality and inexpensive video cameras, and the increasing need for automated video analysis have generated a great deal of interest in object tracking algorithms. Since then, many researchers have developed various methods, algorithms and techniques that can be implemented in the tracking system. These methods primarily differ from each other based on the way of their approaches such as object representation, image feature used for tracking, motion, appearance and shape of the object model. Moreover, the performance of tracking algorithm using image processing method depends on how well they can estimate the motion such an articulate object structure and due to the change of a view point and illumination [5].

There are at least three steps in the tracking system methodologies [6, 7] such as detection of the interest moving objects, tracking of such objects from frame to frame, and analysis of the tracked objects to recognize their behavior.

The first step is object detection. This step performs the detection methodology of the interest objects appear in the scene or FOV of any camera. The problem of objects detection usually appears when the background is clutter, the change of object appearance and the presence of noise.

The second step is the tracking. It can be defined as the problem of estimating the trajectory of an object in the image plane as it moves around a scene. In other words, a tracker assigns consistent labels to the tracked objects in different frames of a video. Additionally, depending on the tracking domain, a tracker can also provide object-centric information, such as orientation, area, or shape of an object. The object tracking can be complex due to: noise in images, complex object motion, non-rigid or articulated nature of objects, partial and full object occlusions, complex object shapes, scene illumination changes, and real-time processing requirements. However, the object tracking can be simplified by imposing constraints on the motion and/or appearance of objects. For example, almost of tracking algorithms assume that the object motion is smooth with no abrupt changes. Another constrain assume that the object motion has a constant velocity or constant acceleration based on a priori information. The prior knowledge about the number and the size of objects, or the object appearance and shape, can also be used to simplify the problem.

The last step is behavior recognition or understanding. Behavior understanding involves the analysis and recognition of motion patterns, and the production of high level description of actions and interactions. Understanding of behaviors may simply be thought as the classification of time varying feature data, i.e., matching an unknown test sequence with a group of labeled reference sequences representing typical behaviors. It is then obvious that a fundamental problem of behavior understanding is to learn the reference behavior sequences from training samples, and to devise both training and matching methods for dealing effectively with small variations of the feature data within each class of motion patterns.

The reference [6] also includes the objects representation as one part of object tracking system. In this step, an object can be represented as shape, point and forth depending on the application. Meanwhile an object tracking also can be represented as object appearances such as probability densities of object appearance, template, active appearance models and multi view appearance models. An object appearance means some features or attributes of an object are considered to represent an object such as color, texture and so on. Furthermore, an object appearance also can be called as appearance representations or appearance features.

Currently, many approaches have been proposed to overcome the difficulties on the object detection and tracking system. Normally, the proposed tracking systems depend on the context/environment in which the tracking is performed and the end use for which the tracking information is being sought. The aim of this chapter is to group tracking systems into broad categories and provides comprehensive descriptions of representative methods in each category. Moreover, we aim to identify new trends and ideas in the tracking community and to provide insight for the development of new tracking methods.

2.2 FEATURES FOR OBJECT TRACKING

Selecting the right features plays a critical role in object tracking system. In general, the most desirable property of a visual feature is its uniqueness so that the objects can be easily distinguished in the feature space. For example, color is used as a feature for histogram-based appearance representations, while for contour-based representation, object edges are usually used as features. In general, many tracking algorithms use a combination of these features. Some common visual features are as follows:

2.2.1 Color

The apparent color of an object is influenced primarily by two physical factors; the spectral power distribution of the illuminant and the surface reflectance properties of the object. A variety of color spaces such as RGB (Red, Green, Blue), HSV (Hue, Saturation, Value), HIS (Hue, Intensity, Saturation) and YUV (Y represents luminance while U and V represent chrominance) have been used in many tracking system. RGB color space is a common color space used by many researchers such as Numiaro et al [8] and Czyz et al [9]. They used the color feature in conjunction with a particle filter algorithm. However, the RGB space is not a perceptually uniform color space, i.e, the differences between the colors in the RGB space do not correspond to the color differences perceived by humans [10]. Additionally, the RGB space is sensitive to illumination change. The others color space are HSV, HIS and YUV. HSV and HIS are approximately uniform color spaces. These color spaces are less sensitive to illumination condition [11]. The YUV color space is standard color space used for analog television transmission and well reflects the human eye's response to color. However YUV is not a linear color space [12]. In summary, there is no last word on which color space is more efficient, therefore a variety of color spaces have been used in tracking.

2.2.2 Edge

Object boundaries usually generate strong changes in image intensities. Edge detection is used to identify these changes. An important property of edges is that they are less sensitive to illumination changes compared to color features. Algorithms that track the boundary of the objects usually use edges as the representative feature. Because of its simplicity and accuracy, the most popular edge detection approach is the Canny edge detector [13].

2.2.3 Optical Flow

Optical flow is a dense field of displacement vectors which defines the translation of each pixel in a region. It is computed using the brightness constraint, which assumes brightness constancy of corresponding pixels in consecutive frames [14]. Optical flow is commonly used as a feature in motion-based segmentation and tracking applications. Popular techniques for computing dense optical flow include methods by Horn and Schunck [14], Lucas and Kanade [15], Black and Anandan [16] and Szeliski and Couglan [17].

2.2.4 Texture

Texture is a measure of the intensity variation of a surface which quantifies properties such as smoothness and regularity. Compared to color, texture requires a processing step to generate the descriptors. There are various texture descriptors such as Gray-Level Co-occurrence Matrices (GLCM's) [18] (a 2D histogram which shows the co-occurrences of intensities in a specified direction and distance), wavelets [19] (orthogonal bank of filters), and steerable pyramids [20]. Similar to edge features, the texture features are less sensitive to illumination changes compared to color. Many of researchers used the texture feature to track the object such as Ferreira et al [21] using the texture cue to track the object. Their method is based on wavelet type features. The feature vector consists of six characteristics extracted from the wavelet detail image for each color component. These texture characteristics automatically generate a fuzzy rule using a fuzzy inference classifier.

Among all features, color is one of the most widely used features for tracking. Despite its popularity, most color bands are sensitive to illumination variation. Hence in scenarios where this effect is inevitable, other features are incorporated to model object appearance. And, alternatively, a combination of these features is also utilized to improve the tracking performance. Shen et al [22] proposed to combine shape and color features in their method. Ki and Delp [23] proposed color and edge features, meanwhile

Brassnett et al [24] proposed color and texture features in their system. They show the effectiveness of their method by combining those features.

2.3 TRACKING SYSTEM

The tracking system can be defined as an action to estimate the trajectory of an object in the image plane as it moves around scene. In other words, a tracker assigns consistent labels to the tracked objects in different frames of a video. The development of tracking system can be divided into 4 categories: indoor or outdoor application and using single camera or multi-camera. The following we discuss advantage and disadvantage of each application.

At early stage of the tracking system, a lot of researches focused on tracking an object movement in an indoor environment such as Cai and Aggarwal [25,26]. They have developed tracking human motions using a plural fixed camera. They have focused on an indoor environment and multivariate Gaussian models are applied to find the most likely matches of human subjects [25]. In other work [26], they proposed a new approach of tracking real objects in an indoor environment. This approach uses two cameras and 3D models. The changes of the objects in position and pose could be tracked and simulated in a virtual world using predefined information of the indoor environment and estimated parameters of the objects. Their system automatically achieves these parameters by searching the best-matching set of template contours in the database and edge pixels of the objects in two given images using generalized Hough transform. Beymer and Konolige [27] used the stereo camera in the indoor application based on disparity template. Their system employed continuous detection to find new people in the scene. The detected people are tracked using correlation on intensity templates with adjustment from stereo detection.

Basically, tracking a human motion in an indoor environment is easier than in an outdoor environment. This is because the indoor environment has other features that can be used for simplifying the process of recognition [26]. Besides that, that curved contour or shape of a human can reduce a possibility of mismatching of object with the background. Moreover, indoor application also gives less noise from other objects other than the object of interest.

However, because of demanding from a society, tracking systems are widely used in outdoor environment activities. The situation of outdoor environment is so different from tracking object motion in the indoor environment. In outdoor environment, tracking an object motion has a problem how to deal with a background itself and also the presence of noise from other non-interest objects. The background of a moving object should be different regarding to the situations and places even though data are taken in the same period. Besides that, the color, pattern or shape of a background sometimes resemble with the shape, color or pattern of an object that we need to track. And the presence of fake motion of non-interest object can make the tracking failure. At this stage, we have to be careful to determine the interest object and to eliminate the noise.

Various researches have been proposed to overcome those issues. Chia et al [28] were focusing their research on pedestrian detection and tracking at crossroad. Their research has used a pedestrian model and a walking rhythm approach to recognize a singe pedestrian. Meanwhile, Qui [29] have proposed corner feature extraction, motion matching and object classification for the detection of a pedestrian and bicycle on a road. Sugandi et al [30] used multi-camera system for outdoor environment application. Their research overcomes the problem of scattering noise in the outdoor environment by applying a PISC image.

However, the tracking systems still have many open problems which should be solved such as depth information for tracking, object occlusion, background clutter, etc. And also, in order to detect unsafe situation, it is required for the tracking system to recognize a natural situation, environment, human an vehicle behaviors in a whole time-length and in a spatial universe such as provide a views taken by cameras scattered and placed in various positions or from various angles in building, campus, airports, stations, streets, highways, or towns for tracking environments, vehicles or humans.

Therefore, nowadays, in order to overcome those problems, most of new research activities in tracking systems are exploring larger dimensions for tracking system [31]. Such as many cameras are employed in a distributed surveillance system to view, watch and recognize irregular and unsafe behaviors. Because, it is not easy to track a human behavior using a camera that has limitation of object view and a limited partial space in a vast universe. Therefore, a multi-camera approach can be an alternative to solve the problem. Using a multi-camera tracking system, the real time processing is one of the important issues to be considered. As we know, using multiple cameras, the system will record a huge volume of video data and for each video data have a different angle of view. Therefore, a challenging task is to track an object movement from a multi video stream. Regarding on that issue, it is possible to ensure the application which can track a moving object on multi video data quickly under minimizing processing time. Literature [26] used two cameras and 3D models to track the object based on the changes of the objects in position and pose using predefined information of the indoor environment and estimated parameters of the objects. In another work, literature [27] used multi-camera to track human body in indoor environment based on line and polygon structure on the background.

Regarding human behavior tracking, Prati et al [32] dealt with enhancing video surveillance systems using multi-modal sensor. In this work, a computer vision system which is able to detect and track people from multiple cameras is integrated with a wireless sensor network mounting PIR (Passive Infra Red) sensors. Wang et al [33] proposed a system based on a distributed, multi-camera video analysis for airport security surveillance. It uses biometric signatures, which are called soft biometry including a person's height, built, skin tone, color of shirts and trousers, motion pattern, trajectory history, etc., to ID and track errant passengers and suspicious events without having to shut down a whole terminal building and cancel multiple flights. Fidaleo et al [34] dealt with analysis of facial gestures through face recognition. They employed facial gestures to understand the meaning of a face. Carranza et al [35] provided a model of 3-D space dynamic real-world scene. They exploited robust motion estimation of a human body and its motion. Ahad et al [90] proposed directional motion history image (DMHI) to recognize the human motion. Their method could obtain high recognition rate compare with motion history image (MHI).

2.4 OBJECT DETECTION

The challenge in a tracking system is to find an efficient method for detecting and tracking an object under fairly difficult natural conditions. In the tracking system, detection means process of localization of object movement or allowing the system to extract and log the object appearance in the repository. Normally, detection process is required during detecting the first object enters in the frame or needed to detect an object in every frame [27]. This temporal information is usually in the form of frame differencing which highlights changing regions in consecutive frames. Given the object regions in the image, it is then the tracker's task to perform object correspondence from one frame to the next to generate the tracks.

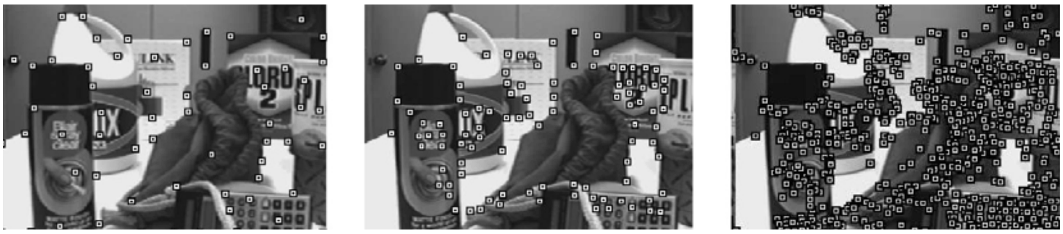
Currently, various detection methods have been proposed for the tracking systems. Based on Collins et al [36], the moving target detection can be divided by three conventional approaches: temporal differencing, background subtraction and optical flow. Meanwhile based on Yilmaz et al [6], they divided onto four categories of a detection method used in tracking systems which are point detectors, segmentation, and background subtraction and supervised learning. We can outline here some methods of object detection.

2.4.1 Point Detector

Point detectors are used to find interest points in the images which have an expressive texture in their respective localities. Interest points have been long used in the context of motion, stereo and tracking problems. A desirable quality of an interest point is its invariance to changes in illumination and camera viewpoint. In the literature, the methods commonly used interest point detectors including Harris interest point detector [37], KLT (Kanade-Lucas-Tomasi) detector [38], and SIFT (Scale Invariant Feature Transform) detector [39]. The survey by Mikolajczyk and Schmid [40] described a comparative evaluation of interest point detectors.

The Harris detector computes the first order image derivatives, (I_x, I_y), in x and y directions to highlight the directional intensity variations, then a second moment matrix, which encodes this variation, is evaluated for each pixel in a small neighborhood as following,

$$M = \begin{pmatrix} \sum I_x^2 & \sum I_x \sum I_y \\ \sum I_x \sum I_y & \sum I_y^2 \end{pmatrix}. \tag{2.1}$$



(a) (b) (c)

Figure 2.1. Point detector taken from [6].

(a) using Harris, (b) using KLT and (c) using SIFT operator

An interest point is identified using the determinant and the trace of M which measures the variation in a local neighborhood,

$$R = \det(M) - k.tr(M)^2 \tag{2.2}$$

where k is constant.

The interest points are marked by threshold R after applying non-maxima suppression as shown in Fig. 2.1(a). The same moment matrix M given in Eq. (2.1) is used in the interest point detection step of the KLT tracking method. Interest point confidence, R, is computed using the minimum eigen value of $M(\lambda_{min})$. Interest point candidates are selected by threshold R. Among the candidate points, KLT eliminates the candidates that are spatially close to each other (Fig. 2.1(b)). In practice, both of these methods find almost the same interest points. The only difference is the additional KLT criterion that enforces a predefined spatial distance between detected interest points.

In theory, the M matrix is invariant to both rotation and translation. However, it is not invariant to affine or projective transformations. In order to introduce robust detection of interest points under different transformations, Lowe [39] introduced the SIFT method, shown by Fig. 2.1(c), which is composed of four steps. First, a scale space is constructed by convolving the image with Gaussian filters at different scales. Convolved images are used to generate difference-of-Gaussians (DoG) images. Candidate interest points are then selected from the minima and maxima of the DoG images across scales. The next step updates the location of each candidate by interpolating the color values using neighboring pixels. In the third step, low contrast candidates as well as the candidates along the edges are eliminated. Finally, remaining interest points are assigned orientations based on the peaks in the histograms of gradient directions in a small neighborhood around a candidate point. SIFT detector generates a greater number of interest points compared to other interest point detectors. This is due to the fact that the interest points at different scales and different resolutions (pyramid) are accumulated. Empirically, it has been shown in Mikolajczyk and Schmid [40] that SIFT outperforms most point detectors and is more resilient to image deformations.

2.4.2 Background Subtraction

Object detection can be achieved by building a representation of the scene called the background model and then finding deviations from the model for each incoming frame. Any significant change in an image region from the background model signifies a moving object. The pixels constituting the regions undergoing change are marked for further processing. Usually, a connected component algorithm is applied to obtain connected regions corresponding to the objects. This process is referred to the background subtraction. Background subtraction is a popular method for motion segmentation, especially under those situations with a relatively static background [41]. It detects moving regions in an image by taking the difference between the current image and the reference background image in a pixel-by-pixel. It is simple, but extremely sensitive to change in dynamic scenes derived from lighting and extraneous events etc. Therefore, it is highly dependent on a good background model to reduce the influence of these changes [42, 43] as part of environment modeling.

There are different approaches to this basic scheme of background subtraction in terms of foreground region detection, background maintenance and post-processing. In [44], Heikkila and Silven used the simple version of this scheme where a pixel location (x, y) in the current image is marked as foreground if the inequality of Eq. (2.3) is satisfied,

$$|I_t(x, y) - B_t(x, y)| > T_h \ . \tag{2.3}$$

Here, T_h is a pre-defined threshold. The background image B_t is updated by the use of a first order recursive filter,

$$B_{t+1} = \alpha I_t + (1-\alpha) B_t \tag{2.4}$$

where α is an adaptation coefficient.

Although background subtraction techniques perform well at extracting most of the relevant pixels of moving regions, they are usually sensitive to dynamic changes when, for instance, stationary objects uncover the background or sudden illumination changes occur. To overcome this limitation some researchers used the statistical background model which will be covered on the next section. And the other using the correlation between reference background and the current image such as the method proposed by Satoh et al [45] using peripheral increment correlation (PISC) image that robust to illumination change. The PISC image is used to detect the moving object based on the trend of the brightness changes in the neighborhood of the pixel under consideration. The matching between two images in PISC is defined as using the following equation,

$$B = \frac{1}{16}\sum_{k=0}^{15} c_k(i,j) \tag{2.5}$$

$$c_k(i,j) = b_k(i,j) \bullet b'_k(i,j) + \left(1 - b_k(i,j)\right) \bullet \left(1 - b'_k(i,j)\right) \tag{2.6}$$

where *b* is background image and *b'* is current image. The PISC image is defined as,

$$I_{i,j} = \begin{cases} 1 & (B \le B_T) \\ 0 & (otherwise) \end{cases}. \tag{2.7}$$

Here, B_T is a threshold.

The comparison result between the conventional background subtraction and PISC method is described in Fig 2.2. The figure is taken from reference [30].

(a) (b) (c)

Figure 2.2. Comparison between background subtraction and PISC image [30].
(a) original image, (b) conventional background subtraction image and (c) PISC image

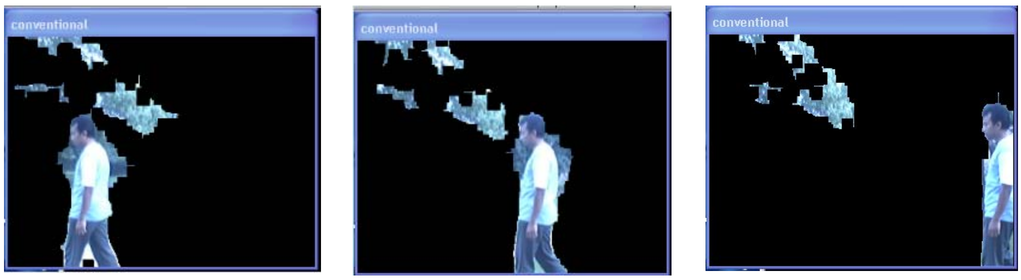
2.4.3 Inter Frame Difference

Frame difference detects the moving object by use of the pixel-wise differences between two or three consecutive frames in an image sequence. Frame difference is very adaptive to dynamic environments, but generally does a poor job of extracting all the relevant pixels, e.g., there may be holes left inside moving entities. This method also fails to detect stopped objects in the scene. Additional methods need to be adopted in order to detect stopped objects for the success of higher level processing.

Lipton et al [46] presented a two-frames differencing to detect moving targets in real video streams. After the absolute difference between the current and the previous frame is obtained, a threshold function is used to determine changes as expressed in following equation,

$$I_t(x, y) - I_{t-1}(x, y)| > Th. \tag{2.8}$$

In order to overcome disadvantage of two-frames differencing, in some cases three-frames differencing can be used. For instance, Collins et al [36] developed a hybrid method that combines three-frame differencing with an adaptive background subtraction model for their VSAM (Video Surveillance and Monitoring) project. The hybrid algorithm successfully segments moving regions in video without the defects of temporal differencing and background subtraction.



(a)

(b)

Figure 2.3. Comparison of moving object detection technique using frame difference [47]. (a) frame difference on normal image, (b) frame difference on low resolution image

Sugandi et al [47] used three-frame differences in low resolution images to overcome the fake motion arise on the background. Fig. 2.3 shows an example of three differences image on normal image and low resolution image. The low resolution image can remove non-interest moving object or fake motion from image detection result.

2.4.4 Optical Flow

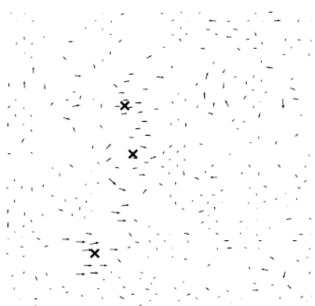
Optical-flow-based object detection uses characteristics of flow vectors of moving objects over time to detect moving regions in an image sequence. For example, Meyer et al [51] compute the displacement vector field to initialize a contour based tracking algorithm, called active rays, for the extraction of articulated objects. The results are used for gait analysis. Fig 2.4 shows a segmentation result of this method.

Horn and Schunck [14] and Lucas and Kanade [15] proposed the computation of optical flow under the brightness constancy constraint,

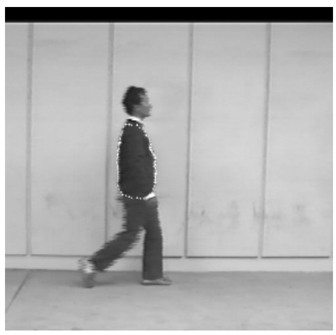
$$I(x, y, t) - I(x + dx, y + dy, t + dt) = 0. \tag{2.9}$$

This computation is always carried out in the neighborhood of the pixel either geometrically [14] or algebraically [15]. Extending optical flow methods to compute the translation of a rectangular region is trivial. Reference [38] proposed the KLT which iteratively computes the translation (du, dv) of a region centered on an interest point,

$$\begin{pmatrix} \sum I_x^2 & \sum I_x \sum I_y \\ \sum I_x \sum I_y & \sum I_y^2 \end{pmatrix} \begin{pmatrix} du \\ dv \end{pmatrix} = \begin{pmatrix} \sum I_x I_t \\ \sum I_y I_t \end{pmatrix}. \tag{2.10}$$



(a)



(b)

Figure 2.4. Object segmentation using optical flow taken from [51].
(a) optical flow of two image, (b) segmentation result of the trunk

This equation is similar in construction to the optical flow method proposed by Lucas and Kanade [15]. Once the new location of the interest point is obtained, the KLT evaluates the quality of the patch by computing the affine transformation,

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \tag{2.11}$$

between the corresponding patches in consecutive frames. If the sum of square difference between the current patch and the projected patch is small, they continue tracking the feature, otherwise the feature is eliminated. More detailed discussion of optical flow can be found in Barron's work [52].

Optical-flow-based methods can be used to detect independently moving objects even in the presence of camera motion. However, most flow computation methods are computationally complex and very sensitive to noise, and cannot be applied to video streams in real time without specialized hardware.

2.4.5 Skin Color Detection

Skin-color detection has been employed in many applications such as face detection, gesture recognition, human tracking, etc. It has been proved that using the skin-color information in the preprocessing can reduce the difficulties of problems. But due to the effects of different human races, ambient lights and confusing backgrounds, detecting skin-color accurately is not an easy task.

Many approaches have been proposed to detect skin-color in static images, and can be classified into three categories: explicitly defined skin region [53-55], nonparametric skin distribution modeling [56] and parametric skin distribution modeling [57]. Another method proposed by Huynh-Thu et al [58]. They proposed a method to find the optimal threshold automatically for the each sub-model in the GMM (Gaussian Mixture Model). In [59], Phung et al proposed an adaptive skin segmentation technique which employed the texture characteristics of the human skin. Cho et al [60] also presented a skin-color filter that was capable of adaptively adjusting its thresholds and effectively separating skin-color regions from similar background color regions. An example of face detection using skin color is shown in Fig 2.5.

(a)

(b)

Figure 2.5. Face detection taken from reference [57].
(a) original image, (b) extraction result

2.4.6 Statistical Method

More advanced methods that make use of the statistical characteristics of individual pixels have been developed to overcome the shortcomings of basic background subtraction and frame difference methods. These statistical methods are mainly inspired by the background subtraction methods in terms of keeping and dynamically updating statistics of the pixels that belong to the background image process. Foreground pixels are identified by comparing each pixel's statistics with the background model. This approach is becoming more popular due to its reliability in scenes that contain noise, illumination changes and shadow [48].

The W4 [49] system uses a statistical background model where each pixel is represented with its minimum (M) and maximum (N) intensity values and maximum intensity difference (D) between any consecutive frames observed during initial training period where the scene contains no moving objects. A pixel in the current image I_t is classified as foreground if it satisfies,

$$\begin{aligned} |M(x, y) - I_t(x, y)| &> D(x, y), \\ |N(x, y) - I_t(x, y)| &> D(x, y). \end{aligned} \tag{2.12}$$

After thresholding, a single iteration of morphological erosion is applied to the detected foreground pixels to remove one-pixel thick noise. In order to grow the eroded regions to their original size, a sequence of erosion and dilation is performed on the foreground pixel map. Also, small-sized regions are eliminated after applying connected component labeling to find the regions. The statistics of the background pixels that belong to the non-moving regions of current image are updated with new image data. Fig 2.6 shows an example of foreground detection taken from [49].

Figure 2.6. Foreground region detection while background has different intensity variation [49].

As another example of statistical methods, Stauffer and Grimson [50] described an adaptive background mixture modeled by a mixture of Gaussians which are updated on-line by incoming image data. In order to detect whether a pixel belongs to a foreground or background process, the Gaussian distributions of the mixture model for that pixel are evaluated.

In this model, the values of an individual pixel (e.g. scalars for gray values and vectors for color images) over time is considered as a "pixel process" and the recent history of each pixel, $X_1, \ldots, X_t$, is modeled by a mixture of K Gaussian distributions. The probability of observing current pixel value then becomes,

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \tag{2.13}$$

where $\omega_{i,t}$ is an estimate of the weight of the i^{th} Gaussian ($G_{i,t}$) in the mixture at time t, $\mu_{i,t}$ is the mean value of $G_{i,t}$ and $\Sigma_{i,t}$ is the covariance matrix of $G_{i,t}$. η is a Gaussian probability density function defined as,

$$\eta(X, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} \left|\Sigma\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(X-\mu)^T \Sigma^{-1} (X-\mu)} . \tag{2.14}$$

Decision on K depends on the available memory and computational power. Also, the covariance matrix is assumed to be of the following form for computational efficiency,

$$\Sigma_{k,t} = \sigma_k^2 I \tag{2.15}$$

which assumes that red, green and blue color components are independent and have the same variance.

The procedure for detecting foreground pixels is as follows. At the beginning of the system, the K Gaussian distributions for a pixel are initialized with predefined mean, high variance and low prior weight. When a new pixel is observed in the image sequence, to determine its type, its RGB vector is checked against K Gaussians, until a match is found. A match is defined as a pixel value within γ (=2.5) standard deviations of a distribution. Next, the prior weights of the K distributions at time t ($\omega_{k,t}$), are updated as follows,

$$\omega_{k,t} = (1-\alpha)\omega_{k,t-1} + \alpha(M_{k,t}) \tag{2.16}$$

where α is the learning rate and $M(_{k,\, t})$ is 1 for the matching Gaussian distribution and 0 for the remaining distributions. After this step, the prior weights of the distributions are normalized with the new observation as follows,

$$\mu_t = (1-\rho)\mu_{t-1} + \rho(X_t) \tag{2.17}$$

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T (X_t - \mu_t) \tag{2.18}$$

where

$$\rho = \alpha\eta(X_t | \mu_{t-1}, \sigma_{t-1})\ . \tag{2.19}$$

If no match is found for the new observed pixel, the Gaussian distribution with the least probability is replaced with new distribution with the current pixel values as its mean value, an initially high variance and low prior weight.

(a) (b) (c)

Figure 2.7. Mixture of Gaussian modeling for background subtraction [50].

(a) original image, (b)the mean of the highest-weighted Gaussians at each pixels position,

(c) foreground result

In order to detect the type (foreground or background) of the new pixel, the K Gaussian distributions are sorted by the value ω/σ. This ordered list of distributions reflect the most probable background from top to bottom since by Eq. (2.20) background pixel processes make the corresponding Gaussian distribution have larger prior weight and less variance. Then the first B distributions are chosen as the background model, where

$$B = \arg\min_{b}\left(\sum_{k=1}^{b} \omega_k > T\right) \tag{2.20}$$

and T is the minimum portion of the pixel data that should be accounted for by the background. If a small value is chosen for T, the background is generally unimodal. Fig.2.7 shows a result of object detection using Gaussian mixture model taken from [50].

2.5 OBJECT TRACKING

After motion detection, the tracking systems generally track moving objects from one frame to another in an image sequence. The tracking algorithms usually have considerable intersection with motion detection during processing. Tracking over time typically involves matching objects in consecutive frames using features such as points, lines or blobs. Useful mathematical tools for tracking include the Kalman filter, the Condensation algorithm, the dynamic Bayesian network, the geodesic method, etc.

The objective of tracking method is to generate the path of an object by locating an object position in every frame from a video stream. Basically, two main purposes of tracking method are to determine when a new object enters the system and secondly, to estimate the position of an object over time. Generally tracking object movements is a challenging task because a target changes dynamically. In a traditional way, an object motion can be measured by two models: stochastic models with deterministic and random components and secondly stochastic model by classical deterministic mechanics [64].

In tracking systems, to track an object motion based on constant velocity and constant acceleration model is difficult because this model is not a linear motion and also will affect the size of an object during movement. Therefore we need a compatible method to overcome this problem. Various techniques have been developed for an object tracking system. A. Yilmaz et al [6] have classified object tracking methods into three categories: point tracking, kernel tracking and silhouette tracking. Meanwhile, Wu et al [7] categories the object tracking based on four categories: region based tracking, active contour based tracking, feature based tracking and model based tracking. Fig. 2.8 gives hierarchical structure of the existing methods in object tracking [6].

Here, we will discuss the main tracking categories of the object tracking based on [6].

2.5.1 Point Tracking

Objects detected in consecutive frames are represented by points, and the association of the points is based on the previous object state which can include object position and motion. This approach requires an external mechanism to detect the objects in every frame. This method can be employed using either deterministic methods [65], or probabilistic methods like Kalman or particle filtering technique [66]. The deterministic methods use *qualitative motion heuristics* [65] to constrain the correspondence problem. On the other hand, probabilistic methods explicitly take the object measurement and take uncertainties into account to establish correspondence.

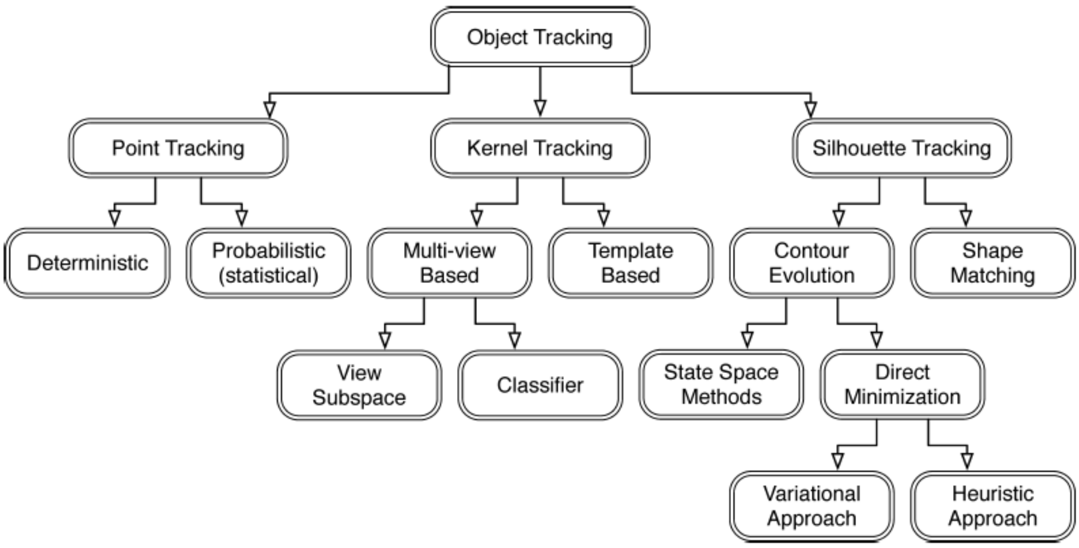


Figure 2.8. A hierarchical method of object tracking [6].

Regarding to our study, we will explain more in the probabilistic method based on Kalman and Particle filter. The probabilistic methods use the state space approach to model the object properties such as position, velocity, and acceleration. Measurements usually consist of the object position in the image, which is obtained by a detection mechanism. The probabilistic methods solve the tracking problems by taking the measurement and the model uncertainties into account during object state estimation.

The information representing the object, for example, location, is defined by a sequence of states $X_t : t = 1, 2, \ldots, t$. The change in state over time is governed by the dynamic equation,

$$X_t = ft(X_{t-1}) + W_t \tag{2.21}$$

where $W_t : t = 1, 2, \ldots$ is white noise. The relationship between the measurement and the state is specified by the measurement equation $Z_t = h_t(X_t, N_t)$, where N_t is the white noise and is independent of W_t. The objective of tracking is to estimate the state X_t given all the measurements up to that moment or, equivalently, to construct the probability density function $p(X_t|Z_{1,\ldots,t})$.

A theoretically optimal solution is provided by a recursive Bayesian filter which solves the problem in two steps. The *prediction* step uses a dynamic equation and the already computed pdf of the state at time $t-1$ to derive the prior pdf of the current state, that is, $p(X_t|Z_{1,\ldots,t-1})$. Then, the *update* step employs the likelihood function $p(Z_t|X_t)$ of the current measurement to compute the posterior pdf $p(X_t|Z_{1,\ldots,t})$. In the case where the measurements only arise due to the presence of a single object in the scene, the state can be simply estimated by the two steps as defined. On the other hand, if there are multiple objects in the scene, measurements need to be associated with the corresponding object states.

The use of Kalman or Particle filter depends on the tracking function itself. If f_t and h_t are linear functions and the initial state X_1 and noise have a Gaussian distribution, then the optimal state estimate is given by the Kalman Filter. However, in the general case, the system is not linear and the object state is not assumed to be a Gaussian, then the state estimation can be performed using the particle filters [67].

Here we will discuss briefly about Kalman and Particle Filter to track the moving object.

2.5.1.1 Kalman Filter

Kalman filter is used to estimate the state of a linear system where the state is assumed to be distributed by a Gaussian. Kalman filtering is composed of two steps, prediction and correction. The prediction step uses the state model to predict the new state of the variables,

$$\begin{aligned} \overline{X}^t &= \mathbf{D}X^{t-1} + W, \\ \overline{\sum}^t &= \mathbf{D}\sum^{t-1}\mathbf{D}^T + Q^t, \end{aligned} \tag{2.22}$$

where $\overline{X}^t$ and $\overline{\sum}^t$ are the state and the covariance predictions at time t. **D** is the state transition matrix which defines the relation between the state variables at time t and $t-1$. Q is the covariance of the noise W. Similarly, the correction step uses the current observations Z_t to update the object's state:

$$\begin{aligned} K^t &= \overline{\sum}^t \mathbf{M}^T [\mathbf{M}\overline{\sum}^t \mathbf{M}^T + R^t]^{-1}, \\ X^t &= \overline{X}^t + \underbrace{K^t [Z^t - \mathbf{M}\overline{X}^t]}_{v}, \\ \sum^t &= \overline{\sum}^t - K^t \mathbf{M}\overline{\sum}^t, \end{aligned} \tag{2.23}$$

where v is called the innovation, **M** is the measurement matrix, K is the Kalman gain, which is the Riccati Equation (the first equation in Eq (2.23)) used for propagation of the state models. Note that the updated state, X^t is still distributed by a Gaussian. In case the functions f_t and h_t are nonlinear, they can be linearized using the Taylor series expansion to obtain the extended Kalman filter [68]. Similar to the Kalman filter, the extended Kalman filter assumes that the state is distributed by a Gaussian.

2.5.1.2 Particle Filter

One limitation of the Kalman filter is the assumption that the state variables are normally distributed (Gaussian distribution). Thus, the Kalman filter will give poor estimations of state variables that do not follow Gaussian distribution. This limitation can be overcome by using particle filtering [67]. In particle filtering, the conditional state density $p(X_t|Z_t)$ at time t is represented by a set of samples or particles $\{s_t^{(n)}: n = 1, \ldots, N\}$ with weights $\pi_t^{(n)}$ (sampling probability). The weights define the importance of a sample, that is, its observation frequency [69]. To decrease computational complexity, for each set ($s^{(n)}$, $\pi^{(n)}$), a cumulative weight $c^{(n)}$ is also stored, where $c^{(N)} = 1$. The new samples at time t are drawn from $\mathbf{S}_{t-1} = \{(s^{(n)}{}_{t-1}, \pi^{(n)}{}_{t-1}, c^{(n)}{}_{t-1}) : n = 1, \ldots, N\}$ at the previous time $t - 1$ step based on different sampling schemes. The most common sampling scheme is *importance sampling* which can be stated as follows.

(1) **Selection**. Select N random samples $\hat{s}_t^{(n)}$ from $\mathbf{S}_{t-1}$ by generating a random number $r \in [0, 1]$, finding the smallest j such that $c_{t-1}^{(j)} > r$ and setting $\hat{s}_t^{(n)} = s_{t-1}^{(j)}$.

(2) **Prediction**. For each selected sample $\hat{s}_t^{(n)}$, generate a new sample by $s_t^{(n)} = f(\hat{s}_t^{(n)}, W_t^{(n)})$, where $W_t^{(n)}$ is a zero mean Gaussian error and f is a non-negative function.

(3) **Update**. Weights $\pi_t^{(n)}$ corresponding to the new samples $s_t^{(n)}$ are computed using the measurements z_t by $\pi_t^{(n)} = p(z_t \mid x_t = s_t^{(n)})$, where $p(.)$ can be modeled as a Gaussian density.

(4) **Resampling**. It will resample the samples based on their weight.

Using the new samples $\mathbf{S}_t$, one can estimate the new object position by $\varepsilon_t = \sum_{n=1}^{N} \pi_t^{(n)} f(s_t^{(n)}, W)$. Particle filter-based trackers can be initialized by either using the first measurements, $s_0^{(n)} \sim X_0$, with weight $\pi_0^{(n)} = \frac{1}{N}$ or by training the system using sample sequences. In addition to keeping track of the best particles, an additional resampling is usually employed to eliminate samples with very low weights. Note that the posterior density does not have to be a Gaussian.

Particle filters recently became popular in computer vision. They are especially used for object detection and tracking. The particle filter, also known as sequential Monte Carlo [91], is the most popular approach which recursively constructs the posterior pdf of the state space using Monte Carlo integration. It has been developed in the computer vision community and applied to tracking problem and is also known as the Condensation algorithm [69]. For another particle filter, Bayesian bootstrap filter was introduced [92]. Fig. 2.9 shows a tracker result of color-based particle filter tracking. This figure is taken from reference [89].

#1 #5 #10

#25 #50 #65

Figure 2.9. Color-based particle filter tracking with unknown initial position [89].

2.5.2 Kernel Tracking

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. These algorithms differ in terms of the appearance representation used, the number of objects tracked, and the method used to estimate the object motion.

Mean-shift tracking technique [70] comes under this class of methods. Given a set $\{x_i\}_{i=1,\ldots,n}$ of n points in the d-dimensional space R^d, the multivariate kernel density estimate with kernel $K(x)$ and window radius (band-width) h, computed in the point $\mathbf{x}$ is given by,

$$\hat{f}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{N} K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right) . \tag{2.24}$$

The minimization of the average global error between the estimate and the true density yields the multivariate Epanechnikov kernel,

$$K_E(\mathbf{x}) = \begin{cases} \frac{1}{2} c_d^{-1}(d+2)(1-\|\mathbf{x}\|^2) & \text{if } \|\mathbf{x}\| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.25}$$

where c_d is the volume of the unit d-dimensional sphere.

Let us introduce the profile of a kernel K as a function $k : [0;\infty) \rightarrow R$ such that $K(\mathbf{x}) = k(\|x\|^2)$. Employing the profile notation, we can write the density estimate as,

$$\hat{f}_K(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} k\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right). \tag{2.26}$$

A kernel G can be defined as,

$$G(\mathbf{x}) = Cg(\| \mathbf{x} \|^2) \tag{2.27}$$

where C is a normalization constant. Then, by taking the estimate of the density gradient as the gradient of the density estimate we have,

$$\begin{aligned}
\hat{\nabla} f_K(\mathbf{x}) \equiv \nabla \hat{f}_K(\mathbf{x}) &= \frac{2}{nh^{d+2}} \sum_{i=1}^{n} (\mathbf{x} - \mathbf{x}_i) k'\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \\
&= \frac{2}{nh^{d+2}} \sum_{i=1}^{n} (\mathbf{x} - \mathbf{x}_i) g\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \\
&= \frac{2}{nh^{d+2}} \left[\sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \right] \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)} - \mathbf{x} \right]
\end{aligned} \tag{2.28}$$

where $\sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)$ can be assumed to be nonzero. Note that the derivative of the Epanechnikov profile is the uniform profile, while the derivative of the normal profile remains a normal. The last bracket in Eq.(2.28) contains the sample mean shift vector,

$$M_{h,G}(\mathbf{x}) \equiv \frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)} - \mathbf{x} \ . \tag{2.29}$$

And the density estimate at **x**,

$$\nabla \hat{f}_G(\mathbf{x}) = \frac{C}{nh^d} \sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) \tag{2.30}$$

computed with kernel G. Using Eq.(2.29) and (2.30), Eq. (2.28) becomes

$$\hat{\nabla} f_K(\mathbf{x}) = \hat{f}_G(\mathbf{x}) \frac{2/C}{h^2} M_{h,G}(\mathbf{x}) \ . \tag{2.31}$$

From where it follows that,

$$M_{h,G}(\mathbf{x}) = \frac{h^2}{2/C} \frac{\hat{\nabla} f_K(\mathbf{x})}{\hat{f}_G(\mathbf{x})} . \tag{2.32}$$

Expression (2. 32) shows that the sample mean shift vector obtained with kernel *G* is an estimate of the normalized density gradient obtained with kernel *K*.

The mean shift procedure is defined recursively by computing the mean shift vector $M_{h,G}(\mathbf{x})$ and translating the center of kernel *G* by $M_{h,G}(\mathbf{x})$. Let us denote by $\{\mathbf{y}_j\}_{j=1,\ldots,n}$ the sequence of successive locations of the kernel *G*, where

$$\mathbf{y}_{j+1} = \frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{y}_j - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{y}_j - \mathbf{x}_i}{h}\right\|^2\right)}, \quad (j = 1,2\ldots) \tag{2.33}$$

is the weighted mean at $\mathbf{y}_j$ computed with kernel and y_1 is the center of the initial kernel. The density estimates computed with kernel *K* in the points (2.33) are

$$\nabla f_K(\mathbf{x}) = \left\{\hat{f}_K(j)\right\}_{j=1,2,..} \equiv \left\{\hat{f}_K(\mathbf{y}_j)\right\}_{j=1,2,..} \tag{2.34}$$

These densities are only implicitly defined to obtain $\hat{\nabla} f_K(\mathbf{x})$. Fig 2.10 shows some results from mean shift tracking taken from [70].

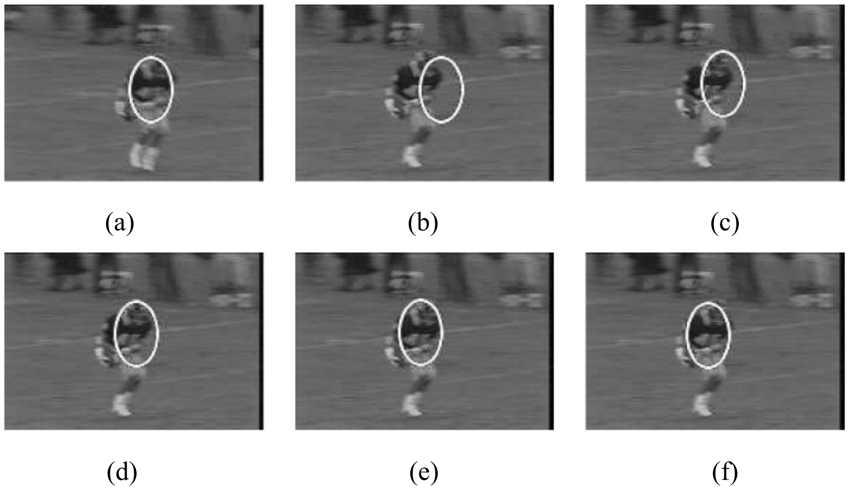


Figure 2.10. Mean-shift tracking iteration [70].
(a) estimated object location at time t-1,
(b) frame at time t with initial location estimate using the previous object position,
(c), (d), (e) location update using mean-shift iterations, (f) final object position at time t

2.5.3 Silhouette Tracking

Tracking is performed by estimating the object region in each frame. Silhouette tracking methods use the information encoded inside the object region. This information can be in the form of appearance density and shape models which are usually in the form of edge maps. Given the object models, silhouettes are tracked by either shape matching or contour evolution. 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 object contour. We can divide silhouette trackers into two categories, namely, shape matching and contour tracking. Shape matching approaches search for the object silhouette in the current frame. 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.

Shape matching can be performed similar to tracking based on template matching, where an object silhouette and its associated model is searched in the current frame. The search is performed by computing the similarity of the object with the model generated from the hypothesized object silhouette based on previous frame. In this approach, the silhouette is assumed to only translate from the current frame to the next, therefore non-rigid object motion is not explicitly handled. The object model, which is usually in the form of an edge map, is reinitialized to handle appearance changes in every frame after the object is located. This update is required to overcome tracking problems related to viewpoint and lighting condition changes as well as non-rigid object motion.

One of examples of this approach is proposed by Huttenlocher et al [71]. They performed shape matching using an edge-based representation. They use the Hausdorff distance to construct a correlation surface from which the minimum is selected as the new object position. The Hausdorff metric is a mathematical measure for comparing two sets of points $A = \{a_1, a_2,\ldots, a_n\}$ and $B = \{b_1, b_2,\ldots, b_m\}$ in terms of the least similar members,

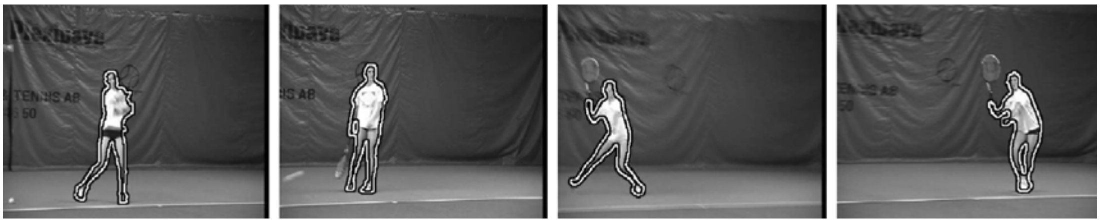
$$H(\mathbf{A},\mathbf{B}) = \max\{h(\mathbf{A},\mathbf{B}), h(\mathbf{B},\mathbf{A})\} \tag{2.35}$$

where $h(\boldsymbol{A}, \boldsymbol{B}) = \sup_{a\in \boldsymbol{A}} \inf_{b\in \boldsymbol{B}} \|a - b\|$ and $\|.\|$ is the norm of choice. In the context of matching using an edge-based model, Hausdorff distance measures the most mismatched edges.

Contour tracking methods, in contrast to shape matching methods, iteratively evolve an initial contour in the previous frame to its new position in the current frame. This contour evolution requires that some part of the object in the current frame overlap with the object region in the previous frame. Tracking by evolving a contour can be performed using two different approaches.

The first approach uses state space models to model the contour shape and motion. The object's state is defined in terms of the shape and the motion parameters of the contour. The state is updated at each time instant such that the contour's a posteriori probability is maximized. The posterior probability depends on the prior state and the current likelihood which is usually defined in terms of the distance of the contour from observed edges. One example of this method is proposed by Cormick and Blake [72]. They extended the particle filter-based object tracker to track multiple objects by including the *exclusion principle* for handling occlusion. The exclusion principle integrates into the sampling step of the particle filtering framework such that, for two objects, if a feature lies in the observation space of both objects, then it contributes more to the samples of the object which is occluding the other object. Since the exclusion principle is only defined between two objects, this approach can track at most two objects undergoing occlusion at any time instant.

The second approach directly evolves the contour by minimizing the contour energy using direct minimization techniques such as gradient descent. On this approach, Yilmaz et al [73] used evolution of an object contour using the color and texture models generated in a band around the object's boundary (see Fig 2.11(a)). The width of the band serves as a means to combine region and boundary-based contour tracking methods into a single framework. They modeled the object shape and its changes by means of a level set- based shape model. In this model, the grid points of the level set hold the means and the standard deviations of the distances of points from the object boundary. The level set-based shape model resolves the object occlusions during the course of tracking (see Fig. 2.11(b)).



(a)

(b)

Figure 2.11. Contour tracking results taken from [73].

Figure 2.12. (a) tracking of a tennis player, (b) tracking in presence of occlusion

Besides those kinds of tracking method, nowadays many of researchers have proposed their own method. For example, in real-time tracking of plural people using continuous detection, Beymer and Konilige used Kalman filtering technique to track movement of people [27]. However, based on Isard and Blake [74], this technique has limited use because they are based on unimodal Gaussian densities that cannot support simultaneous alternative motion hypotheses. Meanwhile Collins et al [36] used a simple approach based on a frame-to-frame matching cost function to track the object movement such as humans and vehicles. Fung and Jerrat [75] proposed a neural network based intelligent intruder's detection and tracking system using CCTV images used barycentre calculation technique in a tracking module. This technique can be loosely compared to the center of gravity of an object.

2.6 OBJECT BEHAVIOR RECOGNITION AND UNDERSTANDING

As we mentioned above, object detection, object tracking and object representation are a main module in tracking system and a part of module in tracking and monitoring system. But these modules are not sufficient in obtaining the final result to the end user. Therefore, we need a system that can recognize an object behavior to ensure the system can give the decision making.

Recognizing an object behavior or object activity has been addressed by many researchers in different fields such as computer vision, multimedia processing and pervasive computing. Behaviors refer to the actions or reactions of an object or organism, usually in relation to the environment. Meanwhile human behavior is the collection of activities performed by human beings. Therefore, to understand an object behavior is one of most difficult open problem in tracking application. After detection and tracking an object, a tracking system should determine what kind of activity the object is doing. For example using star skeletonization procedure to analyze the human motion. Using this procedure, they could distinguish a human motion between walking and running [36]. Park and Aggarwal [79] used a hierarchical Bayesian network (BN) to recognize of two person interactions. Where using a low level of BN can track a part of a human body, to track a whole body they used a high level of the BN. In statistical synthesis of facial expression, Gralewski et al [80] used principal components analysis (PCA) and the application of an auto-regressive process (ARP) to define a set of emotion model. This technique generates a video texture from sequences of coherent facial expression and head motions. Meanwhile, Gunes et al [81] interpreted a human movement, body behavior and facial expression in order to make a human-computer interface truly nature.

Leo et al [82] used Hidden Markov Models (HMM) to recognize of human behavior. In this research, they could recognize the human's postures such as a human standing, squatted and bent. Using Hidden Markov Models, they could identify four kinds of activities: walking, probing the subsoil by a stick, damping the ground with a tank and picking-up some objects from the ground. Ahad et al [90] proposed DMHI to recognize the human motion. Their method could obtain high recognition rates compare with MHI method.

2.6.1 Behavior Understanding

One of the objectives of visual tracking system is to analyze and interpret individual behaviors and interactions between objects to decide for example whether people are carrying, depositing or exchanging objects, whether people are getting on or getting off a vehicle, or whether a vehicle is overtaking another vehicle, etc.

Recently, related research has still focused on some basic problems like recognition of standard gestures and simple behaviors. Some progress has been made in building the statistical models of human behaviors using machine learning. Behavior recognition is complex, as the same behavior may have several different meanings depending upon the scene and task context in which it is performed. This ambiguity is exacerbated when several objects are present in a scene [83].

The following problems within behavior understanding are challenging: statistical learning for modeling behaviors, context-dependent learning from example images, real-time performance required by behavior interpretation, classification and labeling of motion trajectories of tracked objects, automated learning of the *priori* knowledge [84] implied in object behaviors, visually mediated interaction, and attention mechanisms.

2.6.2 Anomaly Detection and Behavior Prediction

Anomaly detection and behavior prediction are significant in practice. In applications of visual tracking system, not only should the systems detect anomalies such as traffic accidents and car theft etc, according to requirements of functions, but also predict what will happen according to the current situation and raise an alarm for a predicted abnormal behavior. Implementations are usually based on one or other of the following two methods. First is probability reasoning and prior rules combined methods. A behavior with small probability or against the prior rules would be regarded as an anomaly. And the second is behavior-pattern-based methods. Based on learned patterns of behaviors, we can detect anomalies and predict object behaviors. When a detected behavior does not match the learned patterns, it is classed as an anomaly. We can predict an object behavior by matching the observed sub-behavior of the object with the learned patterns.

Generally, patterns of behaviors in a scene can be constructed by supervised or unsupervised learning of each object's velocities and trajectories, etc. Supervised learning is used for known scenes where objects move in pre-defined ways. For unknown scenes, patterns of behaviors should be constructed by self-organizing and self-learning of image sequences. Fernyhough et al [85] establish the spatio-temporal region by learning results of tracking objects in an image sequence, and construct a qualitative behavior model by qualitative reasoning and statistical analysis.

2.7 CONCLUSIONS

We have presented an overview of recent developments in visual tracking methodologies within a general processing framework for visual tracking systems. The state-of-the-art of existing methods in each key issue is described with the focus on the following tasks: detection, tracking, understanding and recognizing of behaviors, personal identification, and interactive visual tracking using multiple cameras.

Three stages of tracking system are addressed: *object detection*, *object tracking*, *and object behavior recognition*. We have discussed six methods for object detection such as: *point detector*, *background subtraction*, *frame differencing*, *optical flow*, *skin color extraction* and *statistical method*. In the object tracking section, we discussed some methods such as *point tracking*, *kernel tracking* and *silhouette tracking*. The last part of object tracking system is *object behavior recognition* and *understanding*. This part can be used to detect and recognize the anomaly behavior of the detected object and so on.

CHAPTER III

TRACKING THE INTEREST OBJECT AND ITS IDENTIFICATION METHOD

CHAPTER III
TRACKING THE INTEREST OBJECT AND ITS IDENTIFICATION METHOD

3.1 INTRODUCTION

This chapter presents a method for tracking the interest object and its identification method. The main study in this chapter includes three main categories for building an automated tracking system, which can be listed as object detection, object tracking and object identification as shown in Fig. 3.1. In each step, we introduce our proposed method.

Nearly, every tracking system starts with motion detection. Motion detection aims at separating the corresponding moving objects region from the background image. The first process in the motion detection is capturing the image information using a video camera. The motion detection stage includes some image preprocessing and filtering step such as; gray-scaling and smoothing, reducing image resolution using low resolution image technique, frame difference, morphology filter and connected component labeling. The preprocessing steps are applied to reduce the image noise in order to achieve a higher accuracy of the tracking. The image smoothing technique is performed by using median filter. The low resolution image is performed in three successive frames to remove the small or fake motion in the background. Then frame difference is performed on those frames to detect the moving objects emerging in the scene. The next process is applying morphology filter such as dilation and erosion as a filter to reduce the noise that is remained in the moving object. Connected component labeling is then performed to label each moving object in different label.

The second stage is tracking the moving object. In this stage, we perform a block matching technique to track only the interest moving object among the moving objects emerging in the background. The blocks are defined by dividing the image frame into non-overlapping square parts. The blocks are made based on PISC image [30, 46] that considers the brightness change in all the pixels of the blocks relative to the considered pixel.

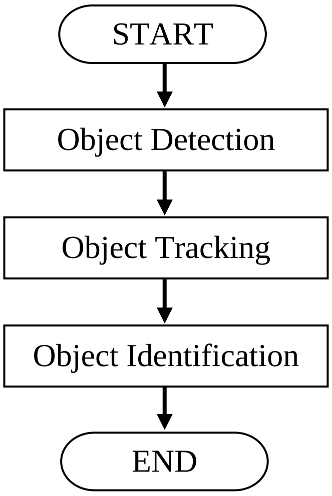


Figure 3.1. Entire flow of processing.

The last stage is object identification. For this purpose we use spatial and color information of the tracked object as the image feature [93]. Then, a feature queue is created to save the features of the moving objects. When the new objects appear on the scene, they will be tracked and labeled, and the features of the object are extracted and recorded into the queue. Once a moving object is detected, the system will extract the features of the object and identify it from the identified objects in the queue.

The details of each stage are described as following.

3.2 OBJECT DETECTION

Performance of an automated visual tracking system considerably depends on its ability to detect moving objects in the observed environment. A subsequent action, such as tracking, analyzing the motion or identifying objects, requires an accurate extraction of the foreground objects, making moving object detection a crucial part of the system. In order to decide on whether some regions in a frame are foreground or not, there should be a model for the background intensities. This model should also be able to capture and store necessary background information. Any change, which is caused by a new object, should be detected by this model, whereas un-stationary background regions, such as branches and leaves of a tree or a flag waving in the wind, should be identified as a part of the background. In this thesis we propose method to handle those problems related to un-stationary background such as branches and leaves of a tree by reducing the resolution of the image.

Our object detection method consists of two main steps. The first step is pre-processing step including gray scaling, smoothing, and reducing image resolution and so on. The second step is filtering to remove the remaining image noise contained in the object. The filtering is performed by applying the morphology filter such as dilation and erosion. And finally connected component labeling is performed on the filtered image. The entire process of moving object detection is illustrated in Fig. 3.2.

3.2.1 Preprocessing

The first step on the moving object detection process is capturing the image information using a video camera. Image is capture by a video camera as 24 bit RGB (red, green, blue) image which each color is specified with 8-bit unsigned integers (0 through 255) that representing the intensities of each color. The size of the captured image is 320×240 pixels. This RGB image is used as input image for the next stage.

In order to reduce the processing time, gray-scale image is used on entire process instead of color image. The gray-scale image only has one color channel that consists of 8 bit while RGB image has three color channels. Image smoothing is performed to reduce image noise from input image in order to achieve high accuracy for detecting the moving objects. The smoothing process is performed by using a median filter with kernel size 5×5 pixels.

We consider un-stationary background such as branches and leaves of a tree as part of the background. The un-stationary background is often considered as a fake motion other than the motion of the object interest and can cause the failure of detection of the object. To handle this problem, we reduce the resolution of the image to be a low resolution image. A low resolution image is done by reducing spatial resolution of the image with keeping the image size [94] and [47]. In this thesis, the low resolution image is done by averaging pixels value of its neighbors, including itself. For example, a video image with resolution 320x240 pixels has original image size 320×240 pixels. After reducing the resolution, the numbers of pixels will be 160×120, 80×60, or 40×30 pixels, respectively, while the image size is still 320×240 pixels. The low resolution image can be used for reducing the scattering noise and the small fake motion in the background because of un-stationary background such as leaves of a tree. These noises that have small motion region will be disappeared in low resolution image.

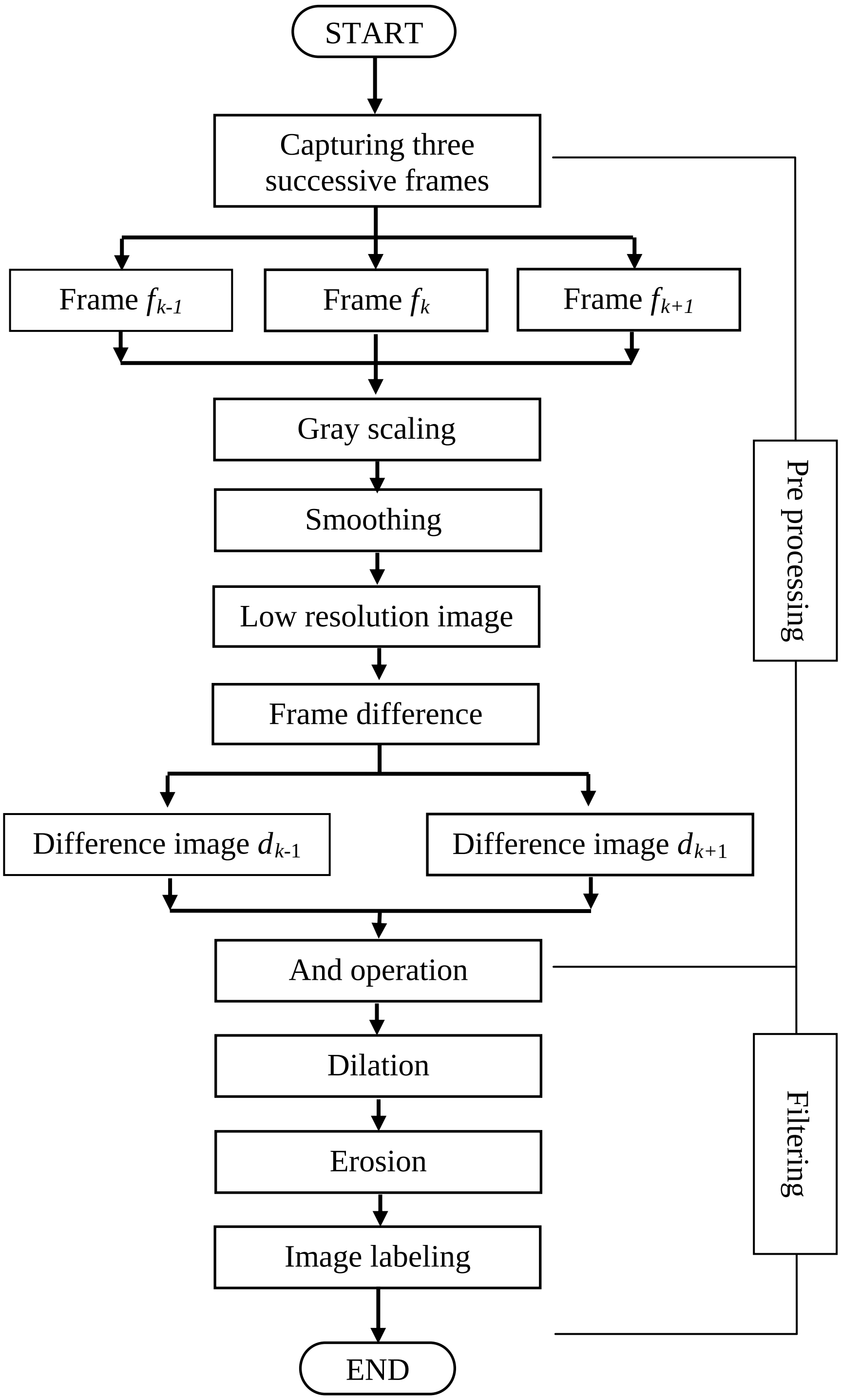


Figure 3.2. Entire flow of the object detection procedure.

To detect the moving object from the background based of image subtraction, generally there are three approaches can be performed: (i) background subtraction as discussed in [41–44], (ii) frame difference as discussed in [46–47], and (iii) combination of background subtraction and frame difference as discussed in [96]. Background subtraction is computing the difference between the current and the reference background image in a pixel-by-pixel. Frame difference is computing the difference image between the successive frames image. In this chapter, we applied frame difference method to detect the moving objects. In our case, frame difference method is performed on the three successive frames, which are between frame f_k and f_{k-1} and between frame f_k and f_{k+1}. The output image as frame difference image is two difference images d_{k-1} and d_{k+1} as expressed in Eq. (3.1). Threshold is performed by threshold value T on the difference image d_{k-1} and d_{k+1} as defined in Eq. (3.2) to distinguish between the moving object and background.

$$d_{k-1} = |f_k - f_{k-1}|$$
$$d_{k+1} = |f_k - f_{k+1}| \tag{3.1}$$

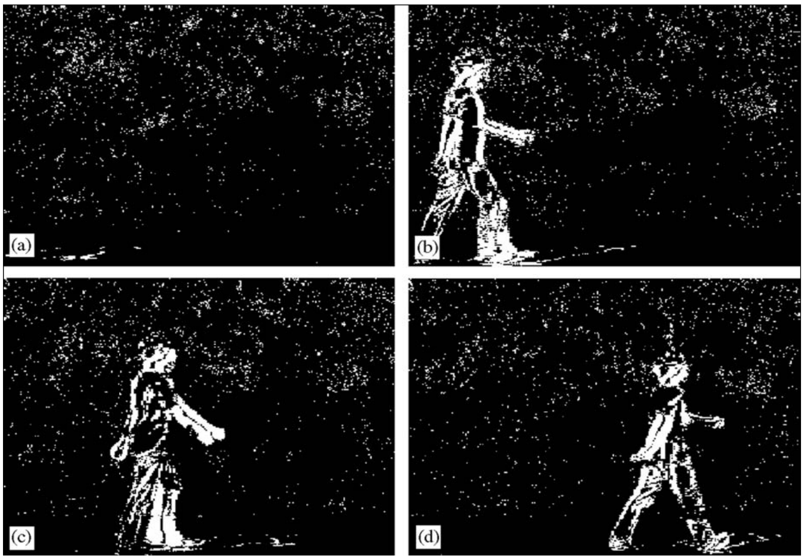
$$d'_k(x,y) = \begin{cases} 1, & if\ \ d'_k(x,y) > T \\ 0, & otherwise \end{cases} \tag{3.2}$$

where $k' = k-1$ and $k+1$.

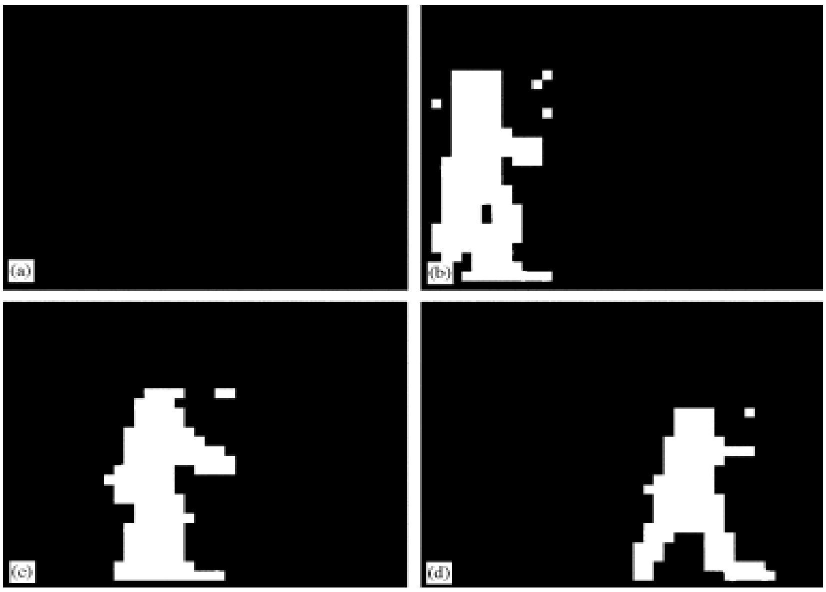
The process is followed by applying AND operator to d_{k-1} and d_{k+1} as expressed in Eq. (3.3). The output image of this operation is named as motion mask m_p.

$$m_p = d_{k-1} \cap d_{k+1}. \tag{3.3}$$

The comparison of moving object detection using a conventional method (frame different on normal resolution) and frame different method on low resolution image is shown in Fig. 3.3. On those figures, we use same threshold to determine the moving object. As shown in those figures, using the conventional method the detected moving object is still greatly affected by small noise such as moving leaves. On the other hand, by reducing the resolution of the image before taking the difference frame, that kind of noise can be removed.



(a)



(b)

Figure 3.3. Comparison of object detection method.

(a) conventional method, (b) based on low resolution image

3.2.2 Object Filtering Using Morphology Filter and Image Labeling

In order to fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour, a morphology filter is applied to the image. As a result, small gaps between the isolated segments are erased and the regions are merged. To extract the bounding boxes of detected objects, connected component analysis was used. We find all contours in image and draw the rectangles around corresponding contours with minimum area. Since the image may contain regions which are composed of background noise pixels and these regions are smaller than actual motion regions, we discard the region with a smaller area than the predefined threshold. As a result, the processing produces perfect bounding boxes.

Morphology filter is performed to fill small gaps inside the moving object and to reduce the noise remained in the moving objects [97]. We implement the morphology filter using dilation followed by erosion. In dilation, each background pixel that is touching an object pixel is changed into an object pixel. Dilation adds pixels to the boundary of the object and closes isolated background pixel. Dilation can be expressed as:

$$f(x,y)=\begin{cases}1, & \textit{if there is one or more pixels of the } 8 \textit{ neighbors are } 1\\ 0, & \textit{otherwise}\end{cases}. \qquad (3.4)$$

In erosion, each object pixel that is touching a background pixel is changed into a background pixel. Erosion removes isolated foreground pixels. Erosion can be expressed as:

$$f(x,y)=\begin{cases}0, & \textit{if there is one or more pixels of the } 8 \textit{ neighbors are } 0\\ 1, & \textit{otherwise}\end{cases}. \qquad (3.5)$$

Morphology filter eliminates background noise and fills small gaps inside an object. This property makes it well suited to our objective since we are interested in generating masks which preserve the object boundary.

Connected component labeling is performed to label each moving object emerging in the background. The connected component labeling [94] groups the pixels into components based on pixel connectivity (same intensity or gray level). In this article, connected component labeling is done by comparing the pixel with the pixel in four neighbors. If the pixel has at least one neighbor with the same label, this pixel is labeled as same as neighbor’s label. The algorithm of connected component labeling algorithm is described as follows:

1. Firstly, image labeling is done on binary image where object is shown as 1 (white) and background is shown as 0 (black).
2. The image is scanned from top-left to search the object pixel. The label is done by scanning the image from left to right and comparing label with the neighbor’s label

in the same line. If the neighbor has the same pixel value, the pixel is labeled as same as previous label.

3. Next, the labeled image is scanned from top-left to bottom-right by comparing with the four neighbors pixel which have already been encountered in the scan (the neighbors (i) to the left of the pixel, (ii) above it, and (iii and iv) the two upper diagonal terms). If the pixel has at least one neighbor, then this pixel is labeled as same as neighbor's label.
4. On the last scanning, the image is scanned from bottom-right to top-left by comparing with the four neighbors pixel as step 3.

3.2.3 Comparative Result Based on Different Image Resolution

To evaluate the effectiveness of low resolution method on tracking system, we did several experiments in the noisy environments to track the moving object by applying inter frame difference on low resolution image on various image resolutions. The experiments are performed to track the single moving object and two moving objects. Fig. 3.4 – Fig. 3.8 show the comparative results of single moving object in various image resolutions, while Fig. 3.9 and Fig. 3.10 show the comparative result of multiple objects tracking. On Fig. 3.4 and Fig. 3.5, where the resolution factor is 1 and 1/2, respectively, image noise still appears in the scene and the system fails to track the moving object. On those figures, not only the moving object is tracked but also the fake motion of leaves is tracked. However, from Fig. 3.6 – Fig. 3.8, where the resolution factor is 1/4, 1/8 and 1/16, respectively, the image noise is significantly removed while only the moving object is tracked. Those experiments show that the frame difference based on low resolution image removes the noise and can track the moving objects successfully. The rectangle area around the object on each frame shows the extracted moving objects.

Figure 3.4. Single moving object tracking using a conventional method.

Figure 3.5. Single moving object tracking with resolution factor = 1/2.

Figure 3.6. Single moving object tracking with resolution factor = 1/4.

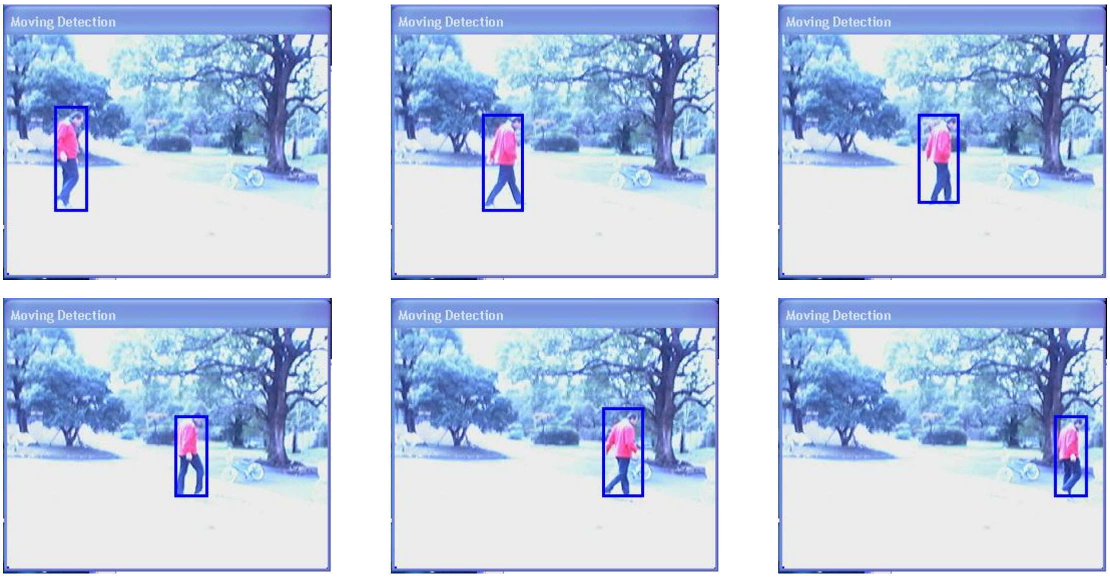


Figure 3.7. Single moving object tracking with resolution factor = 1/8.

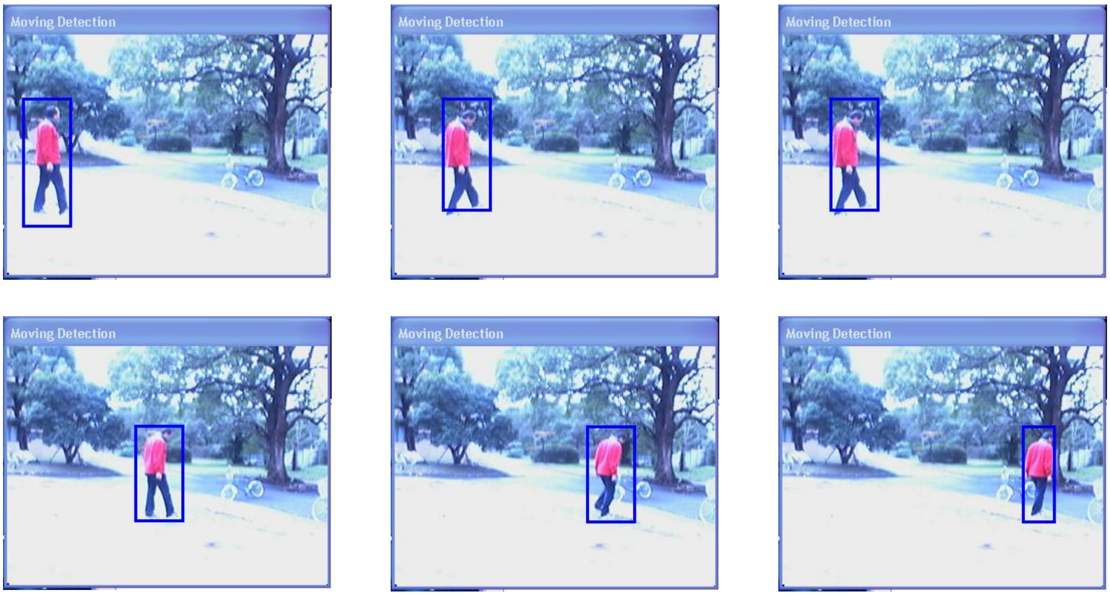


Figure 3.8. Single moving object tracking with resolution factor = 1/16.

The following results, Fig. 3.9 and Fig. 3.10, show two moving objects tracking. In those figures, we successfully tracked the moving objects based on frame difference on low resolution image in resolution factor 1/8 and 1/16, respectively. The image noise caused by wavering trees or fake motion is completely removed in the low resolution image.

Figure 3.9. Two moving objects tracking with resolution factor = 1/8

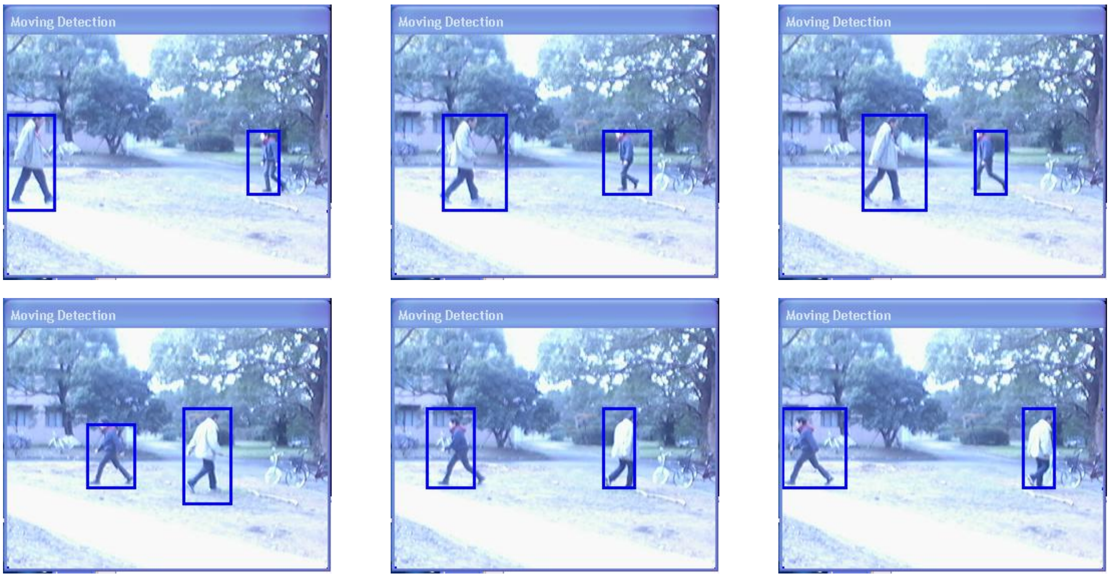


Figure 3.10. Two moving objects tracking with resolution factor = 1/16

3.3 OBJECT TRACKING

On the previous section, we described the object detection method based on frame difference on various image resolutions. We have given the successfully comparative result based on different image resolution.

After the object detection is achieved, the problem of establishing a correspondence between object masks in consecutive frames should arise. Indeed, initializing a track, updating it robustly and ending the track are important problems of object mask association during tracking. Obtaining the correct track information is crucial for subsequent actions, such as object identification and activity recognition. Tracking process can be considered as a region mask association between temporally consecutive frames and estimating the trajectory of an object in the image plane as it moves around a scene. In this thesis, we use block matching technique for this purpose. We consider tracking the interest object from multiple objects appearing in the scene. The entire process of the tracking methodology is illustrated in Fig 3.11. The details of the tracking mechanism of the interest moving object are described in the following sections.

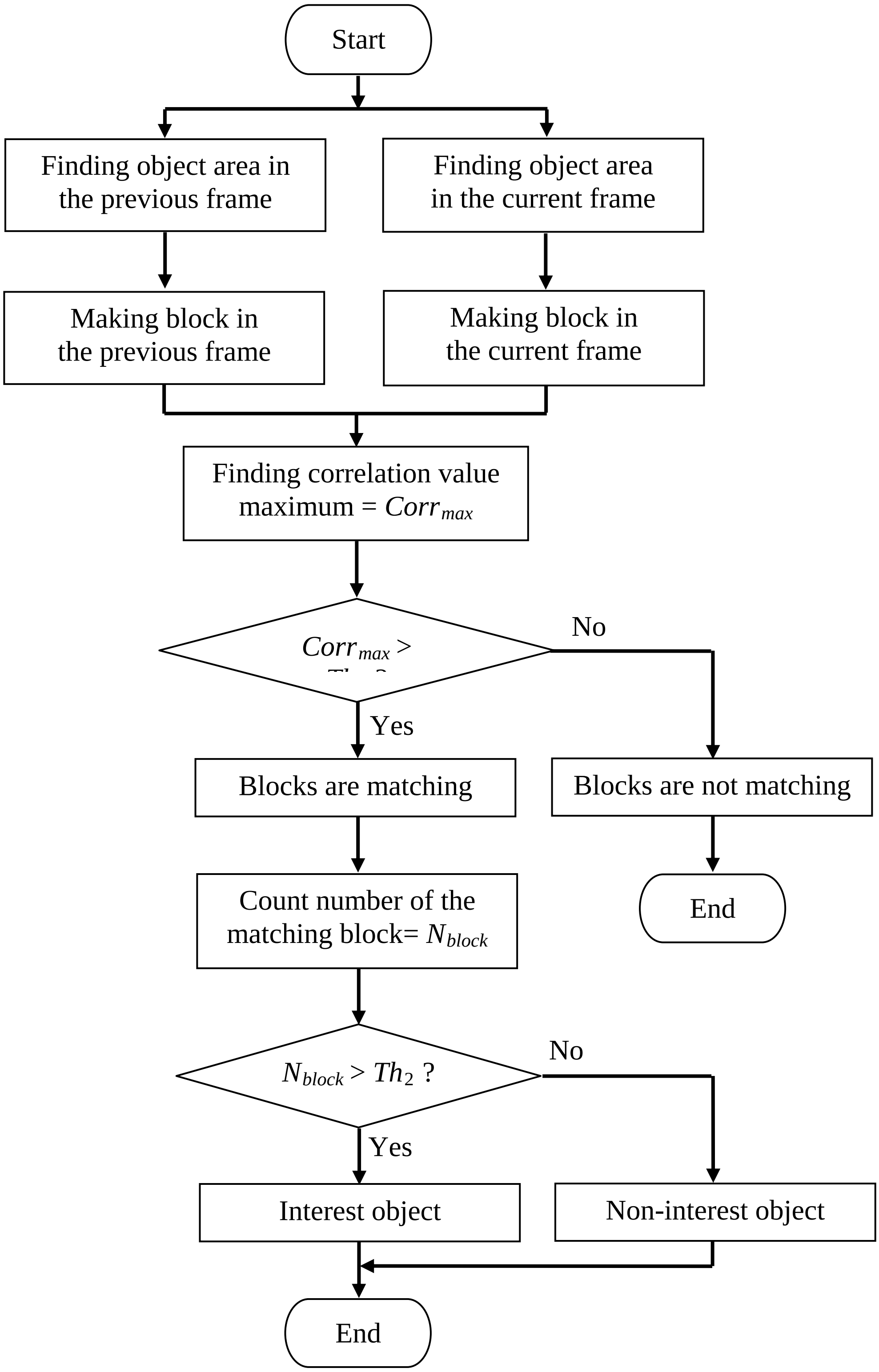


Figure 3.11. Flow of the interest object tracking method based on block matching.

3.3.1 Block Matching Based on PISC Image

Block matching is a technique for tracking the interest moving object among the moving objects emerging in the scene. In this thesis, the blocks are defined by dividing the image frame into non-overlapping square parts. The blocks are made based on peripheral increment sign correlation (PISC) image [30, 46] that considers the brightness change in all the pixels of the blocks relative to the considered pixel. Fig. 3.12 shows the block in PISC image with block size is 5×5 pixels, i.e, one block consists of 25 pixels. The blocks of the PISC image in the previous frame are defined as shown in Eq. (3.6). Similarly, the blocks of the PISC image in the current frame are defined in Eq. (3.7). To determine the matching criteria between the blocks in two successive frames, we evaluate using correlation value that express in Eq. (3.8). This equation calculates the correlation value between block in the previous frame and the current one for all pixels in the block. The high correlation value shows that the blocks are matched each other. The interest moving object is determined when the number of matching blocks in the previous and current frame are higher than the certain threshold value. The threshold value is obtained experimentally.

$$b_{np} = \begin{cases} 1, & \text{if} \quad f_{np} \geq f(i,j) \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

$$b'_{np} = \begin{cases} 1, & \text{if} \quad f_{np} \geq f(i,j) \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

$$corr_n = \sum_{p=0}^{N} b_{np} * b'_{np} + \sum_{p=0}^{N} (1-b_{np}) * (1-b'_{np}) \tag{3.8}$$

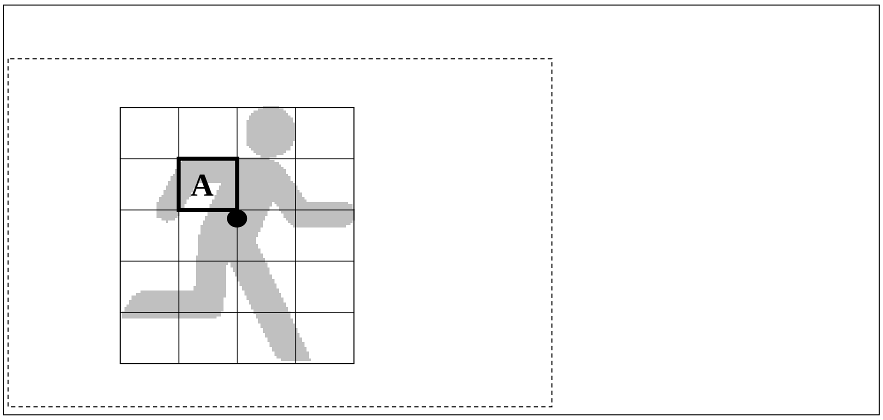
where : b and b' are the block in previous and current frame, n is block number and N is number of pixels on each block, respectively.

$f_{i-2,j-2}$	$f_{i-1,j-2}$	$f_{i,j-2}$	$f_{i+1,j-2}$	$f_{i+2,j-2}$
$f_{i-2,j-1}$	$f_{i-1,j-1}$	$f_{i,j-1}$	$f_{i+1,j-1}$	$f_{i+2,j-1}$
$f_{i-2,j}$	$f_{i-1,j}$	$f_{i,j}$	$f_{i+1,j}$	$f_{i+2,j}$
$f_{i-2,j+1}$	$f_{i-1,j+1}$	$f_{i,j+1}$	$f_{i+1,j+1}$	$f_{i+2,j+1}$
$f_{i-2,j+2}$	$f_{i-1,j+2}$	$f_{i,j+2}$	$f_{i+1,j+2}$	$f_{i+2,j+2}$

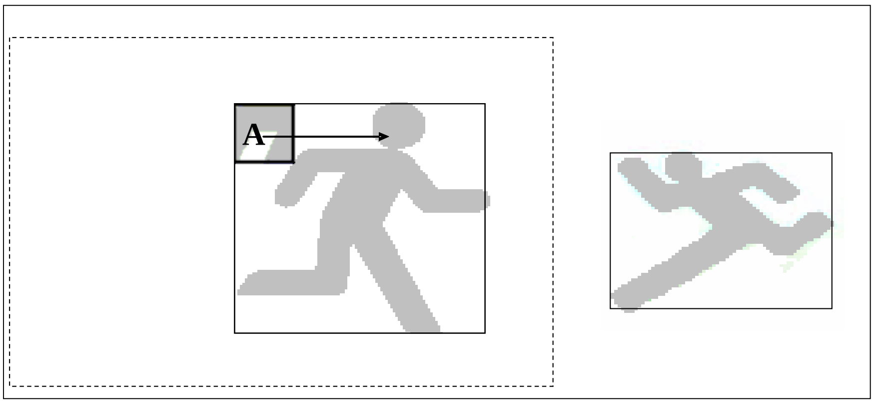
Figure 3.12. PISC image for block size 5 × 5 pixels.

3.3.2 Tracking the Interest Object

The tracking method used in this thesis is described as following. After defining the blocks on the previous frame and the current one, the next process is matching process between each block on each frame. The matching process is illustrated in Fig. 3.13. Firstly, based on the previous object detection process, the blocks and the tracking area are made only in the area of detected moving object to reduce the processing time. We make the blocks (block *A*) with size 9×9 pixels on the previous frame. We assume that the object coming firstly will be tracked as the interest moving object. Then, the block *A* on the previous frame will search the matching blocks in each block of the current frame based on the correlation value as express in Eq. (3.8). On the current frame, the interest moving object is tracked when the object has number of matching blocks greater than certain threshold. When that matching criteria is not satisfied, the above matching process is performed again by enlarging the tracking area (the rectangle area with dash line). The blocks still are made inside the area of moving object. When the interest moving object still cannot be tracked, then the moving object is categorized as non-interest moving object and the tracking process is begun again from the beginning.



(a)



(b)

Figure 3.13. Matching process between frames.

(a) previous frame, (b) current frame

3.4 OBJECT IDENTIFICATION

The object identification is the last stage of our tracking system. In this stage, firstly, we will explain about features extraction for object detection [93]. In this thesis, the extracted features are divided into the following two types, color and spatial information of the moving objects.

3.4.1 Spatial Feature Extraction

The feature of objects extracted in the spatial domain is the position of the tracked object. The spatial information combined with the features in time domain represents the trajectory of the tracked object, so we can estimate the movement and speed of the moving objects that we tracked. Therefore, the features of spatial domain are very important to object identification. The bounding box defined in Eq. (3.9) is used as spatial information of moving objects.

After getting the interest moving object, we extract the interest moving object by using a bounding box. The bounding box can be determined by computing the maximum and minimum value of x and y coordinates of the interest moving object according to the following equation:

$$\begin{aligned} B^i_{min} &= \{(x^i_{min}, y^i_{min}) \mid x, y \in O^i\} \\ B^i_{max} &= \{(x^i_{max}, y^i_{max}) \mid x, y \in O^i\} \end{aligned} \tag{3.9}$$

where O^i denotes the set of the coordinate of points in the interest moving object i, $B^i{}_{min}$ is the left-top corner coordinates of the interest moving object i and $B^i{}_{max}$ is the right-bottom corner coordinates of the interest moving object i, respectively. Fig. 3.14 shows an example of the bounding box of the object tracking.

(a) (b)

Figure 3.14. Bounding box of moving object.

(a) single moving object, (b) two moving objects

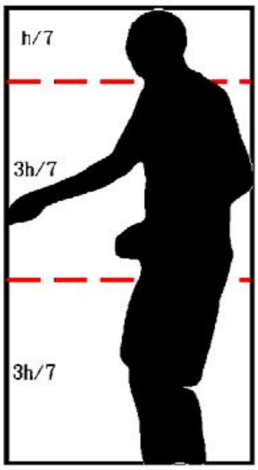


Figure 3.15. Definition of a human body ratio.

3.4.2 Color Feature Extraction

The color feature extracted from the object is RGB color space as the RGB color information can be obtained from video capture device directly. We extract the information from upper and lower part of the object to obtain more color information for identification. The ratio of these three parts can be defined as shown in Fig. 3.15. However, in this thesis, we only calculate the color information of the upper and lower part excluding the head part of the object. The first color information calculated is mean value of each human body part as calculated by Eq. (3.10) for upper part and Eq. (3.11) for lower part. The mean value is calculated for each color component of RGB space.

$$\mu_{f_k}^{O_U^i} = \frac{\sum_{x=x_{min}^i}^{x_{max}^i} \sum_{y=y_{min}^i}^{y_{max}^i} f_k(x,y)}{\#O_U^i} \tag{3.10}$$

$$\mu_{f_k}^{O_L^i} = \frac{\sum_{x=x_{min}^i}^{x_{max}^i} \sum_{y=y_{min}^i}^{y_{max}^i} f_k(x,y)}{\#O_L^i} \tag{3.11}$$

where i is number of the moving objects and (x, y) is the coordinate of pixels in moving object. $\left(x_{max}^i, y_{max}^i\right)$ and $\left(x_{min}^i, y_{min}^i\right)$ are the coordinates of the bounding box of moving object i, $f_k(x,y)$ denotes pixel value for each color component in RGB space of the current frame, O_U^i and O_L^i denote the set of coordinates of upper and lower part of human body of moving object i and $\#O_i$ is the number of pixels of moving object i.

The standard deviation has proven to be an extremely useful measure of spread in part because it is mathematically tractable. Standard deviation is a statistical term that provides a good indication of volatility. It measures how widely the values are dispersed from the average. Dispersion is the difference between the actual value and the average value. The larger the difference between the actual color and the average color is, the higher the standard deviation will be, and the higher the volatility. We can extract more useful color features by computing the dispersed color information from upper and lower part of body as shown in Eq. (3.12) for the upper part and Eq. (3.13) for the lower part.

$$SD_{f_k}^{O_U^i} = \sqrt{\frac{\sum_{x=x_{min}^i}^{x_{max}^i} \sum_{y=y_{min}^i}^{y_{max}^i} \left(f_k(x,y) - \mu_{f_k}^{O_U^i} \right)^2}{\# O_U^i}} \tag{3.12}$$

$$SD_{f_k}^{O_L^i} = \sqrt{\frac{\sum_{x=x_{min}^i}^{x_{max}^i} \sum_{y=y_{min}^i}^{y_{max}^i} \left(f_k(x,y) - \mu_{f_k}^{O_L^i} \right)^2}{\# O_L^i}} \tag{3.13}$$

where $SD_{f_k}^{O_U^i}$ and $SD_{f_k}^{O_L^i}$ denote the standard deviation of each color component of *RGB* space for upper and lower part of the human body of moving object *i*, respectively.

3.4.3 Object Feature Similarity

After the feature extraction, we can represent the moving object *i* by the following feature vectors;

$$F^i = \left(\mu_{f_k}^{O_U^i}, \mu_{f_k}^{O_L^i}, SD_{f_k}^{O_U^i}, SD_{f_k}^{O_L^i}, B_{min}^i, B_{max}^i \right). \tag{3.14}$$

To identify a moving object, a feature queue is created to save the features of the moving objects. When a new object enters the system, it will be tracked and labeled, and the features of the object are extracted and recorded into the queue. Once a moving object *i* is detected, the system will extract the features F^i of the object *i* and identify it from the identified objects in the queue by computing the similarity $S(F^i, F^j)$, $j = 1 \ldots n$, where *j* is one of the *n* identified objects. The similarity, $S(F^i, F^j)$ is computed as in Eq. (3.15),

$$\begin{aligned} S(F^i, F^j) = {} & Mc\left(\left| \mu_{f_k}^{O_U^i} - \mu_{f_k}^{O_U^j} \right| \right) + Mc\left(\left| \mu_{f_k}^{O_L^i} - \mu_{f_k}^{O_L^j} \right| \right) \\ & + Msd\left(\left| SD_{f_k}^{O_U^i} - SD_{f_k}^{O_U^j} \right| \right) + Msd\left(\left| SD_{f_k}^{O_L^i} - SD_{f_k}^{O_L^j} \right| \right) \\ & + 0.5Mp\left(\left| B_{min}^i - B_{min}^j \right| \right) + 0.5Mp\left(\left| B_{max}^i - B_{max}^j \right| \right) \end{aligned} \tag{3.15}$$

where *Mc* and *Msd* are the membership function of color information and standard deviation as defined in Eq. (3.16) and Eq. (3.17), *Mp* is the membership function of spatial information as defined in Eq. (3.18).

$$Mc(x) = \begin{cases} 1 - x/Thr & \text{if } x < Thr \\ 0 & \text{if } x \geq Thr \end{cases} \tag{3.16}$$

$$Msd(x) = \begin{cases} 1 - x/Thr & \text{if } x < Thr \\ 0 & \text{if } x \geq Thr \end{cases} \tag{3.17}$$

$$Mp(x) = \begin{cases} 1 - 3x/W & \text{if } x < W/3 \\ 0 & \text{if } x \geq W/3 \end{cases} \tag{3.18}$$

where *Thr* is threshold which is obtained experimentally and *W* is the width or height of the image frames. We compare the features of detected object with those of the objects in the feature queue. The one with the maximum similarity is identified as the same object. The whole object identification flow is shown in Fig. 3.16.

3.5 EXPERIMENTAL RESULTS

We have done the experiments using a video camera in outdoor environment and real time condition. The experiment is performed in 2.54 [GHz] Pentium 4 PC with 512 MB memory. The image resolution is 320 × 240 [pixels]. The size of each block is 9 × 9 [pixels]. The experimental results are shown in Fig. 3.17–Fig. 3.19. The rectangle area on the object shows the tracked object. The identification result is shown in Table 1. In the experimental results, we can extract the moving objects on the successive frame successfully and identification rates of 92.8% were achieved.

In our experiment, we tracked the interest object from two and three moving objects that occluded between each other when objects move in the same and different direction. We assumed that the first moving object emerging in the scene as the interest moving object. We did the experiments in three conditions. In the first condition, we tracked the interest moving object when it is occluded by another moving object when they move in the different direction as shown in Fig. 3.17. In the second condition, we tracked the interest moving object when it is occluded by another moving object when they move in the same direction as shown in Fig. 3.18. In the third condition, we tracked the interest moving object when three moving object appear in the scene as shown in Fig. 3.19.

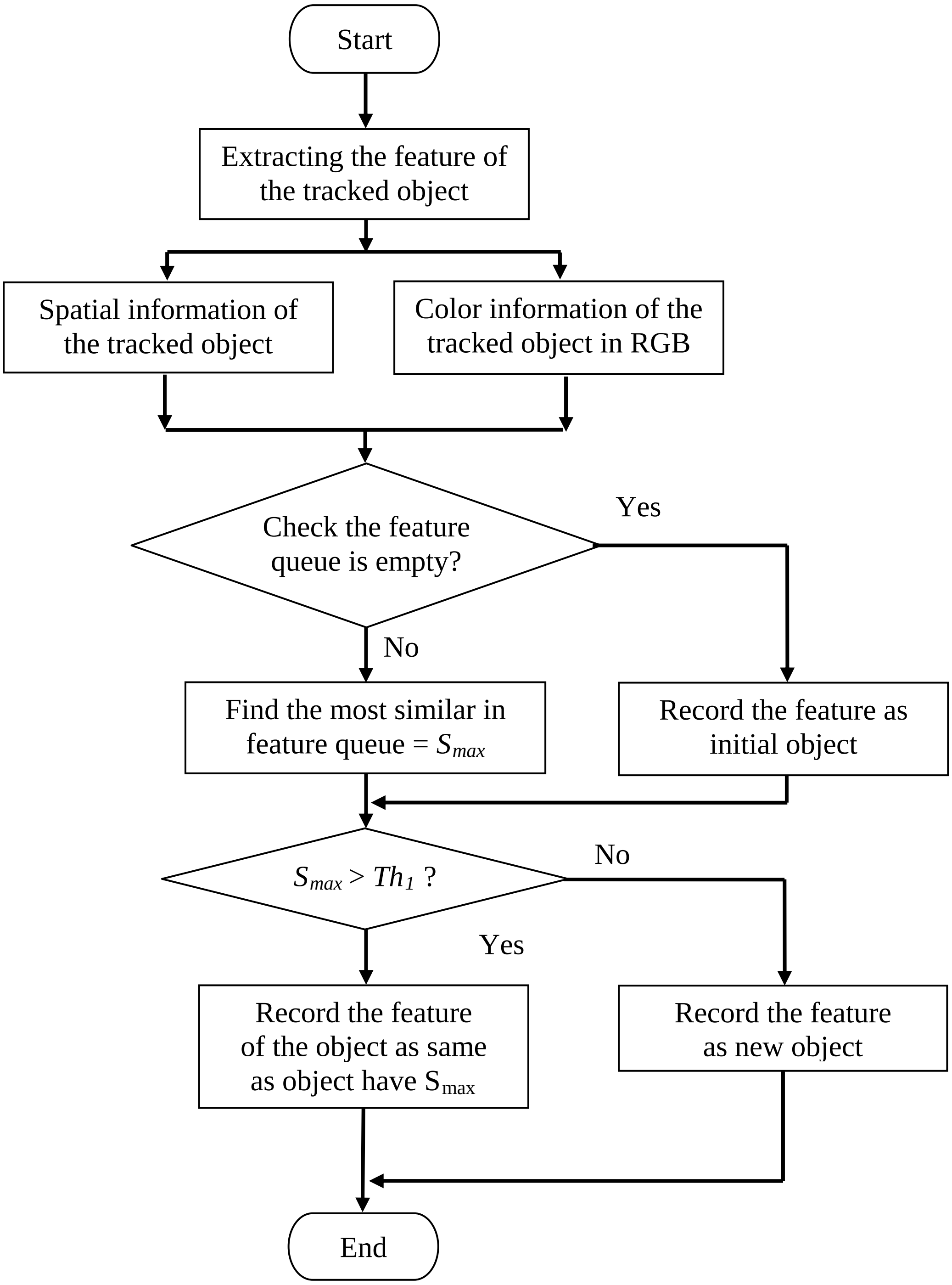


Figure 3.16. Flow chart of object identification.

On the first case (Fig. 3.17), at first, the man wearing the white shirt enters the scene from the left side. This object is successfully detected as the interest moving object. While the first object is being tracked, another object (man wearing the blue shirt) enters the scene from the right side. They move in the different direction and occlude each other in the middle of the scene. We successfully track the first moving object as the interest moving object as our assumptions while the other moving object is not tracked.

On the second case (Fig. 3.18), at first, the man wearing the blue shirt enters the scene from the left side. This object is successfully detected as the interest moving object. Then on the next frame, the man wearing the white shirt enters the scene from the same side. They move in the same direction and occlude each other in the middle of the scene. We successfully track the first moving object as the interest moving object as our assumptions while the other moving object is not tracked.

On the third case (Fig. 3.19), at first, the man wearing the white shirt enters the scene from the right side. This object will be tracked as the interest moving object. Then on the next frame, another man enters the scene from right side. They move in the same direction. The third man enters the scene from the left side. They occlude each other in the middle on the scene. We successfully track the first moving object as interest moving object as our assumptions while the other moving objects are not tracked.

To evaluate our proposed method, we performed the object identification method based on spatial and color information of the tracked object. Table 3.1 shows the result of the object identification. From the table, we notice that some objects are not correctly identified in some frames of each experiment. The wrong identification occurs in two types.

First, it occurs when the moving object just enters or leaves the scene. When the moving object is just entering the scene, the system can detect the moving object. However, the extracted features of the moving object in this case cannot represent the moving object very well, because only partial of the features of the moving objects are extracted. This error is illustrated in Fig 3.20.

The second error occurs when the moving object is slowing down. In this situation, the frame difference of the object becomes smaller. Therefore, only smaller bounding box and less moving pixels are obtained. On this condition, the extracted features will lose its representative. This error is illustrated in Fig. 3.21.

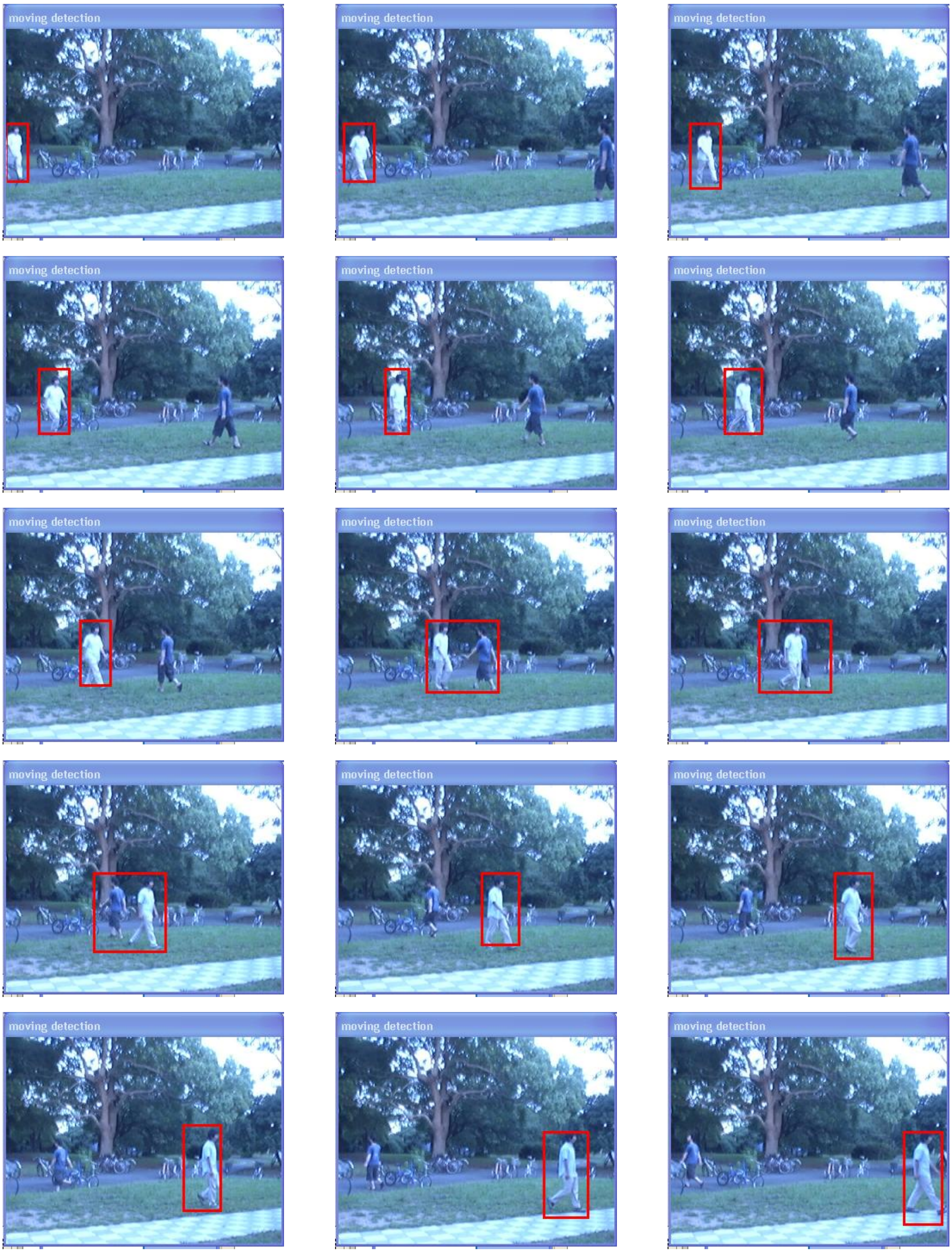


Figure 3.17. Two moving objects occlude in different direction.

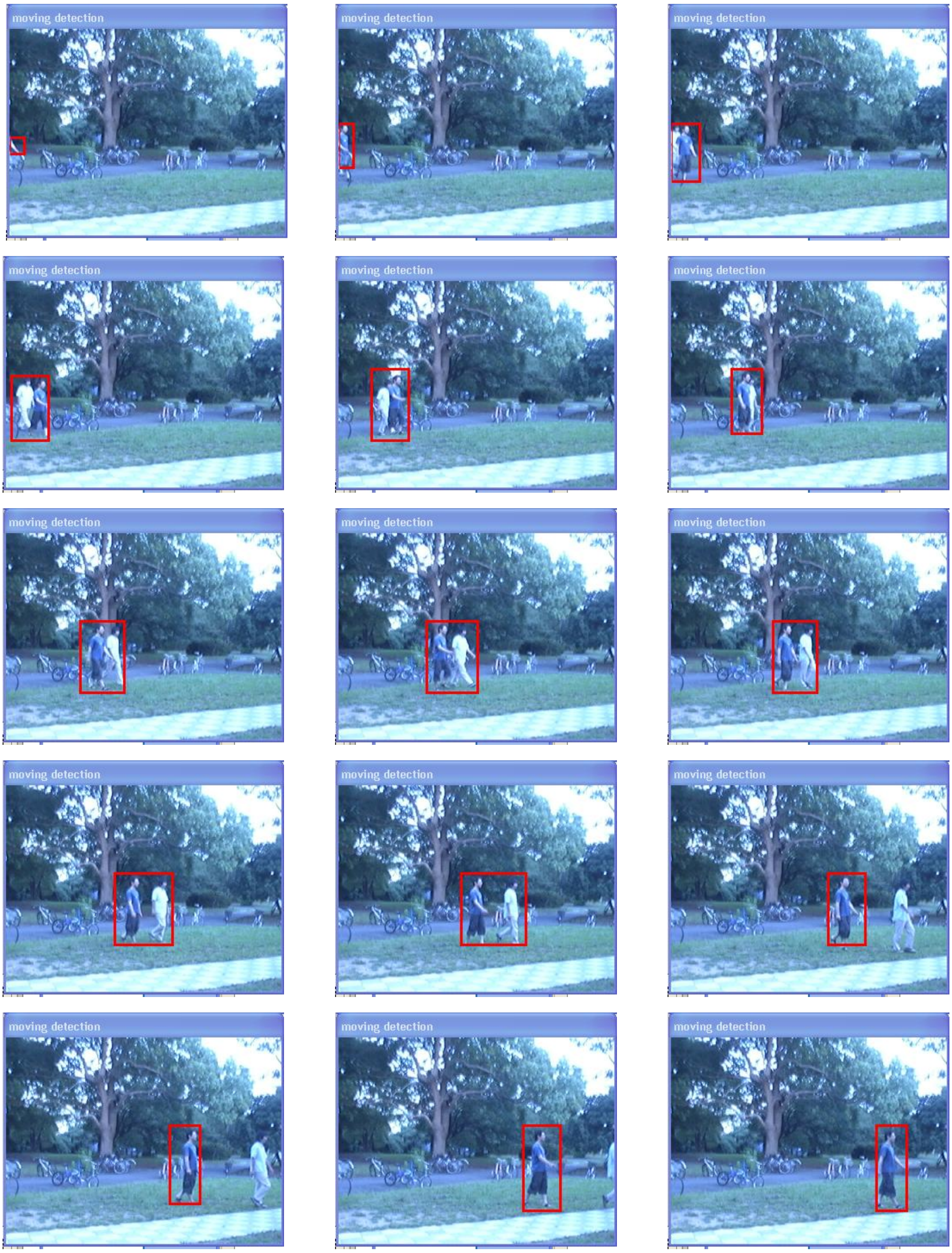


Figure 3.18. Two moving objects occlude in same direction.

Figure 3.19. Three moving objects appear in the scene.

Experiment	Object detected	Correct identification	Identification rates [%]
1	126	117	92.8
2	148	136	91.9
3	154	141	91.6

Table 3.1 Object identification results.

Figure 3.20. First error identification.

Figure 3.21. Second error identification.

3.6 CONCLUSIONS

This chapter presented a method for detecting the moving object employing frame difference on low resolution image, tracking the interest moving object among the moving objects emerging in the background employing block matching technique based on peripheral increment sign correlation image for and identifying the moving objects employing color and spatial information of the tracked object. The experiment results and data show the effectiveness and the satisfaction of the proposed methods. Using our method, we can achieve the identification rate of 92.1[%] in average.

However, the proposed method still has limitations. The limitations can be investigated as followings. Firstly, the detection method based on frame difference on low resolution image has a limitation when the moving object is too small to be detected. It is occurred because the low resolution image removes the small moving objects emerging in the background. To overcome this limitation, we can add another method such as skin color detection. By using this method, even if the moving object is too small, it can still be detected based on the skin color of the object. Secondly, the block matching technique has successfully tracked the interest moving object in the occlude condition. However, when the moving objects appear in the same time, we cannot judge any object to be an interest object. Moreover, when the interest moving object is covered by the occluded object, the image information of the interest moving object cannot be read by the camera. This condition cause the system cannot recognize the interest moving object. Those limitations can be solved by adding other information to the interest moving object such as flow of moving object based on optical flow, dimension or another feature and also we can add the color information to each object. So whenever the objects appear, they have their own model that different from each other. And we can track them based on the model.

Thirdly, color and spatial information method show the high correct identification rate. However, the system still cannot identify the objects sometimes when they are just entering or leaving the scene. The extracted features in this case are not enough to be used to identify the moving objects. The system also has limitation when the object is moving slowly. In this condition, the inter-frame difference image of the object will become smaller and we will get smaller bounding box and less moving pixels. Therefore, the extracted features will lose its representative. The correct identification rate highly depends on the correctness of the moving object detection and feature representation. This problem can be improved by a better feature selection method and moving object detection method.

By considering those limitations and implement some improvements to our method including speed up the processing time, implement multi camera tracking and implement multi objects tracking, they could lead to some improvements in the tracking system.

CHAPTER IV

OBJECT TRACKING USING MULTI CAMERA UNDER OUTDOOR ENVIRONMENT

CHAPTER IV
OBJECT TRACKING USING MULTI-CAMERA UNDER OUTDOOR ENVIRONMENT

4.1 INTRODUCTION

As mentioned in Chapter 2, various methods for object detection and tracking based on the successive frames have been proposed in the past such as background subtraction, frame difference, optical flow and statistical method. However, those methods still have limitation and problem to detect the moving object. To overcome those problems, in Chapter 3, we proposed the tracking system using a single camera. Our proposed method has implemented in three main steps of tracking system, which are object detection, tracking and identification. On object detection method, we applied frame difference method on low resolution image and we could reduce the noise caused by fake motion in the background. We have tracked the interest object among the moving objects appeared in the background by applying block matching method based on peripheral increment sign correlation (PISC) image [30, 45]. We could obtain the high identification rate by applying color and spatial features on object identification step. However, the proposed method still has a lot of limitations need to overcome such as failing to track the small interest object due to incapability of detection process using low resolution image and limitation of field of view (FOV) of camera due to limitation of single camera view. To overcome those limitations, in this chapter, we propose a new tracking system employing PISC image as image matching to detect the moving object. And in order to increase the FOV of the camera, we implement the proposed method using multi-camera system.

To overcome the limitation of detection method, we propose a new method for object detection employing PISC image. PISC image is an image matching method with robust performance, high accuracy, and high computational efficiency. The PISC image is constructed from only the trend of the brightness changes in the neighborhood of the pixel under consideration. The extraction method proposed in this chapter focuses on discriminating between the similar background areas without being affected by brightness changes due to adverse conditions, by utilizing the robust matching performance of the increment sign correlation procedure. The proposed method is based on applying such a similarity decision to all pixels in the scene.

The other limitation is FOV of camera due to implementation of single camera. To overcome this problem, Cai et al. [25] proposed a tracking method using single moving camera with a substantial degree of rotational freedom to increase the viewing angle to certain degree. However, it complicates the implementation by adding the motion estimation of both the viewing system and the subject of interest, and is still limited in the amount of viewing area. Others literature proposed a multi-camera tracking in overlapping view to cover large environment of tracking [26-30]. The multi-camera system is able to switch between each other when the tracking objects move from FOV of one camera to another FOV. Therefore, the system is possible to track the moving objects continuously and smoothly.

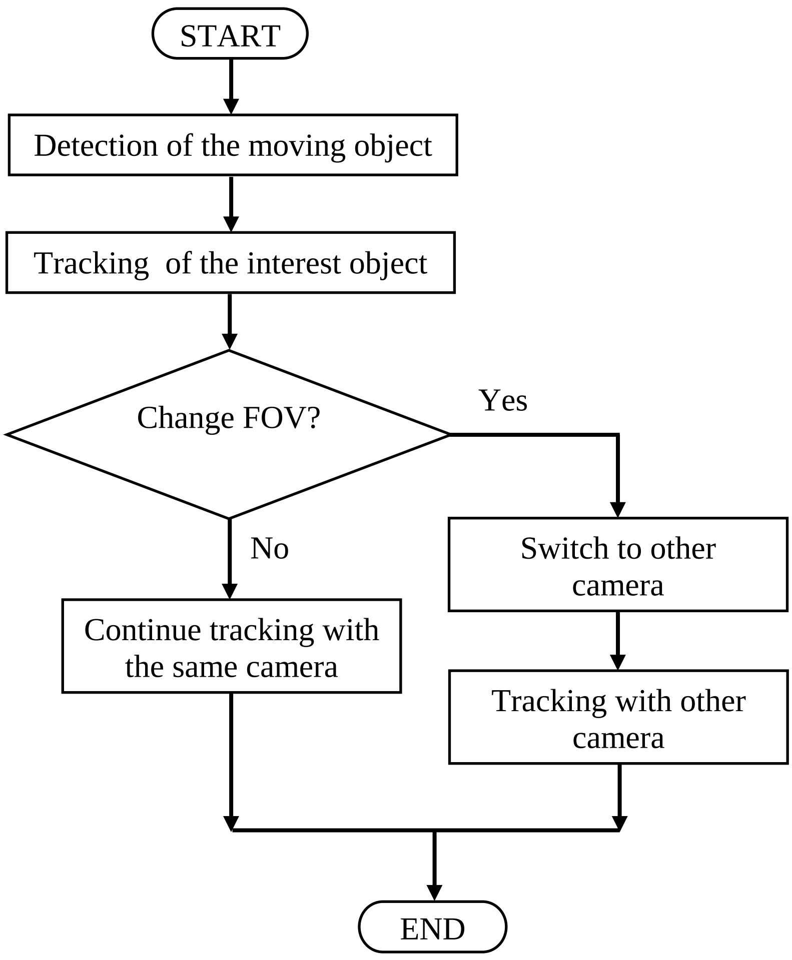


Figure 4.1. Entire process of multi-camera tracking.

In this chapter, we implement our proposed method under multi-camera in overlapping view to get wider FOV. We introduce multi-camera system under local area network (LAN) environment where each camera connected to each PC. The camera can be switched between each other to exchange the image information and track the interest moving object. The entire process of our propose system is described in Fig. 4.1.

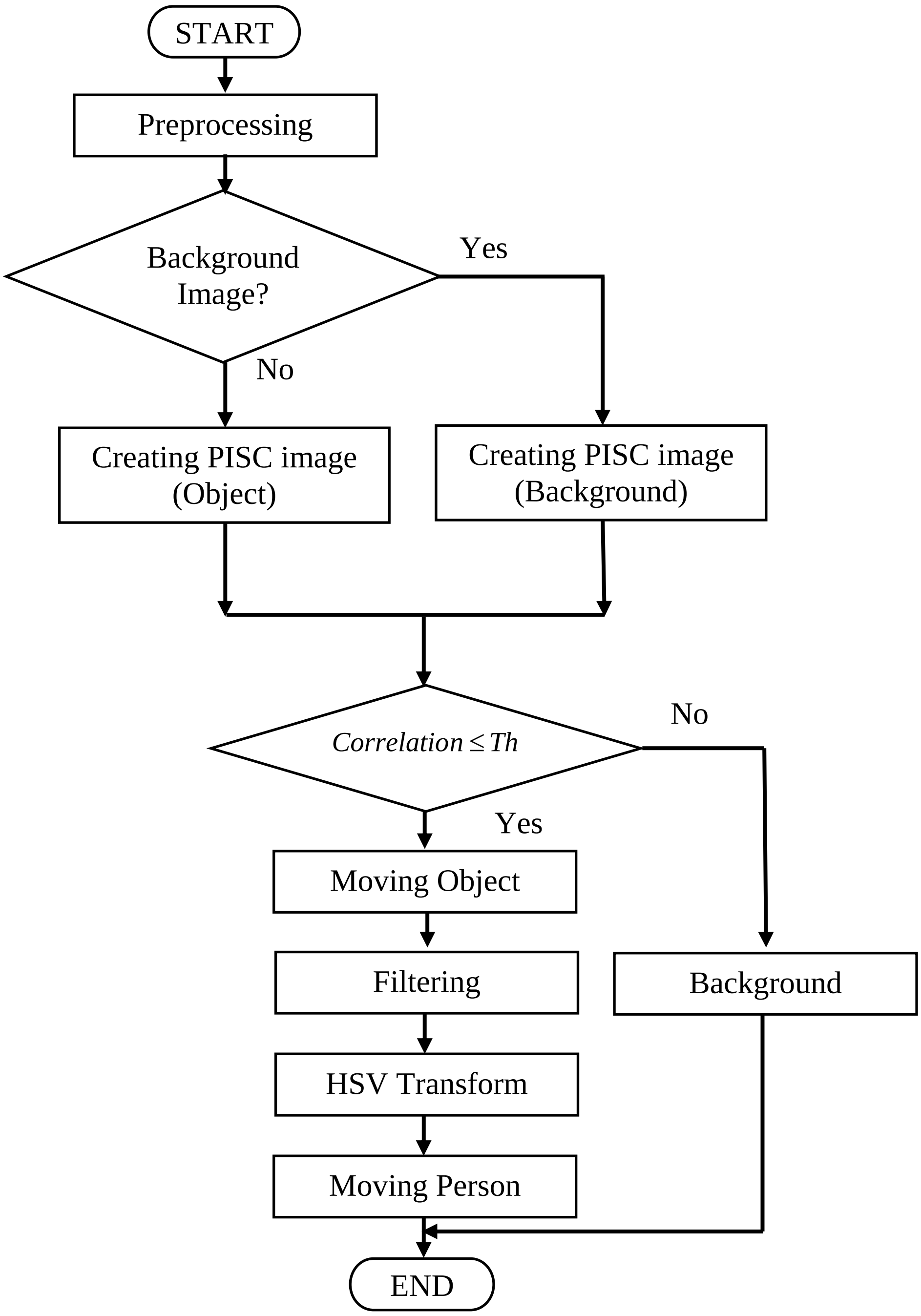


Figure 4.2. The overall object detection procedure.

4.2 OBJECT DETECTION BASED ON PISC IMAGE

Before applying object detection method, to get an accurate object detection result, the preprocessing step is performed. The preprocessing step includes capturing video using a camera, gray scaling process from RGB image to reduce the processing time, image smoothing using median filtering with kernel size 9 × 9 pixels to reduce the noise. Once the background image is acquired from the smoothing image, the moving objects can be separated from the background image using PISC image technique. Then, the image filtering, including morphological filter and image labeling are performed to obtain the accurate object detection result. Furthermore, the HSV color transform is performed as skin color detection to distinguish between moving objects and moving persons. The procedure of the detection method is described in Fig 4.2.

4.2.1 PISC Image for Object Detection

The PISC image takes a value of 1 or 0 according to whether the increment near the considered pixel is positive or negative. This is a logical code representing the trend of brightness change. The increment sign is a code that represents the trend of the brightness change in a certain direction. The correlation coefficient over the whole image is called the increment sign correlation.

The PISC image is used to detect the moving object based on the trend of the brightness changes in the neighborhood of the pixel under consideration. The PISC image consider the brightness changes in the 16-neighborhood of the considered pixel as shown in Fig. 4.3.

In order to detect the moving object in the image sequence, the background image is defined firstly. In the background image $F = \{f_{i,j}\}_{j=1,\cdots,M-1}^{i=1,\cdots,N-1}$ the peripheral increment sign $b_{k(i,j)}$ is defined as,

$$\begin{aligned} b_0(i,j) &= \begin{cases} 1 & (f_{i+2,j} \geq f_{i,j}) \\ 0 & (otherwise) \end{cases} \\ &\vdots \\ b_{15}(i,j) &= \begin{cases} 1 & (f_{i+2,j-1} \geq f_{i,j}) \\ 0 & (otherwise) \end{cases}. \end{aligned} \tag{4.1}$$

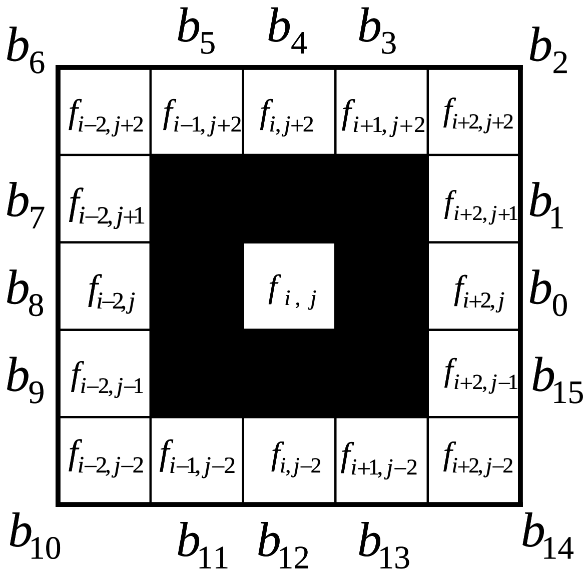
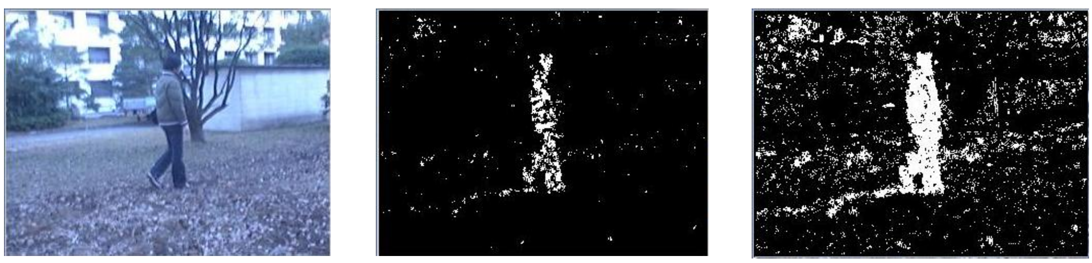


Figure 4.3. Neighborhood for peripheral increment sign.

For any object image $G=\{g_{i,j}\}$ the peripheral increment sign $b'_{k(i,j)}$ (k=0,1,…,15) is similarly defined. In two images, the extent of matching between the increments signs in the 16 directions at the corresponding position is defined as the peripheral increment sign correlation B:

$$B = \frac{1}{16}\sum_{k=0}^{15} c_{k}(i,j) \tag{4.2}$$

$$C_k = b_{k(i,j)} \bullet b'_{k(i,j)} + \left(1 - b_{k(i,j)}\right) \bullet \left(1 - b'_{k(i,j)}\right). \tag{4.3}$$



(a) (b) (c)

Figure 4.4. PISC Image.
(a) original image, (b) PISC image (threshold 0.5) and (c) PISC image (threshold 0.8)

The value of B at each pixel is compared to some threshold and a decision is made whether it is similar or non similar pixel. The PISC image is defined as binary image $\{I_{i,j}\}$ that defined as follows,

$$I_{i,j} = \begin{cases} 1 & (B \leq B_T) \\ 0 & (otherwise) \end{cases} \tag{4.4}$$

where B_T is threshold. B_T usually takes a value in the range from 0.5 to 1. Fig. 4.4 shows an example of a PISC images in different threshold.

In the next step, we performed a morphological operation to remove the remaining noises in the object. The morphological operation implemented in this chapter is dilation followed by erosion. In dilation, each background pixel that is touching an object pixel is changed into an object pixel. Dilation will add pixels to the boundary of the object and close isolated background pixel. In erosion, each object pixel that is touching a background pixel is changed into a background pixel. Erosion will remove isolated foreground pixels. Morphological operation eliminates background image noises and fills small gaps inside an object. This property makes it well suited to our objective since we are interested in generating object masks which preserve the object boundary.

Fig. 4.5 shows the comparison of object detection between background subtraction image and PISC image. In that figure, we can see that the background subtraction image is greatly affected by the brightness change between object and background, but the PISC image is little affected by the brightness change. A high-density area that observed in that figure corresponds to the emerging object.

(a) (b) (c)

Figure 4.5. Comparison between background subtraction and PISC image.

(a) original image, (b) background subtraction and (c) PISC image

4.2.2 HSV Color Transform

In order to distinguish the moving persons among the moving objects, the HSV color transform is applied. The HSV color transform can be used to detect the skin color of the moving persons. Based on the skin color detection, the moving persons are extracted from any other moving objects. The HSV color transformation is obtained from RGB color by following formula:

$$S = (V - Z) \tag{4.5}$$

$$V = max(R,G,B) \tag{4.6}$$

$$H = \begin{cases} \pi/3(b-g), & R = V \\ \pi/3(2+r-b), & G = V \\ \pi/3(4+g-r), & B = V \end{cases} \tag{4.7}$$

where

$$\begin{gathered} Z = min(R,G,B),\ r = (V-R)/(V-Z),\ g = (V-G)/(V-Z) \\ b = (V-B)/(V-Z),\ 0 \le H \le 360,\ 0 \le S,V \le 1,\ 0 \le R,G,B \le 1. \end{gathered} \tag{4.8}$$

4.3 TRACKING THE INTEREST OBJECT

To track the interest object, we use block matching as we explained in Chapter 3. The object in one frame is considered as an interest object if the object contains a certain number of matching blocks with the blocks of the object in the previous frame. When the interest object appears in one camera, the system will track the object with this camera until it disappears. And when the object appears again on the second camera after disappearance on the first camera, the system will exchange the object information from the first camera to the second camera. Then, the system will track the object with the second camera. On the other word, the system can automatically exchange the information form one camera to others. It is because the system has capability to exchange the information between cameras based on LAN connection. And when the interest object disappears on both FOVs of camera, the system will start to detect a new interest object.

4.4 EXPERIMENTAL SET UP

The system configuration used in this chapter is shown in Fig. 4.6. Each camera is connected to each PC under LAN connection. The camera can be switched between each other when the tracking object moves from FOV of one camera to FOV of another camera. The network programming, called socket programming, is used to communicate between PCs. By specifying IP address of the server from the client sides, the tracked object information can be exchanged between server and clients.

4.5 EXPERIMENTAL RESULTS AND ANALYSIS

We have done the experiments using two cameras connected to each PC under LAN connection in outdoor environment and real time condition. The experimental environment used in our experiment is described in Table 4.1. The image resolution is 320 × 240 [pixels]. The size of each block used in matching process is 9×9 [pixels]. The experimental results are shown in Fig. 4.7 – Fig. 4.10. On each result, the rectangle area on the object shows the tracked object.

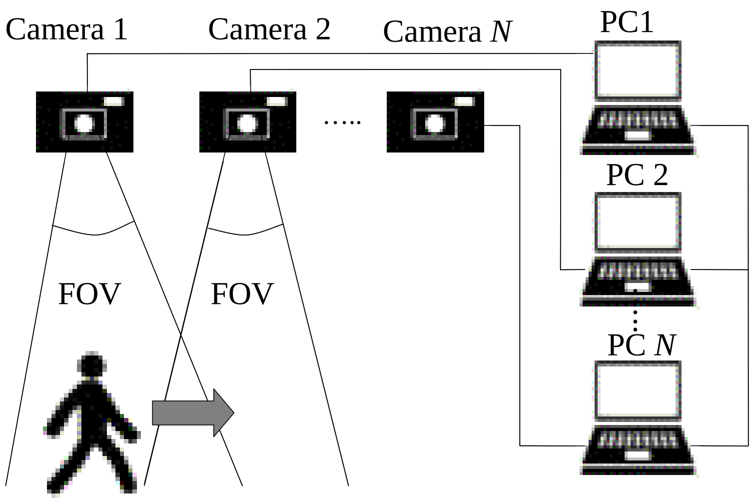


Figure 4.6. Configuration of the system.

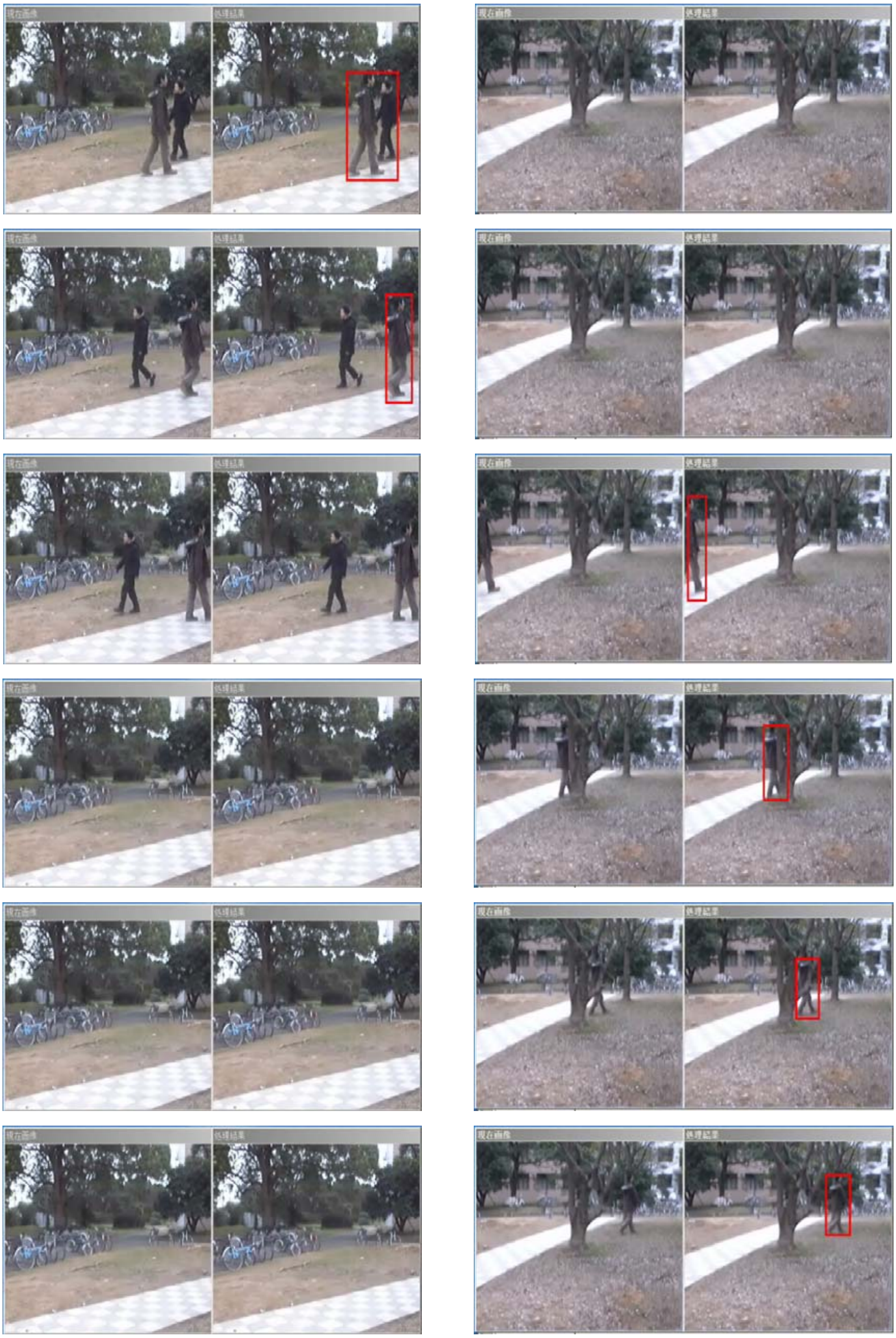
Table 4.1 The experimental environment.

	PC1	PC2
CPU	Intel(R) Pentium(R) 4 3.06 [GHz]	Mobile Intel(R) Pentium(R)4 2.20 [GHz]
Memory	1[GB]	512 [MB]
OS	Windows 2000 Professional	Windows 2000 Professional

In our experiment, we track the interest object from two and three moving objects that occluded between each other when objects move in the same and different direction. We assumed that the first moving object appearing in the scene as the interest moving object. We did the experiments in four conditions. On the first condition, we tracked the interest moving object when it is occluded by another moving object when they move in the different direction as shown in Fig. 4.7. On the second condition, we tracked the interest moving object when it is occluded by another moving object when they move in the same direction as shown in Fig. 4.8. On the third condition, we tracked the interest moving object when three moving object appear in the scene as shown in Fig. 4.9. And the forth condition, we track the interest object when it moves from one camera to other camera and appears again on the first camera as shown in Fig. 4.10.

On the first case (Fig. 4.7), at first, the first object enters the scene of camera 1 from the left side. The system recognizes the first object as the interest moving object and track it. While the first object is being tracked, the second object enters the scene of camera 1 from the right side. They move in the different direction and occlude each other in the middle of the scene. The system successfully tracks the first object as the interest object on camera 1 while the second object is not tracked. After the first object disappears on camera 1, the first object enters and appears on camera 2. The system successfully tracks the interest moving object on camera 2. The system also successfully tracks the object although there is another occlusion by tree.

Figure 4.7. Multi-camera tracking with two objects move on different direction (cont.).

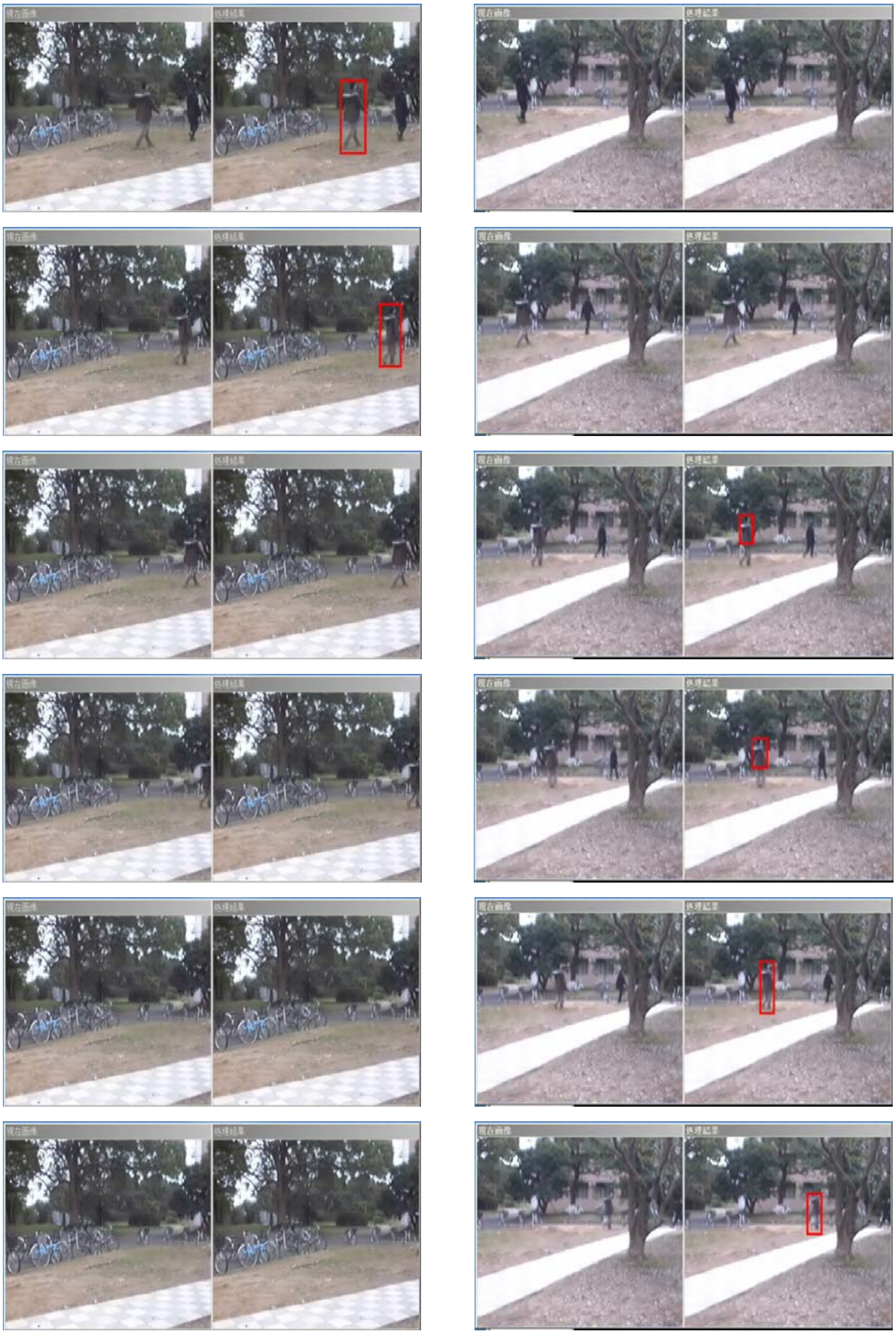


(a) Camera 1 (b) Camera 2

Figure 4.7. Multi-camera tracking with two objects move on different direction.

On the second case (Fig. 4.8), at first, the first object enters the scene of camera 1 from the left side. The system recognizes the first object as the interest moving object and tracks it. While the first object is being tracked, the second object enters the scene of camera 1 from the same side. They move in the same direction and occlude each other in the middle of the scene. The system successfully tracks the first object as the interest object on camera 1 while the second object is not tracked. After the first object disappears on camera 1, the first object enters and appears on camera 2. The system successfully exchanges the tracked object to camera 2.

Figure 4.8. Multi-camera tracking with two objects move on same direction (cont.).



(a) Camera 1 (b) Camera 2

Figure 4.8. Multi-camera tracking with two objects move on same direction.

On the third case (Fig. 4.9), the first object enters the scene of camera 1 from the left side. The system recognizes the first object as the interest moving object and tracks it. While the first object is being tracked, the second object enters the scene of camera 1 from the left side and the third object enters the scene of camera 2 from the right side. They occlude each other in the middle of the scene of camera 1. The system successfully tracks the first object as the interest object on camera 1 while the second and the third object are not tracked. Because we consider only tracking the interest object as the first object appears in any camera, the third object is not tracked on camera 2 although on the camera 2 only the third object appears. The second object also appears on camera 2 for the first time, yet the system does not consider it as interest object and the system does not track it. After some while, the first object disappears on camera 1 and appears on camera 2. The system starts tracking the interest object on camera 2.

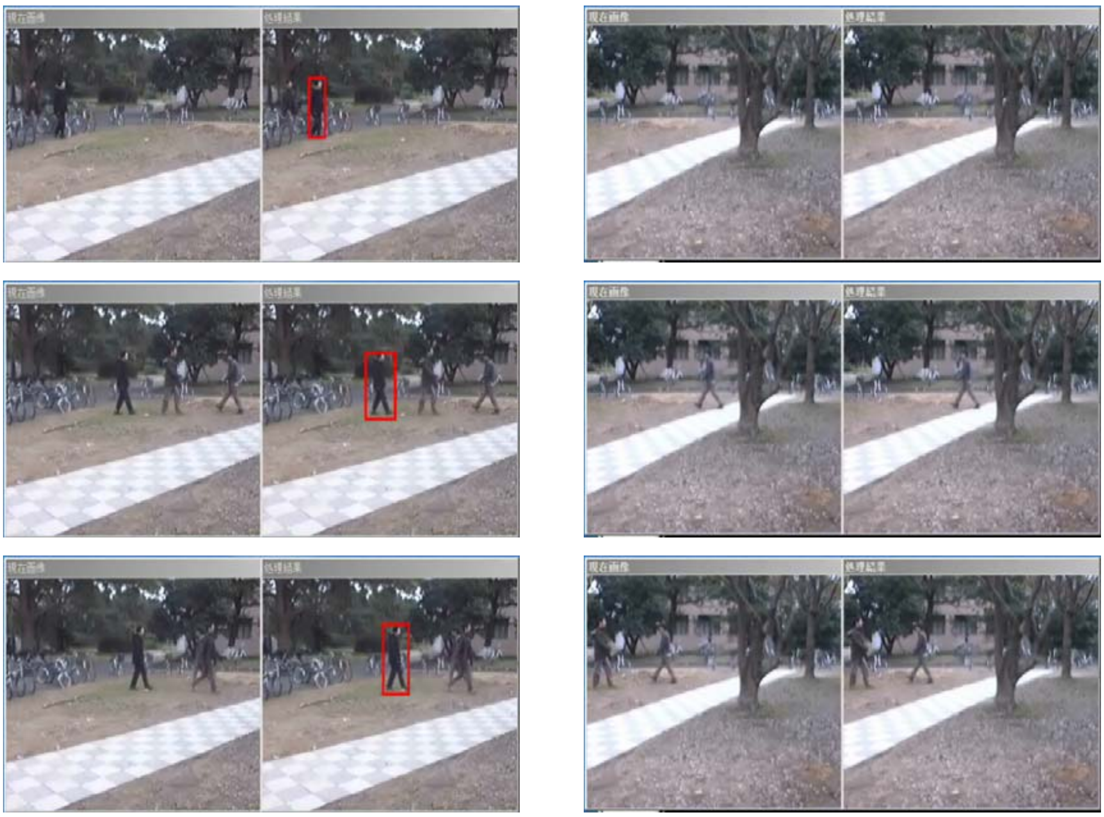
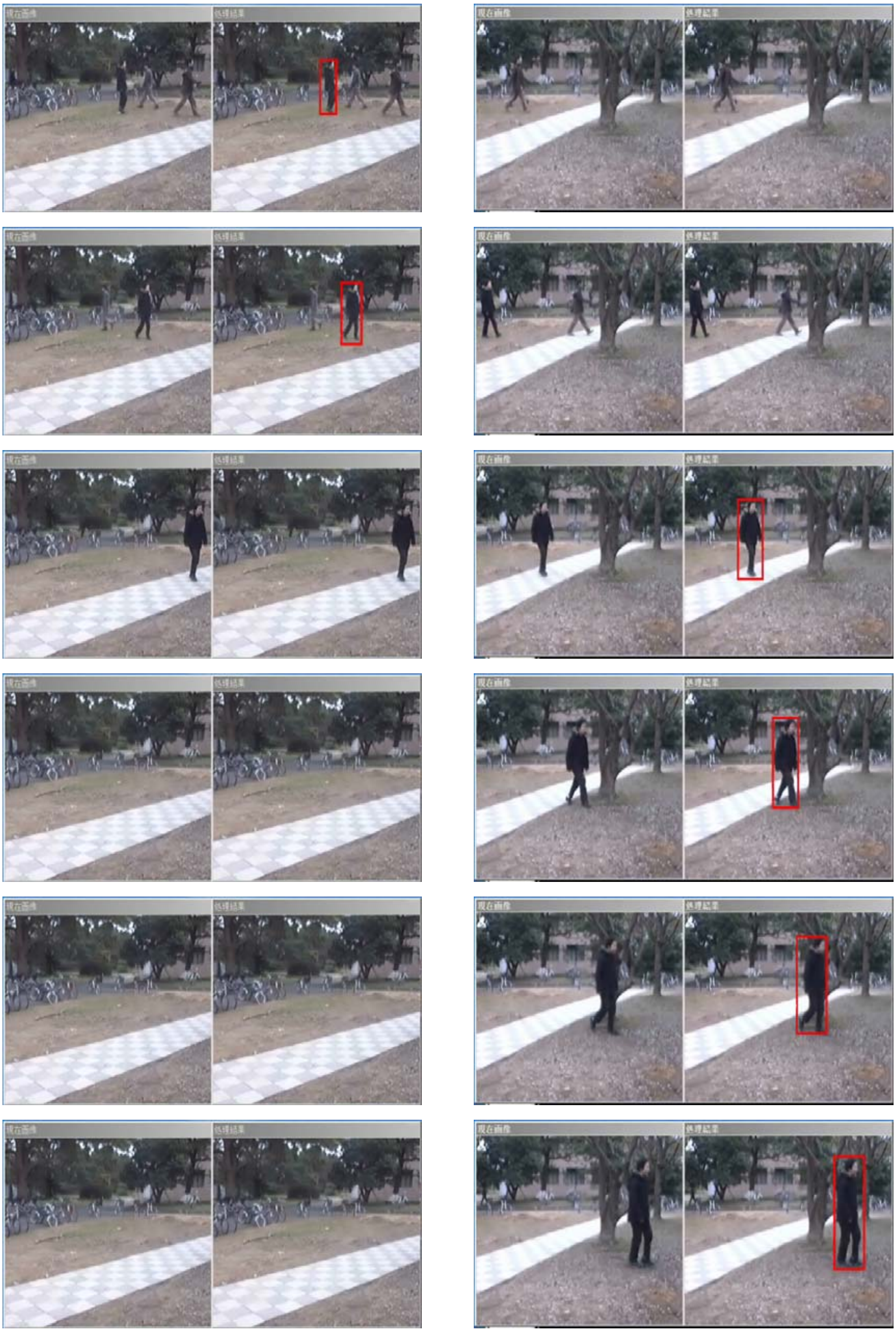


Figure 4.9. Multi-camera tracking with three objects appearing in the scene (cont.).



(a) Camera 1 (b) Camera 2

Figure 4.9. Multi-camera tracking with three objects appearing in the scene.

On the last case (Fig. 4.10), the system will track the interest object when it moves from one FOV to other FOV of camera and return back to the first FOV of camera. In this case, the system recognizes the interest moving object enters camera 1 and track it. The second object appears in camera 2 but it is not considered as interest object and the system does not track it. The objects occlude each other in the middle of the scene of camera 1. After the first object disappears on camera 1, the first object appears on camera 2 and starts to be tracked by the system. After some while, the object is occluded and disappears behind the tree. Our method successfully recovers the tracking system after occlusion. The object is track in camera 2 until it disappears and appears again on camera 1 and the tracking process is started again on camera 1.

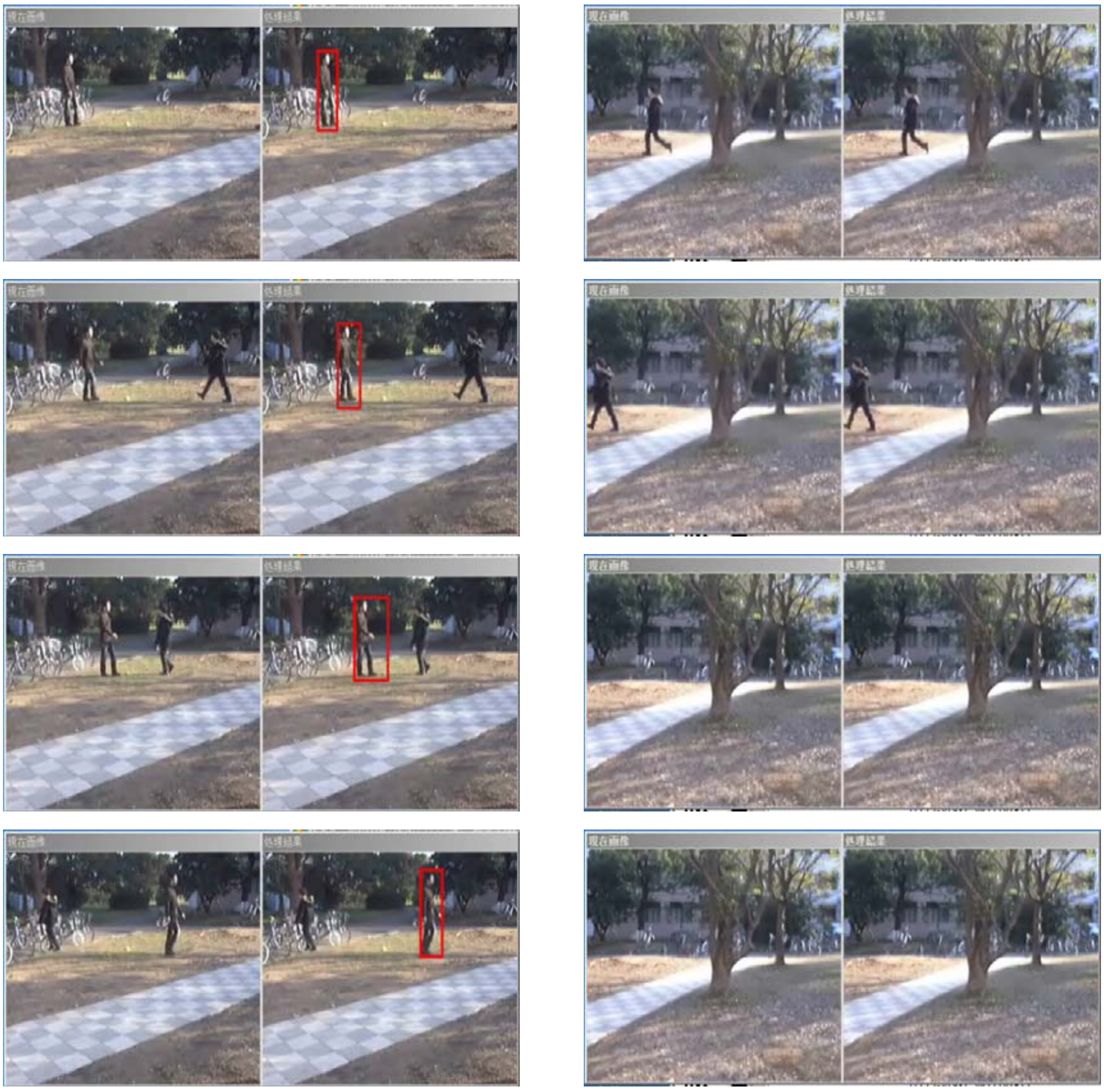
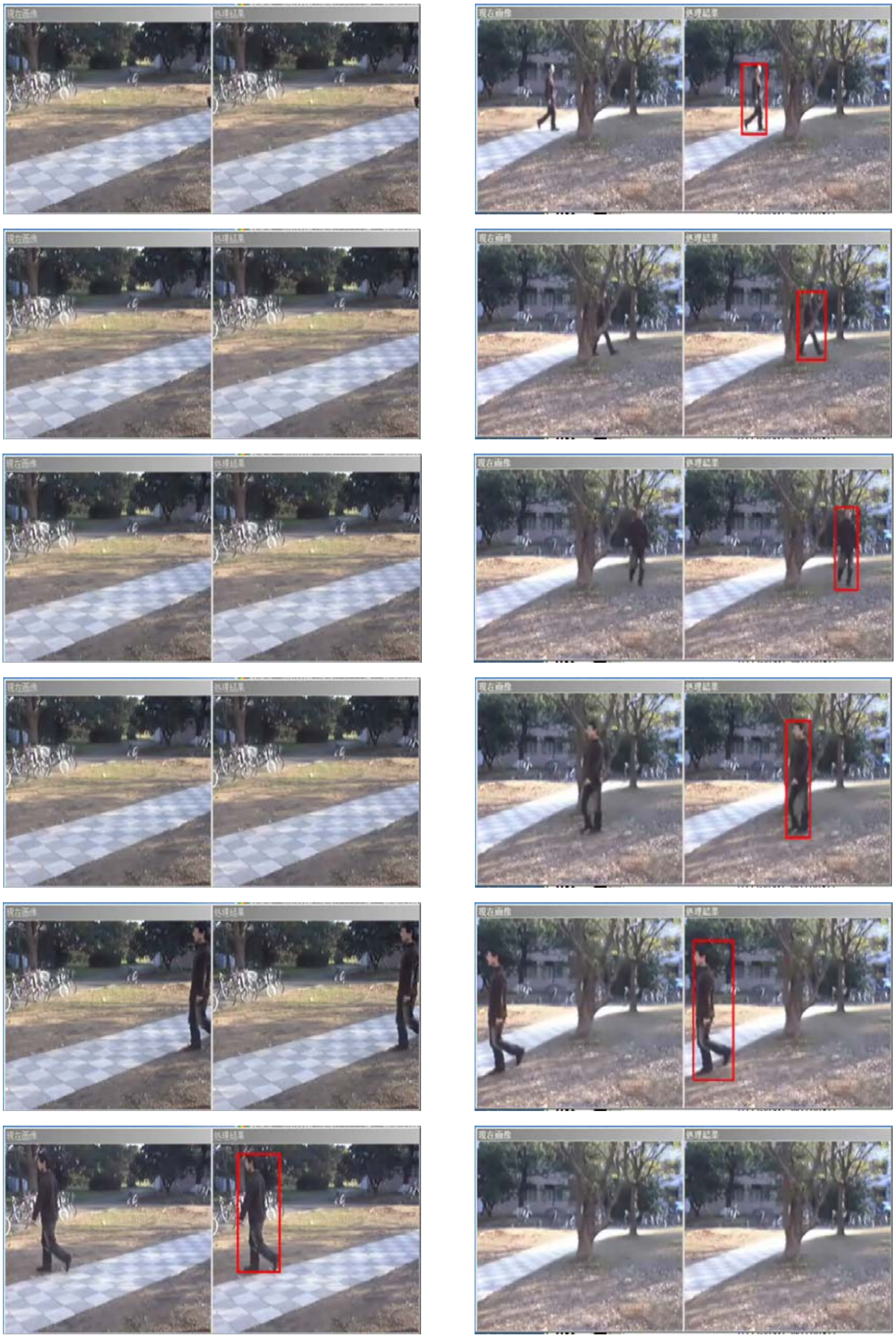


Figure 4.10. Multi-camera tracking when the interest object enters the FOV in the second time (cont.).



(a) Camera 1 (b) Camera 2

Figure 4.10. Multi-camera tracking when the interest object enters the FOV in the second time.

Table 4.2 Evaluation results.

Experiment	TP	FP	TN	FN	TF	DR [%]	Acc [%]
1	48	0	4	1	53	97.96	96.23
2	43	0	6	2	51	95.56	96.07
3	50	0	5	1	56	98.03	98.21
4	37	0	1	1	39	97.37	97.44
Average [%]						97.23	96.98

The experimental results are evaluated based on the following characteristics:

- True Positive (TP)***:*** Number of frames where both ground truth and system results agree to track the presence of one or more objects.
- True Negative (TN)***:*** Number of frames where both ground truth and system results agrees on the absence of any object.
- False Negative (FN)***:*** Number of frames where ground truth contains at least one object while the system does not contain any object.
- False Positive (FP)***:*** Number of frames where system results contain at least one object while ground truth does not contain any object.

From above characteristics, we define the detection rate (DR) and Accuracy (ACC) as $DR = \frac{TP}{TP + FN}$ and $Acc = \frac{TP + TN}{TF}$, where TF is number of frames. The evaluation result is shown on Table 4.2.

From the table, we can notice that our proposed method successfully track the interest object under multi-camera system with high detection rate and high accuracy. However, there is still miss-tracking on some frames of each experiment. We found two conditions that cause the miss-tracking, the exchanging information between camera shown in Fig. 4.11 and the size of the object is too small to be tracked shown in Fig. 4.12.

On the first case, the object has appeared on the FOV of camera 2 while it also still appears in FOV of camera 1. The system has sent the object information to camera

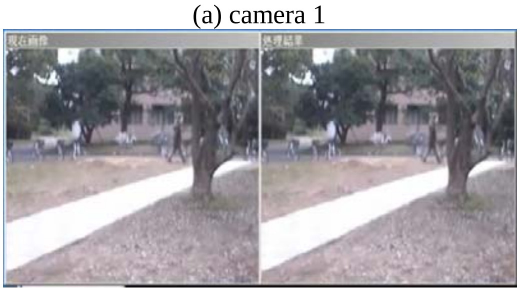
2 and it would be tracked by camera 2. However, the system was unable to track the object with camera 2 because of limited information of the object due to object just enters the FOV of the camera. On the other side (camera 1), the object also could not be tracked because of same condition due to object leaving the FOV of the camera. The second error occurs because the object is too small to detect by PISC image. On this condition, the tracking process fails to track the object

(a) camera 1

(b) camera 2

Figure 4.11. Miss-tracking because of exchanging object information between camera 1 and 2 on experiment 1.

(a) camera 1



(b) camera 2

Figure 4.12. Miss-tracking because of object is too small to be tracked on experiment 2.

4.6 CONCLUSIONS

In this chapter, we proposed a new method for tracking the interest moving object using multi-camera in overlapping view employing PISC image for real time application. Using our new method, the satisfactory tracking results are achieved. The detection results have been improved by PISC image. The limitation of FOV of camera is reduced by using two cameras connected together. Our system can track successfully the interest object in 4 cases of occlusion and changing the FOV of the camera and we can achieve the detection rates of 97.23[%] and accuracy of 96.98[%] in average.

However, the proposed tracking system still has limitations such as miss-tracking because of exchange object information between camera and the size of the object is too small. One of the solutions is implementing the wider range of tracking camera enabled by using more cameras [98, 99] connected together or by using wireless cameras. And also improving the tracking algorithm, such as tracking multi-object by implementing the statistical method, could lead to some improvements in the tracking system.

CHAPTER V

OBJECT TRACKING BASED ON PARTICLE FILTER ALGORTIHM

CHAPTER V
OBJECT TRACKING BASED ON PARTICLE FILTER ALGORITHM

5.1 INTRODUCTION

As mentioned in the previous chapters, the limitation and problem of the tracking system are still remained. This chapter introduces a stochastic approach to minimize those tracking problems. In the stochastic approaches, the tracking problem is formulated as a sequential recursive estimation that is having an estimate of the probability distribution of the target in the previous frame and estimates the target distribution in the new frame using all available prior knowledge and the new information brought by the new frame [100]. The state-space formalism where the current tracked object properties are described in an unknown state vector updated by noisy measurements is very well adapted to model the tracking. The stochastic approaches use the state-space to model the underlying dynamics of the tracking system such as Kalman filter [101, 102] and particle filter [69, 91, 92, 100]. Kalman filter is a common approach for dealing with target tracking in the probabilistic framework. In a linear-Gaussian model with linear measurement, there is always only one mode in the posterior probability density function (pdf). The Kalman filter can be used to propagate and update the mean and covariance of the distribution of the model [101]. But it cannot resolve the tracking problem when the model is nonlinear and non-Gaussian [103]. For nonlinear or non-Gaussian problems, it is impossible to evaluate the distributions analytically and therefore, many algorithms have been proposed to approximate them. The extended Kalman filter can deal to this problem, but still has a problem when the nonlinearity and non-Gaussian cannot be approximated accurately.

To overcome those problems, particle filter has been introduced by many researchers and become popular algorithm to estimate the problem of nonlinear and non-Gaussian estimation framework. The particle filter, also known as sequential Monte Carlo filter [91], is the most popular approach which recursively constructs the posterior pdf of the state-space using Monte Carlo integration. It approximates a posterior probability density of the state such as the object position by using samples or particles. The probability distribution of the state of the tracked object is approximated by a set of particles, which each state is denoted as the hypothetical state of the tracked object and its weight. It has been developed in the computer vision community and applied to tracking problem and is also known as the condensation algorithm [69]. For another particle filter, Bayesian bootstrap filter was introduced [92].

In this chapter, we present a color-based particle filter algorithm to track a moving object under outdoor environment. The particles are predicted by propagating the particles according to a state-space model. The weight is considered as the likelihood of each particle. For this likelihood, we consider the similarity between the color histogram of the target model and the particles. The Bhattacharya distance is used to measure this similarity. And finally, the mean state of the particles is treated as the estimated position of the object.

5.2 PARTICLE FILTER BASED TRACKING

5.2.1 Bayesian Filtering Framework

In this session, we briefly discuss Bayesian approach for state estimation [104]. Suppose we have a nonlinear system described by equations,

$$\begin{aligned} x_{k+1} &= f_k(x_k, \omega_k) \\ y_k &= h_k(x_k, v_k) \end{aligned} \tag{5.1}$$

where, k is the time index, x_k is the state, ω_k is the process noise, y_k is the measurement, and v_k is the measurement noise, respectively. The function $f_k(.)$ and $h_k(.)$ are time-varying non-linear system and measurement equation. The noise sequences $\{\omega_k\}$ and $\{v_k\}$ are assumed to be independent and white with known pdf's. The goal of a Bayesian estimator is to approximate the conditional pdf of x_k based on measurements y_1, y_2,…, y_k. This conditional pdf is denoted as $p(x_k|y_{1:k})$. The first measurement is obtained at $k = 1$, so the initial condition of the estimator is the pdf of x_0, which can be written as $p(x_0) = p(x_0|y_0)$ since y_0 is defined as the set of no measurements. Once we compute $p(x_k|y_{1:k})$ then we can estimate x_k.

The Bayesian estimator will find a recursive way to compute the conditional pdf $p(x_k|y_{1:k})$. Before we find this conditional pdf, we will find the conditional pdf $p(x_k|y_{1:k-1})$. This pdf can be calculated as

$$p(x_k \mid y_{1:k-1}) = \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid y_{1:k-1}) dx_{k-1} . \tag{5.2}$$

The second pdf on the right side of the above equation is not available yet, but it is available at the initial time ($p(x_0) = p(x_0|y_0)$). The first pdf on the right side of the above equation is available. The pdf $p(x_k|x_{k-1})$ is simply the pdf of the state at time k given a specific state at time (k-1). We obtain this pdf from the system equation $f_k(.)$ with known noise ω_k. This pdf can be calculated according to dynamical model of the system $f_k(.)$.

Next, we consider a *posteriori* conditional pdf of x_k . We can write this pdf as

$$p(x_k \mid y_{1:k}) = \frac{p(y_k \mid x_k) p(x_k \mid y_{1:k-1})}{p(y_k \mid y_{1:k-1})} . \tag{5.3}$$

All of the pdf's on the right side of the above equation are available. The pdf $p(y_k|x_k)$ is available from our knowledge of the measurement equation $h_k(.)$ and our knowledge of the pdf of the measurement noise v_k**.** The pdf $p(x_k|y_{1:k-1})$ is available from Eq. (5.2). Finally, the pdf p($y_k|y_{1:k-1}$) is obtained as follows:

$$p(y_k \mid y_{1:k-1}) = \int p(y_k \mid x_k) p(x_k \mid y_{1:k-1}) dx_k . \tag{5.4}$$

Both of the pdf's on the right side of the above equation are available. The pdf $p(y_k|x_k)$ is available from our knowledge of the measurement equation $h(.)$ and the pdf of v_k and $p(x_k|y_{1:k-1})$ is available from Eq. (5.2).

Summarizing the development of this section, the recursive equations of the Bayesian state estimation filter can be summarized as follows.

1. The system and measurement equations are given as follows:

$$x_{k+1} = f_k(x_k, \omega_k)$$

$$y_k = h_k(x_k, v_k)$$

   where $\{\omega_k\}$ and $\{v_k\}$ are independent white noise processes with known pdf's.

2. Assuming that the pdf of the initial state $p(x_0)$ is known, initialize the estimator as follows:

$$p(x_0|y_0) = p(x_0).$$

3. For k = 1, 2, …, perform the following.
   (a) The a *priori* pdf is obtained from Eq. (5.2).

$$p(x_k \mid y_{1:k-1}) = \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid y_{1:k-1}) dx_{k-1}$$

   (b) The a *posteriori* pdf is obtained from Eq. (5.3) and Eq. (5.4).

$$p(x_k \mid y_{1:k}) = \frac{p(y_k \mid x_k) p(x_k \mid y_{1:k-1})}{\int p(y_k \mid x_k) p(x_k \mid y_{1:k-1}) dx_k}.$$

Analytical solutions to these equations are available only for a few special cases. In particular, if $f(.)$ and $h(.)$ are linear, and x_0, $\{\omega_k\}$, and $\{v_k\}$ are additive, independent, and Gaussian, then the solution is the Kalman filter, otherwise it can be solved by particle filter.

5.2.2 Particle Filter for Object Tracking

The particle filter was invented to numerically implement the Bayesian estimator which recursively approximates the posterior distribution using a finite set of weighted samples or particles. It has been introduced by many researchers to solve the estimation problem when the system is nonlinear and non-Gaussian. The basic idea behind the particle filter is Monte Carlo simulation, in which the posterior density is approximated by a set of particles with associated weights. As a Bayesian estimator, particle filter has two main steps: prediction and update. Prediction is done by propagating the samples based on the system model. The update step is done by measuring the weight of each samples based on the observation model. The implementation of particle filter can be described as follows.

1. **Particle initialization**. Starting with a weighted set of samples at k-1 $\{x_{k-1}^i, \pi_{k-1}^i; i = 1:N\}$ approximately distributed according to $p(x_{k-1}|y_{1:k-1})$ as initial distribution $p(x_0)$, new samples set are generated from a suitable proposal distribution, which may depend on the previous state and the new measurements.

2. **Prediction step**. Using the probabilistic system transition model $p(x_k|x_{k-1})$, the particles are predict at time k. It is done by propagating each particle based on the transition or system model,

$$x_{k+1} = f_k(x_k, \omega_k) = p(x_k|x_{k-1}).$$

3. **Update step**. To maintain a consistent sample, the new importance weights are set to

$$\pi_k^i = \pi_{k-1}^i \frac{p\left(y_k|x_k^i\right)p\left(x_k^i\middle|x_{k-1}^i\right)}{q\left(x_k\middle|x_{1:k-1}, y_{1:k}\right)}. \tag{5.5}$$

   It is done by measuring the likelihood of each sample based on the observation model.

4. **Resample**. This step is performed to generate a new samples set according to their weight for the next iteration. The resample step will decrease the number of the samples with low weight and will increase the number of high weight samples. The new particles set is re-sampled using normalized weights $\bar{\pi}_k^i$ as probabilities. This sample set represents the posterior at time k, $p(x_k|y_{1:k})$.
5. Then, the expectations can be approximated as

$$E_p\left(x_k|y_{1:k}\right) \cong \sum_{i=1}^{N} \bar{\pi}_k^i x_k^i. \tag{5.6}$$

5.3 SYSTEM FLOW DIAGRAM

In this thesis, we proposed an algorithm based on particle filter in conjunction with a color feature. The motion of the central point of a bounding box is modeled using first-order dynamics model. In this article, the state of the object is defined as $\mathbf{s}_k = (\mathbf{x}_k, \dot{\mathbf{x}}_k, \mathbf{w}_k, \dot{\mathbf{w}}_k, A_k)$ where the components are position, speed, bounding-box size, bounding-box scale and pixel appearance, respectively. The observations $\boldsymbol{y}_k$ is given by input images I_k. We implement a particle filter in a color model-based framework to track the moving objects under outdoor environment. Firstly, the initialization of the samples done in the first frame is performed by drawing them randomly on entire scene or drawing them based on region where the object is expected to appear. Next, the samples are predicted based on a system/transitional model by propagating each sample based on transitional model. The samples update is performed based on the observation model. In this article, we use color distribution of the object as the observation model.

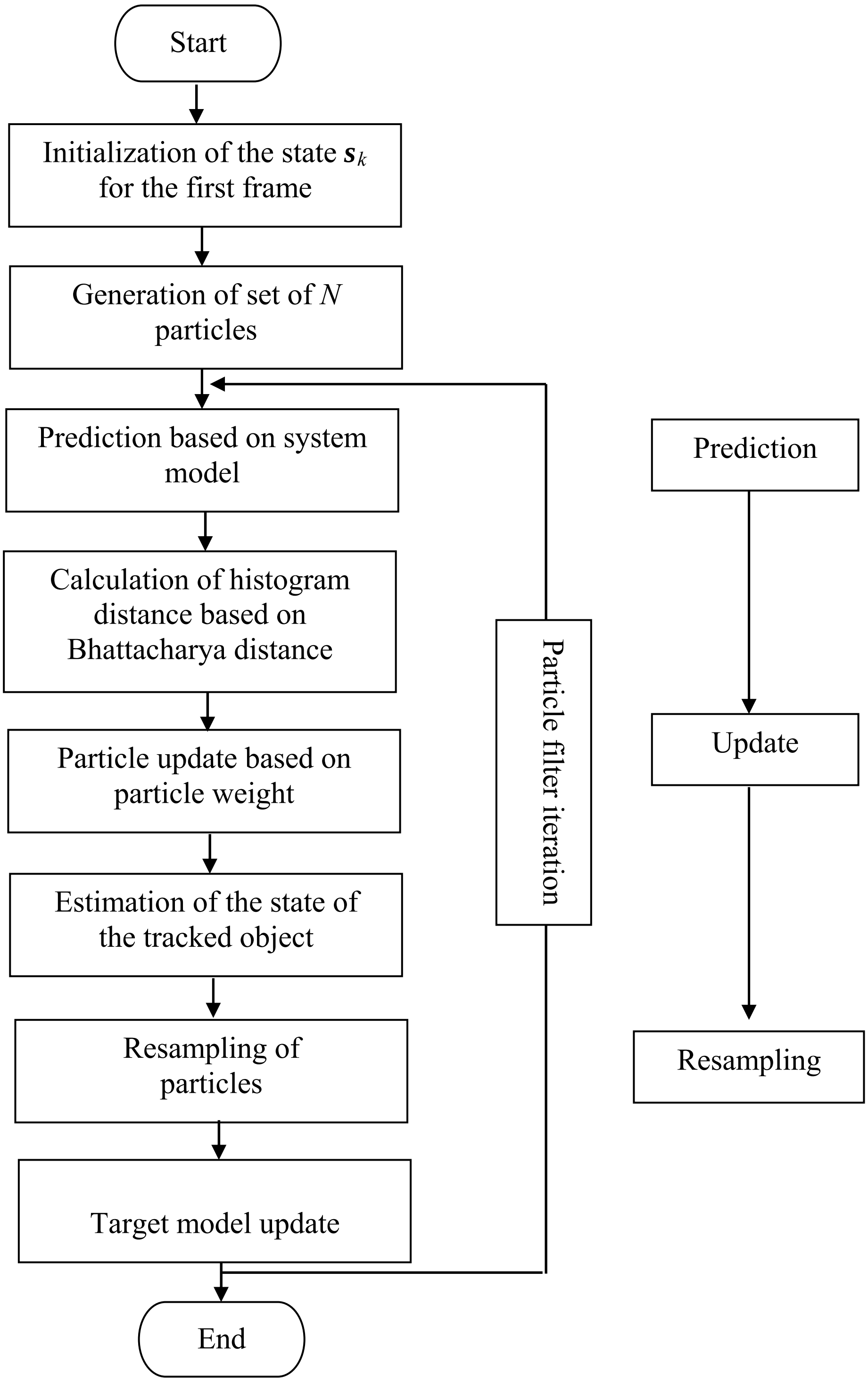


Figure 5.1 Entire flow diagram of color-based particle filter.

Then using the Bhattacharya distance, the similarity between the color distribution of the target and the samples can be measured. Based on the Bhattacharya distance, the weight of each sample can be measured. The target sate estimation is performed based on the sample's weight. The resampling is performed for the next sample iteration to generate a new samples set. During the resample step, samples with a high weight may be chosen several times leading to identical copies, while others with relatively low weights may be ignored and deleted. And finally, the target model update is performed adaptively based on the best match of the target model. The overall flow diagram is shown in Fig. 5.1.

5.4 INITIALIZATION PROCESS

The initialization of the particles is performed on two approaches: (i) samples are initialized by putting the samples randomly on entire scene and (ii) by putting the samples around the region where the target is expected to appear such as edge of the scene or edge of the occluded object.

On the first approach (Fig. 5.2), the samples are initialized randomly on entire scene (frame #1). In this approach, we utilize the variance of the samples position to be a parameter of object tracking. We determine that the object is begun to track when the variance of the samples is below then certain threshold. Fig. 5.3 shows the variance of the samples position for each frame index. From that figure, we can understand that the object has not been tracked yet before the variance of the sample less than the threshold, although the object appeared on the scene. We can see from Fig. 5.2 that the object is begun to be tracked in frame #10 when the variance of the sample position is less than threshold.

#1 #5 #10

Figure 5.2 Initialization based on the variance of the samples.

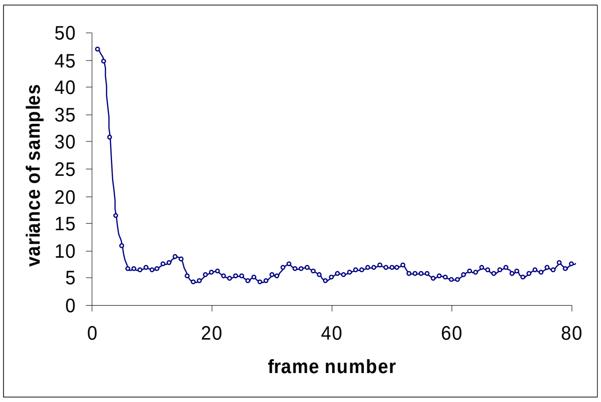


Figure 5.3 Variance of the samples.

#1 #9 #25

Figure 5.4 Initialization based on expected region.

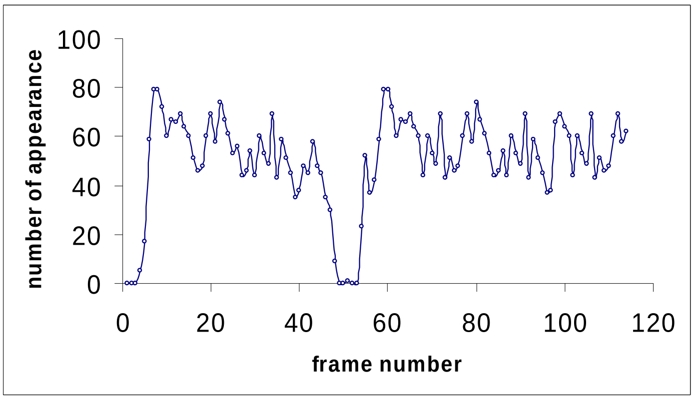


Figure 5.5 Appearance of the samples.

On the other hand, on the second approach, the object tracking is begun when the samples satisfy some special appearance conditions. In this condition, the samples are put around the region where the target is expected to appear such as edge of the scene or edge of the occluded object. In this approach, if the target appears, the Bhattacharyya distances (later we will discuss it) of samples around the object position should be remarkable smaller than the average of sample set. Therefore, the mean value

μ_b and the standard deviation σ_b of the Bhattacharyya distances of all initial samples are firstly calculated as,

$$\mu_b = \frac{1}{I}\sum_{i=1}^{I}\sqrt{1-\rho[p_{x_i},q]} \tag{5.7}$$

$$\sigma_b{}^2 = \frac{1}{I}\sum_{i=1}^{I}\left(\sqrt{1-\rho[p_{x_i},q]}-\mu\right)^2 \tag{5.8}$$

and then an appearance condition is defined as,

$$d = \sqrt{1-\rho[p_{x_i},q]} > \mu + 2\sigma \tag{5.9}$$

indicating a 95[%] confidence that a sample belongs to the object. Here, $\rho[p_{xi},q]$ is the Bhattacharya distance between the target model and the object. Then, an appearing and disappearing threshold T is defined to indicate the quantity of the samples fulfilling the appearance condition during initialization. More than T means the target being found and starting tracking, contrary situation means the tracker will enter the initialization mode. Fig. 5.4 shows the initialization approach (frame #1). The tracking is begun when the appearance condition is fulfilled (frame #9). The appearance of the samples is shown in Fig. 5.5. As presented in that figure, the object is detected and started to be tracked when the number of samples appearance more than threshold T.

5.5 SYSTEM MODEL

In this thesis, we consider the motion of the object as the discrete time 2-dimensional (2D) motion with constant velocity assumption. The state vector at a time step k is denoted by $\boldsymbol{s}_k$, including position, speed, size and bounding box scale of each sample. For this purpose, we model the system based on the following expression.

$$\begin{aligned}
\hat{\mathrm{x}}_k &= \mathrm{x}_{k-1} + \dot{\mathrm{x}}_{k-1}\Delta t + \xi_{\mathrm{x}}, \\
\hat{\dot{\mathrm{x}}}_k &= \dot{\mathrm{x}}_{k-1} + \xi_{\dot{\mathrm{x}}}, \\
\hat{\mathrm{w}}_k &= \mathrm{w}_{k-1} + \dot{\mathrm{w}}_{k-i}\Delta t + \xi_{\mathrm{w}}, \\
\hat{\dot{\mathrm{w}}}_k &= \dot{\mathrm{w}}_{k-1} + \xi_{\dot{\mathrm{w}}}.
\end{aligned} \tag{5.10}$$

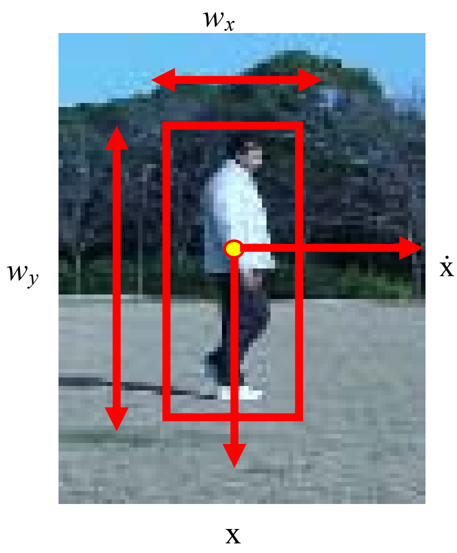


Figure 5.6 Definition of the object state.

Here, $\hat{x}_k, \hat{\dot{x}}_k, \dot{w}_k, \hat{\dot{w}}_k$ are the position, speed, bounding box and bounding box scale of the samples state, respectively. The random vectors $\xi_x, \xi_{\dot{x}}, \xi_w, \xi_{\dot{w}}$ are Gaussian noise providing the system with a diversity of hypotheses. Then, each component of the samples is predicted by propagating the samples according to this transition model. The definition of the object state is described in Fig. 5.6.

5.6 Observation Model

The observation model is used to measure the observation likelihood of the samples. Many observation models have been built for particle filtering tracking. One of them is a contour based appearance template [69]. The tracker based on a contour template gives an accurate description of the targets but performs poorly in clutter and is generally time-consuming. The initialization of the system is relatively difficult and tedious. In contrast, color-based trackers are faster and more robust, where the color histogram is typically used to model the targets to combat the partial occlusion, and non-rigidity [8, 9, 11].

In this article, the observation model is made based on color information of the target obtained by building the color histogram in the RGB color space. This section describes how the color features is modeled in a rectangular region R, where R can be a region surrounding the object to be tracked or region surrounding one of the hypothetical regions. A color histogram is commonly used for object tracking because they are robust to partial occlusion, rotation and scale invariant. They are also flexible in the types of object that they can be used to track, including rigid and non-rigid object.

The color distribution is expressed by an m-bins histogram, whose components are normalized so that its sum of all bins equals one. For a region R in an image, given a set of n samples in R, denoted by $\mathbf{X} = \{x_i,\ i = 1, 2, \ldots, n\} \in R$, the m-bins color histogram $H(R) = \{h_j\}$, $(j = 1, 2, \ldots, m)$ can be obtained by assigning each pixel x_i to a bin, by the following equation:

$$h_j = \frac{1}{n} \sum_{(x_i) \in \mathrm{X}} \delta_j [b(x_i)]. \tag{5.11}$$

Here $b(x_i)$ is the bin index where the color component at x_i falls into, and δ is the Kronecker delta function. We quantize each color channel (256 levels assumed) in RGB color space into 8 bins, and obtain a histogram with 8×8×8 = 512 bins. The quantization function is given by

$$b(x_i) = [R_{x,y} / 32] \times 8^2 + [G_{x,y} / 32] \times 8 + [B_{x,y} / 32] + 1 \tag{5.12}$$

where $R_{x,y}$, $G_{x,y}$, and $B_{x,y}$ are the RGB values of pixel (x, y).

To increase the reliability of the target model, smaller weight are assigned to the pixels that are further away from the center of the region by employing a weighting function

$$g(r) = \begin{cases} 1 - r^2 & r < 1 \\ 0 & otherwise \end{cases}. \tag{5.13}$$

Here, r is the distance from the center of the region.

Based on this weight, the color histogram $p_{\mathbf{y}} = \left\{ p_{\mathbf{y}}^{(u)} \right\} u = 1, \ldots, m$ at location $\boldsymbol{y}$ is calculated as

$$p_{\mathbf{y}}^{(u)} = f \sum_{j=1}^{I} g\left(\frac{\left\| \mathbf{y} - \mathbf{x}_j \right\|}{a} \right) \delta\left[h\left(\mathbf{x}_j\right) - u \right] \tag{5.14}$$

where, m is number of bins, I is the number of pixels in the region R, x_j is the position of pixels in the region R, δ is the Kronecker delta function, a is the normalization factor,

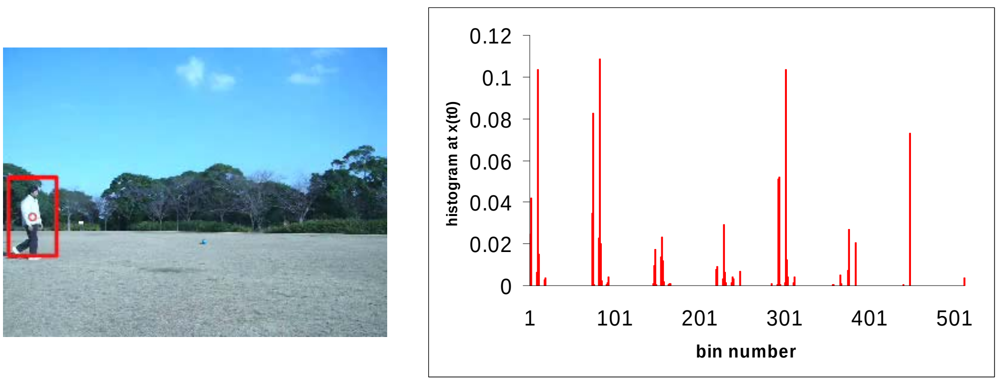
f is the scaling factor to ensures that $\sum_{u=1}^{m} p_{\mathbf{y}}^{(u)} = 1$, and $g(.)$ is weighting function, respectively. Fig. 5.7 shows an example of target histogram at time step k.

In subsequent frames, at every time k, there are N particles that represent N hypothetical states need to be evaluated. The observation likelihood model is used to assign a weight associated to a specific particle (new observation) depending on how similar the model histogram q and the histogram $p(x_t)$ of object in the region described by the i^{th} particle x_k^i are.

The similarity between two color histograms can be calculated using Bhattacharya distance $d = \sqrt{1-\rho[p,q]}$, where $\rho[p,q] = \sum_{u=1}^{m} \sqrt{p^{(u)} q^{(u)}}$. Similar histogram will have a small Bhattacharya distance which corresponds to high sample weight. Based on the Bhattacharya distance, the weight $\pi^{(i)}$ of the sample state $x^{(i)}$ is calculated as,

$$\begin{aligned}\pi^{(i)} &= \frac{1}{\sqrt{2\pi\sigma}} \exp\left(\frac{d^2}{2\sigma^2}\right) \\ &= \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{\left(1-\rho\left[p\left(\mathbf{x}^{(i)}\right), q\right]\right)}{2\sigma^2}\right)\end{aligned} \tag{5.15}$$

Where $p(x^{(i)})$ and q are the color histogram of the samples and target model, respectively.



(a) (b)

Figure 5.7 An example of color histogram distribution of a target model at time k.
(a) target object at time k, (b) histogram of the target

During the resample step, samples with a high weight may be chosen several times leading to identical copies, while others with relatively low weights may be ignored. The resample step is performed for the next sample iteration to generate a new samples set. During the resample step, samples with a high weight may be chosen several times leading to identical copies, while others with relatively low weights may be ignored and deleted.

The resample step can be done in several different ways. One straightforward way is as the following steps [100].

1. Generate a random number r that is uniformly distributed on [0, 1].
2. Calculate the normalized cumulative probability

$$c_k^{(0)} = 0,\ c_k^{(i)} = c_k^{(i-1)} + w_k^{(i)},\ c_k'^{(i)} = \frac{c_k^{(i)}}{c_k^{(N)}} \tag{5.16}$$

by binary search, find the smallest j for which $c_k' \geq r$ and set the new particle $x_k^i = x_k^j$

Each of the target state is estimated according their mean estimation as following equation,

$$\begin{aligned}
\mathrm{x}_k &= (1-\alpha_{\mathrm{x}})(\mathrm{x}_{k-1} + \dot{\mathrm{x}}_{k-1}\Delta t) + \alpha_{\mathrm{x}}\left(\textstyle\sum_{i=1}^{N} \overline{\pi}_k^i \hat{\mathrm{x}}_k^i\right), \\
\mathrm{w}_k &= (1-\alpha_{\mathrm{w}})\mathrm{w}_{k-1} + \alpha_{\mathrm{w}}\left(\textstyle\sum_{i=1}^{N} \overline{\pi}_k^i \hat{w}_k^i\right), \\
\dot{\mathrm{x}}_k &= (1-\alpha_{\dot{\mathrm{x}}})\dot{\mathrm{x}}_{k-1} + \alpha_{\dot{\mathrm{x}}}\left(\frac{\mathrm{x}_k - \mathrm{x}_{k-1}}{\Delta t}\right), \\
\dot{\mathrm{w}}_k &= (1-\alpha_{\dot{\mathrm{w}}})\dot{\mathrm{w}}_{k-1} + \alpha_{\dot{\mathrm{w}}}\left(\frac{\mathrm{w}_k - w_{k-1}}{\Delta t}\right).
\end{aligned} \tag{5.17}$$

Here, $\alpha_{\mathrm{x}}, \alpha_{\dot{\mathrm{x}}}, \alpha_{\mathrm{w}}, \alpha_{\dot{\mathrm{w}}} \in [0,1]$ denote the adaptation rates.

5.7 LIKELIHOOD AND MODEL UPDATE

5.7.1 Adaptive Likelihood

The observation likelihood function is very important for the tracking performance of particle filter. The reason is that the observation likelihood function determines the weights of the particles and determines how the particles are resampled. Resample step is used to decrease the number of low-weighted particles and increase the number of particles with high weights. The observation likelihood function is expressed in Eq. (5.15). In this case, the observation likelihood function is the only function that contributes to particle's weight. We can rewrite the equation as,

$$\pi^{(i)} \propto \exp\left(-\frac{d^2}{2\sigma^2}\right) \tag{5.18}$$

here, the σ parameter determines the steepness of the function, i.e. how fast the function will decrease when the distance is large (bad particles).

It is reported that the value of the parameter σ has a major influence on the properties of the likelihood [105]. Typically, the choice of this value is left as a design parameter to be determined experimentally. However, in this chapter, an adaptive scheme is proposed which aims to maximize the information available in the likelihood using the Bhattacharya distance *d.*

For this purpose, we define the minimum squared distance d^2_{min} as the minimum distance d^2 of the set of distance calculated for all particles. Rearranging the likelihood function yields,

$$log(\pi) \propto -\frac{d^2}{2\sigma^2} \tag{5.19}$$

from which we can obtain σ that gives maximum likelihood as,

$$\sigma \propto \frac{\sqrt{2}}{2}\sqrt{\frac{-d^2_{min}}{log(\pi)}}\,. \tag{5.20}$$

Where d_{min} is the distance of the most reliable particle (the particle with the minimum distance) and $log(\pi) = -1$.

#12 #16 #20

(a)

#12 #16 #20

(b)

Figure 5.8 Object tracking using adaptive and non adaptive likelihood.
(a) Using non-adaptive likelihood, (b) Using adaptive likelihood

Fig. 5.8 shows a comparison the object tracking method using the adaptive and non-adaptive likelihood model. Using non adaptive likelihood, the particles with different Bhattacharya distance will be assigned with equal weights. As a result, the likelihood function will not differentiate the difference between particle with small distance and particle with big distance. This condition may assign the bad particles with high weight (Fig. 5.8 (a)). On the other hand, the adaptive likelihood ensures the best particles are assigned with the highest weights and the worst ones are assigned with small weight (Fig. 5.8 (b)).

5.7.2 Target Model Update

The problem for object tracking with color occurs when the region around the object is cluttered, illumination is change and the object appearance is change. In this way, a color feature based tracking does not provide reliable performance because it fails to fully model the target. To overcome this problem, the adaptive target model is applied based on the best match of the color model. However, this is a sensitive task. The target models are only updated when two conditions hold: (i) the target is not occluded and (ii) the likelihood of the estimated target's state suggests that the estimate is sufficiently reliable. In this case, they are updated using an adaptive filter

$$q_k = (1-\alpha_q)q_{k-1} + \alpha_q p_{est} \tag{5.21}$$

where $\alpha_q \in [0,\ 1]$ is the learning rate that contribute to the updated histogram and p_{est} is histogram of the estimated state, respectively. In order to determine when the estimate is reliable, the likelihood of the current estimate is computed, π_{est}. The appearance is then updated when this value is higher than an indicator of the expected likelihood value and is calculated following an adaptive rule

$$\lambda_k = (1-\alpha_u)\lambda_{k-1} + \alpha_u \pi_{est} \tag{5.22}$$

here λ_k is expected likelihood, $\alpha_u \in [0,\ 1]$ is the learning rate and π_{est} is estimated likelihood, respectively. This value indicates that the object has to be well matched to the model histogram before the update is applied.

#210 #220 #230

(a)

#210 #220 #230

(b)

Figure 5.9 Object tracking with and without model update.

(a) Without target model update, (b) With target model update

#1 #30 #50

(a)

#1 #30 #50

(b)

Figure 5.10 Object tracking with model update shows the scale adaptation.
(a) Without target model update, (b) With target model update

Fig. 5.9 and Fig. 5.10 show the comparison of object tracking with and without target model update. As shown in Fig. 5.9, un-adaptive model fail to detect the object properly. However, the adaptive model can track the object successfully. It is because, using the adaptive model, the target model is always updated based on the best match of the color model. Moreover, as shown in Fig. 5.10, using adaptive model the scale of object appearance is getting adapted correctly compare to without using adaptive model.

5.8 EVALUATION METHOD

This session outlines the set of performance evaluation metrics in order to quantitatively analyze the performance of our object detection and tracking system. We implement a set of both frame-based and object-based metrics for the evaluation [106]. The ground truth information is represented in terms of the bounding box of object for each frame. Similarly, the results of object detection and tracking systems are in terms of the detected or tracked object's bounding box. At the time of evaluation, we employ different strategies to robustly test if the overlap between ground truth and system's results occurs. The simplest form of overlap is testing to see whether the system result's centroid lies inside the ground truth object's bounding box.

5.8.1 Frame-based Metrics

Starting with the first frame of the test sequence, frame-based metrics are computed for every frame in the sequence. From each frame in the video sequence, first a few true and false detection and tracking quantities are computed. Then, we evaluate the following characteristic,

- True Negative (*TN*)**:** Number of frames where both ground truth and system results agrees on the absence of any object.
- True Positive (*TP*)**:** Number of frames where both ground truth and system results agree on the presence of one or more objects, and the bounding box of at least one or more objects coincides among ground truth and tracker results.
- False Negative (*FN*): Number of frames where ground truth contains at least one object, while system either does not contain any object or none of the system's objects fall within the bounding box of any ground truth object.
- False Positive (*FP*): Number of frames where system results contain at least one object, while ground truth either does not contain any object or none of the ground truth's objects fall within the bounding box of any system object.

Total ground truth (*TG*) is the total number of frames for the ground truth objects and *TF* is the total number of frames in the video sequence. Once the above defined quantities are calculated for all frames in the test sequence, in the second step, the following metrics are computed:

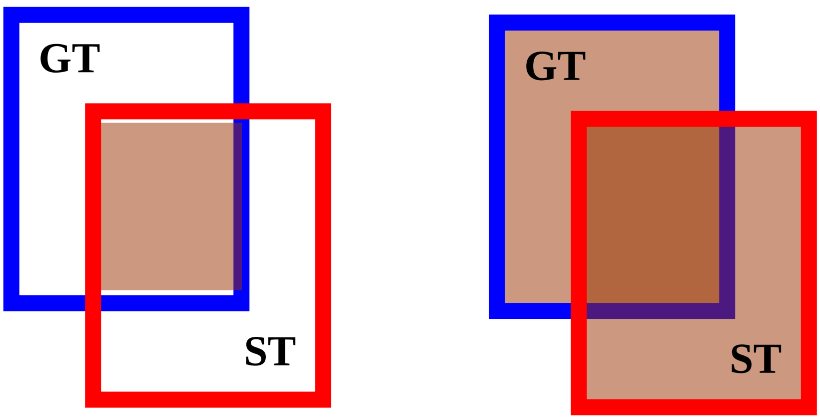
$$\begin{gathered}
\text{Tracker Detection Rate } (TRDR) = \frac{TP}{TG}, \\
\text{Detection Rate} = \frac{TP}{TP+FN}, \\
\text{Accuracy} = \frac{TP+TN}{TF}, \\
\text{False Alarm Rate } (FAR) = \frac{FP}{TP+FP}, \\
\text{False Positive Rate} (FPR) = \frac{FP}{FP+TN}.
\end{gathered} \tag{5.23}$$

5.8.2 Object-based Metrics

Object-based evaluation computes the metrics based on the complete trajectory and lifespan of the individual system and ground truth tracks. In this metric, the core problem is detecting whether the two bounding boxes, one from system track results and the other from ground truth, coincide or not. In this session, we compute the ratio of the intersection and union of the two boxes as illustrated in Fig. 5.11. If this ratio is within a fixed threshold between 0 and 1, the two objects are declared to have a match as shown in Eq. (5.24). Once the correspondence is established, we compute the true positive (*TP*), false positive (*FP*) and total ground truth (*TG*) as explained previously in the context of frame-based metrics. The metric such as *TRDR*, *FAR* and others are then computed as before.

.

$$\text{Degree of coincidence} = \frac{\text{GT} \cap \text{ST}}{\text{GT} \cup \text{ST}}. \tag{5.24}$$



(a) (b)

Figure 5.11 Intersection and union between ground truth and system.
(a) intersection region, (b) union region

5.9 COMPARATIVE RESULTS

To demonstrate the effect of number of particles and number of bins of the color histogram, we implement our color-based particle filter to track a moving object using various number of particles and number of bins of color histogram. Using the sequence of Fig. 5.12, we test our proposed color based particle filter with different number of particles and histogram bins and the results are the following.

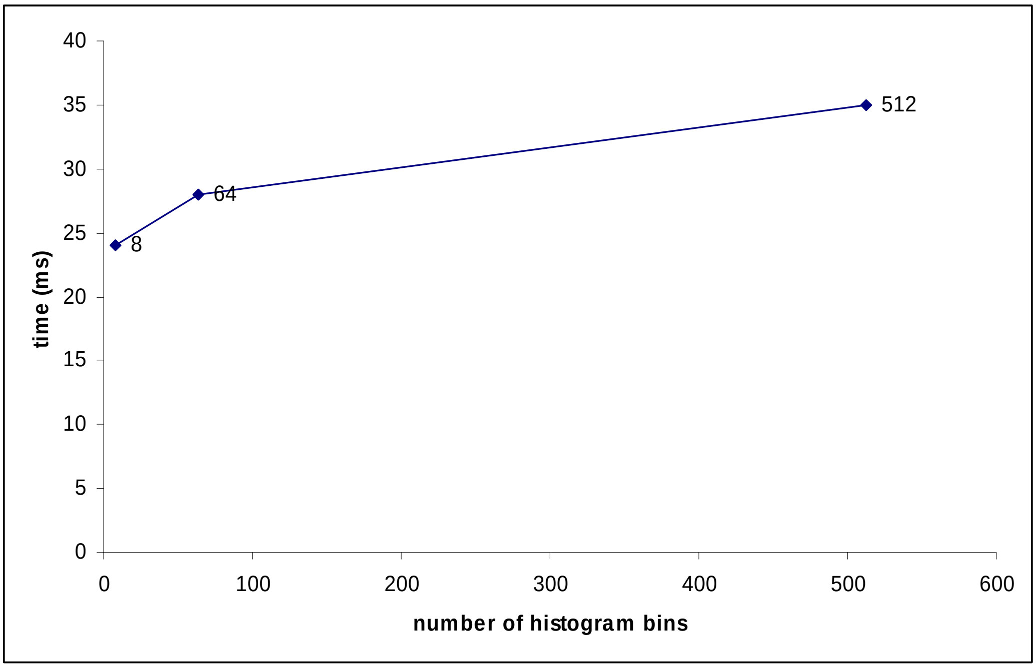
Fig. 5.13 shows the results regarding the processing time on different number of histogram bins and number of particles. From those figures, we can understand that the processing time increases when the number of histogram bins or number of particles increase.

Fig. 5.14 shows the results regarding the accuracy of the tracking process in term of *RMSE* (root mean square error) on different number of histogram bins and number of particles. From those figures, we can understand that the *RMSE* decreases (accuracy increase) when the number of histogram bins or number of particles increase.

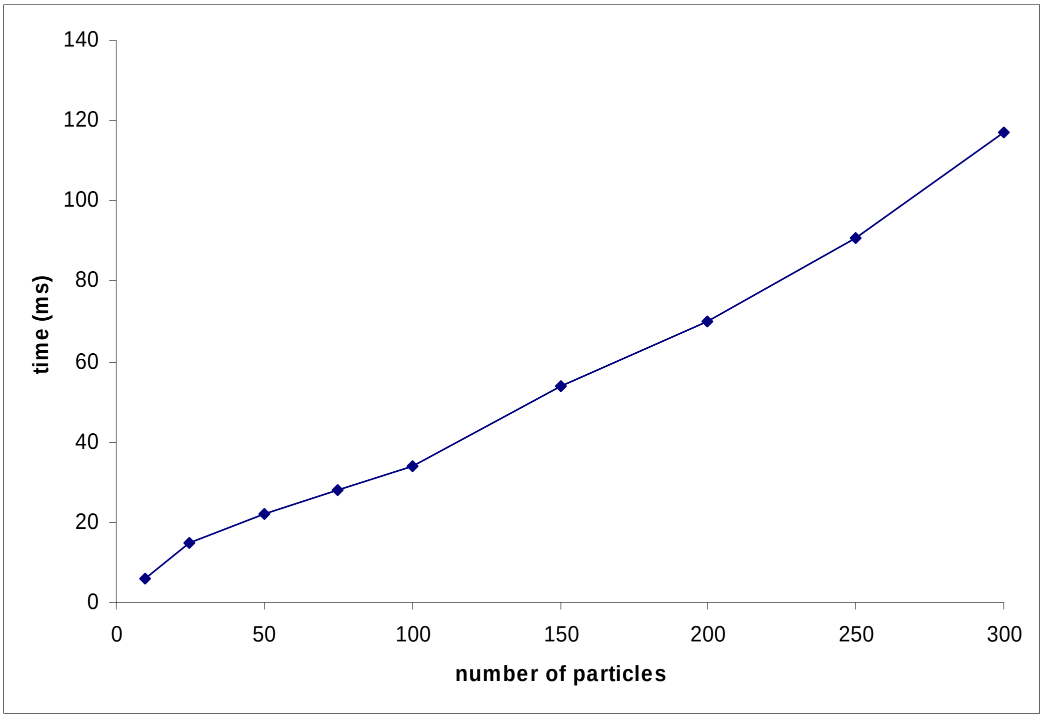
Fig. 5.15 shows the comparison between true and estimated position of the tracked object. From that figure, we obtained the *RMSE* (root mean square error) of the estimated position is about 2.06 for x position and 2.18 for y position.

#1 #30 #50

Figure 5.12 Object tracking sequence with 100 particle and 512 bins.



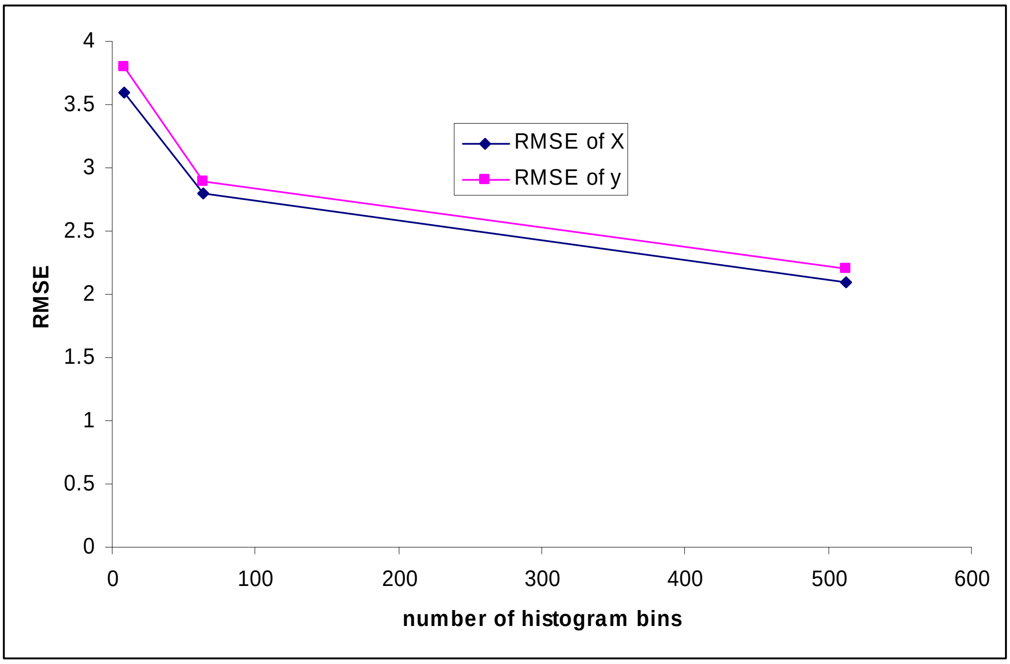
(a)



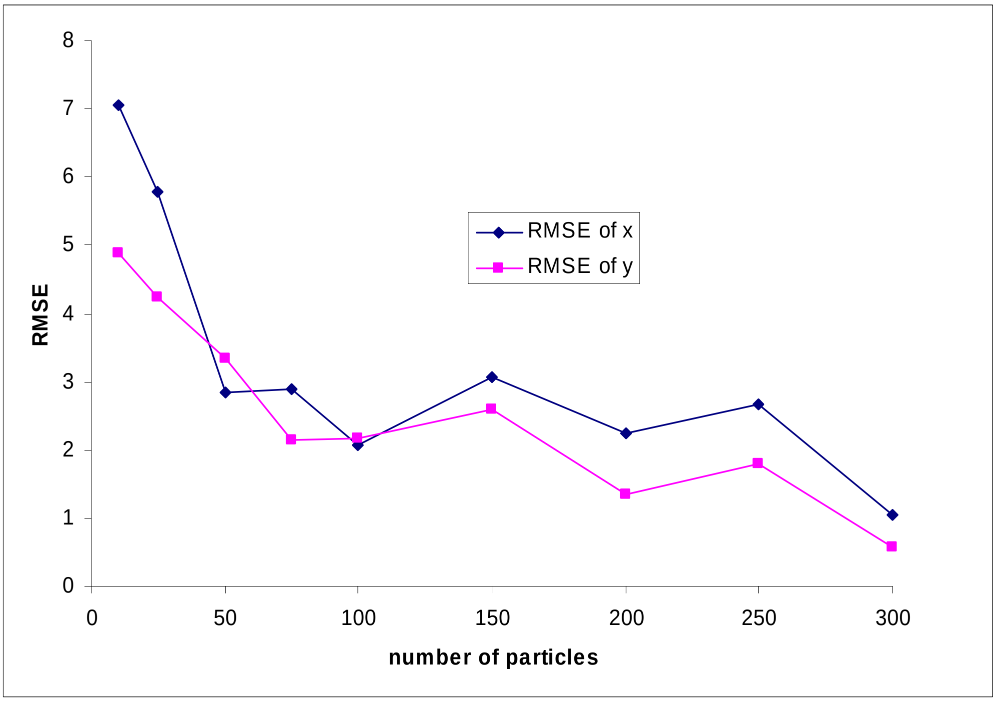
(b)

Figure 5.13 Processing time based on number of bins and particles.

(a) time vs bins number, (b) time vs number of particles



(a)



(b)

Figure 5.14 *RMSE* on *x* and *y* axes.

(a) *RMSE* vs bins number, (b) *RMSE* vs number of particles

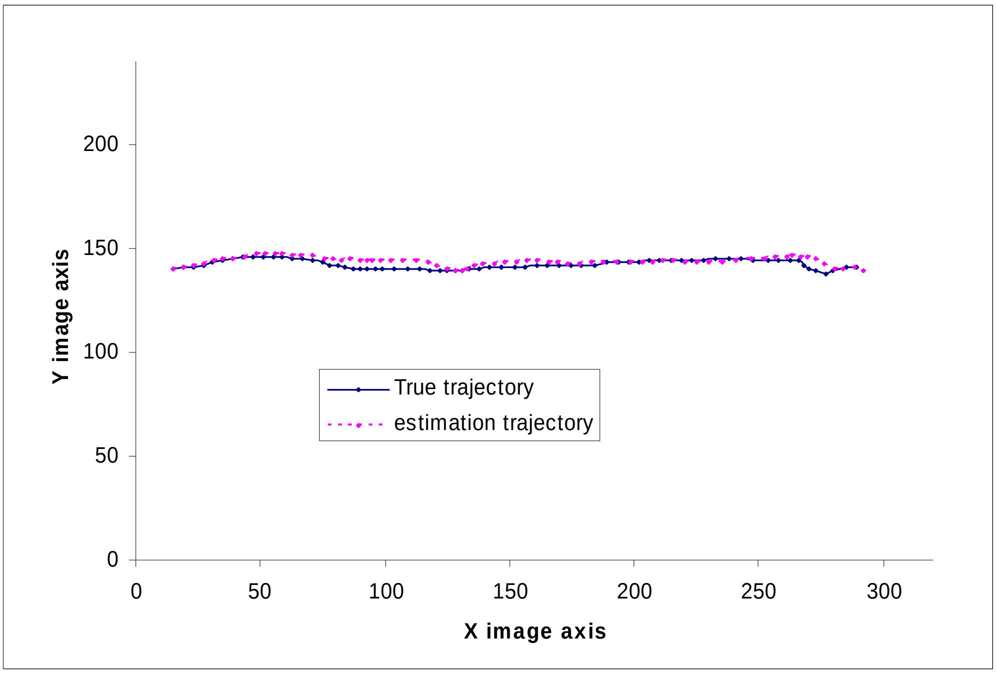


Figure 5.15 Estimated and true trajectory of the sequence of Fig. 5.13 on *x* and *y* axes.

5.10 EXPERIMENTAL RESULTS AND EVALUATION

In order to evaluate our proposed method, we have done the experiments using a video camera to track the objects in outdoor environment. The experiments are implemented on Pentium 4 with 2.54 [GHz] CPU and 512 [MB] RAM. The resolution of each frame is 320×240 [pixels] image. The color histogram is calculated in RGB space with $8 \times 8 \times 8$ [bins]. The experimental results are shown in Fig. 5.16 – Fig. 5.20. In each experimental result, the white dots represent the samples' distribution, the red dot represents the mean state of the samples' position and the red rectangle represents the bounding box of the tracked object used in calculation of the histogram distribution.

We did the experiments in three conditions. In the first condition, we tracked the moving object when the initial position of the object is unknown as shown in Fig. 5.16. In the second condition, we tracked the moving object when the initial position of the object is known as shown in Fig. 5.18 and Fig. 5.19. In the third condition, we tracked the moving object based on the appearance condition as shown in Fig. 5.20.

5.10.1 Object Tracking with Unknown Initial Position

In this experiment, we track the moving object when the initial position of the object is unknown. We set the initial samples to be uniform distribution in x_0~U(1,320) and y_0~U(1,240). The experimental results are shown in Fig. 5.16. As presented in the figure, initially, the samples are distributed uniformly around the scene (frame #1). The object moves from left-hand to right-hand side and begin to appear on frame #5, however, the object starts to be tracked on frame #10. In this experiment, the variance of the samples' position distribution is utilized to judge whether the object has been tracked. We consider the object has been tracked by the system when the variance is below 10 and it occurs on the frame #10. Fig. 5.17 shows the variance positions of the samples. We can understand from that figure that at the beginning the variance is very high (no object appears) but after some while it decreases below the threshold (object appeared). At that time, we determine that object is being tracked.

#1 #5 #10

#25 #50 #65

Figure 5.16 Object tracking with unknown initial position.

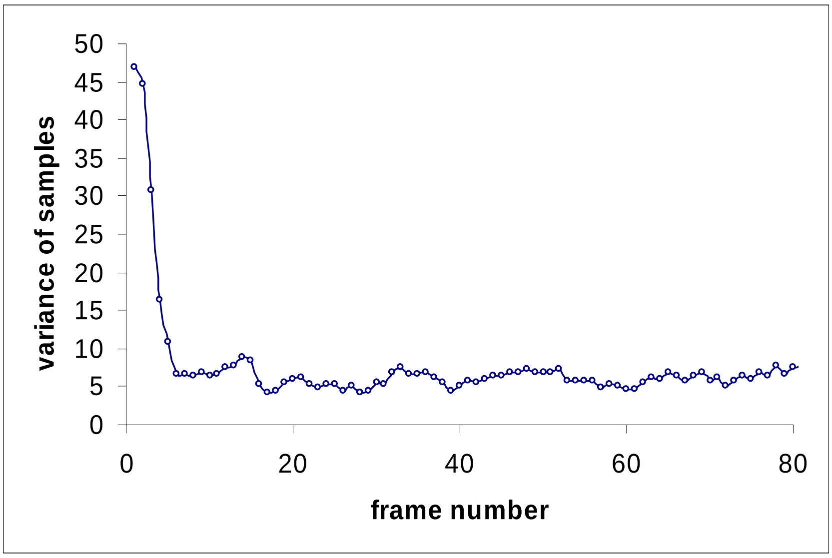


Figure 5.17 Variance of samples' position.

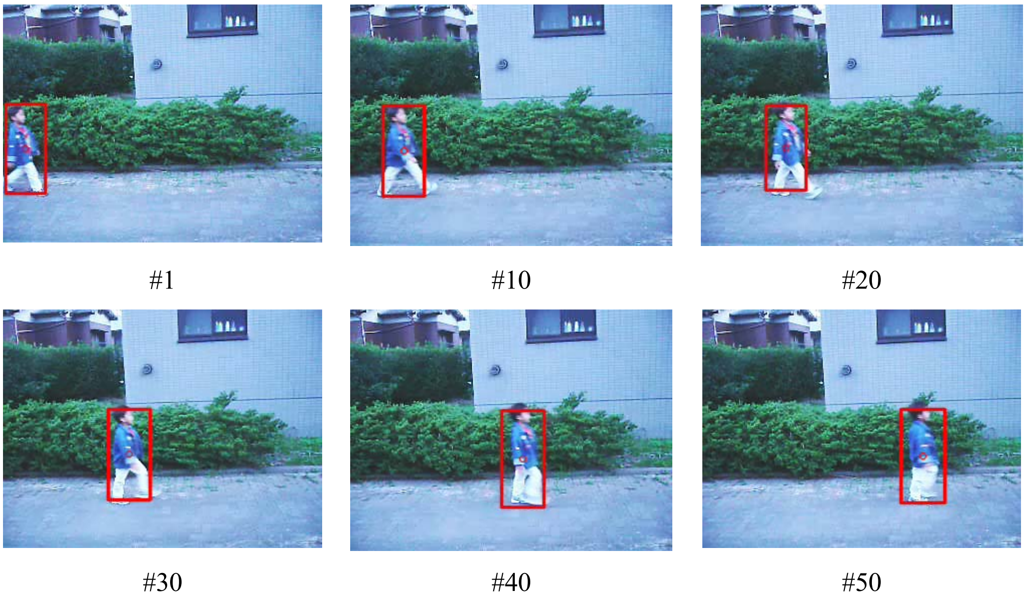


Figure 5.18 Object tracking with known initial position.

5.10.2 Object Tracking with Known Initial Position

In this experiment, the initial position of the moving object is known. We set the initial samples around the initial position of the object i.e. $x_0 \sim N(22,3)$ and $y_0 \sim N(145,3)$. The tracking result is presented in Fig. 5.18. Initially, the samples are distributed around initial position of the object. The object moves from left-hand to right-hand side and is tracked from the initial position (frame #1). We successfully tracked the object from the initial position until the end of the scene.

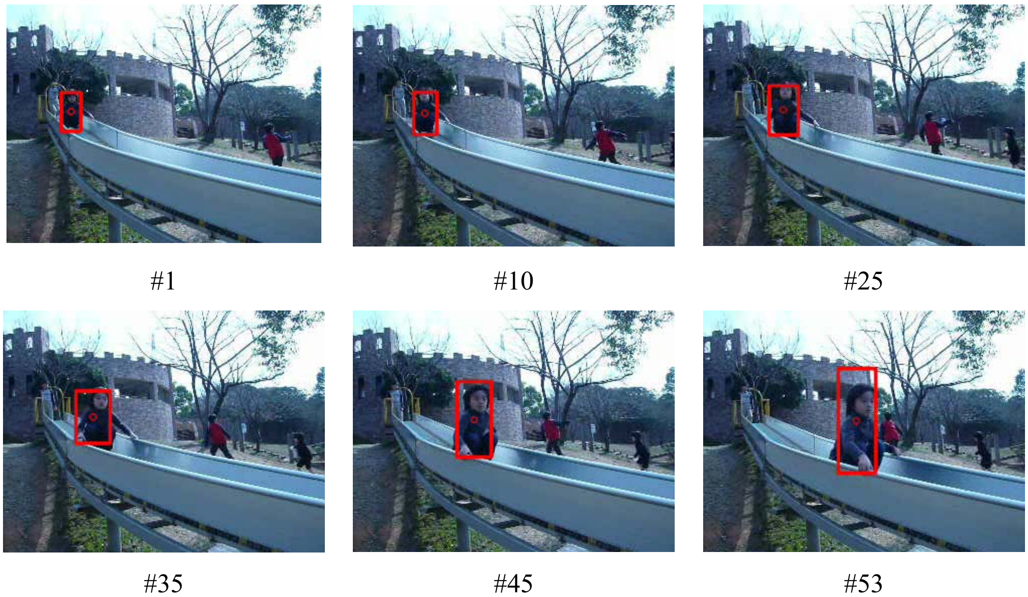


Figure 5.19 Another result of object tracking with known initial position.

Another result is shown in Fig. 5. 19. Here, we set the initial samples around the initial position of the object i.e. $x_0 \sim N(65,3)$ and $y_0 \sim N(106,3)$. The object is tracked from the initial position. We can observe that the scale of object bounding box is adaptively updated. It is because, in this experiment, the target model is always updated using the adaptive model based on the best match of the color model.

5.10.3 Object Tracking Based on Appearance condition

In this experiment, the samples are initialized around the expected region where the target is most likely to appear such as image borders and edge of the occluded object. As described in Fig. 5.20, the initial samples are placed at the position where the object is most likely to appear as shown in frame #1. The object is started to appear on frame #6, but as the appearance condition is not fulfilled, the object is still judged not to be tracked. However, when the appearance condition is fulfilled (frame #9), the object starts to be tracked.

The number of samples that fulfill the appearance condition is shown in Fig. 5.21. As presented on that figure, the object is detected and started to be tracked when the number of appearance more than 10. We can see from the figure, the number of appearance is started by zero at the initial position. After some while, the number of appearance increases due to appearance of the object. Using the threshold 10 samples appearance, the object starts to be tracked on frame #9 (Fig. 5.20).

Figure 5.20 Object tracking based on appearance condition.

The number of samples appearance can be used also to predict the object occlusion. As presented in Fig. 5.21, when occlusion is not detected, the number of samples appearance is always greater than certain threshold. During that time, the object is tracked based on normal particle filter steps. However, when the occlusion is detected, the number of samples appearance is dropped to zero between frames #45 to frame #50. During occlusion, the samples are just propagated based on the system model. Based on that condition, the object is tracked although the occlusion occurs. At frame #55, the number of samples appearance condition is higher again. It shows that the object is no more occluded by tree.

The tracking result is shown in Fig. 5.22. We successfully track the object in the presence of part and full occlusion. When occlusion occurs, the final mean state estimate is still calculated and the object is still tracked (frame #45 – frame #50). On that condition, the update and resample step are not performed due to no object color information can be represented the object. The samples are just propagated based on the system model and the target model is not updated until no more occlusion is detected. The object appears behind the tree (frame #55), due to the fulfilling of appearance condition, then the sample update and resample is performed again and the object continues to be tracked until the end of the frame.

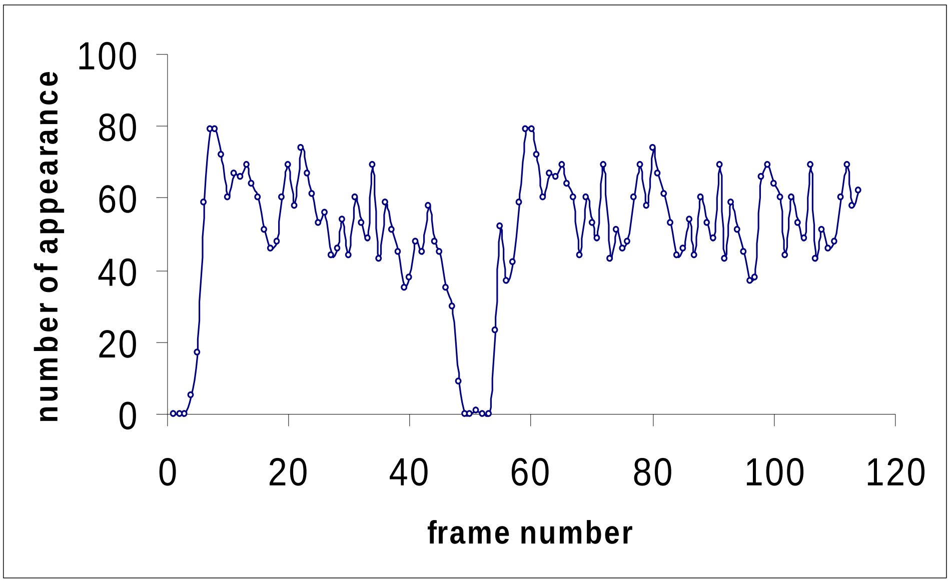
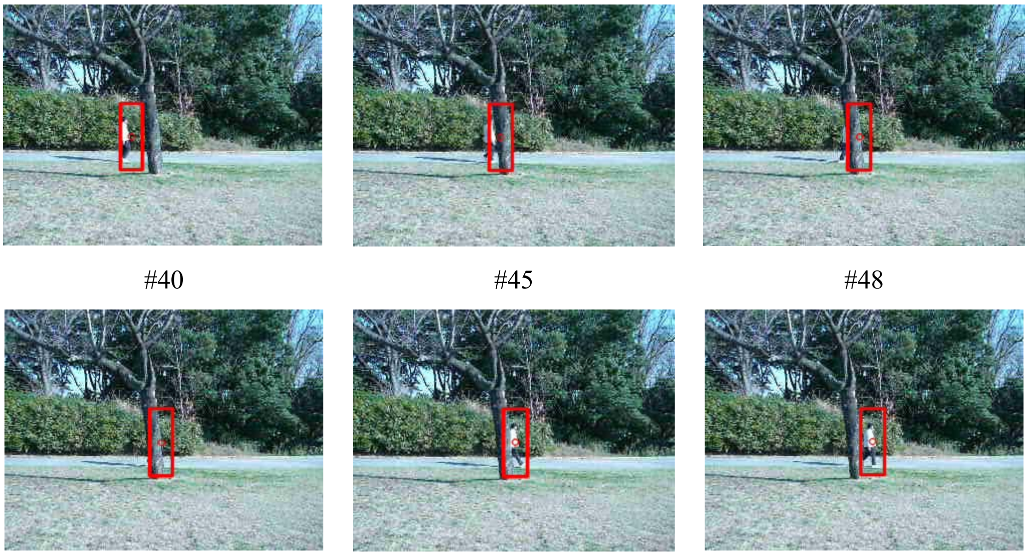


Figure 5.21 Number of samples fulfilling the appearance condition.



#40 #45 #48

#50 #53 #55

Figure 5.22 Single object tracking in the presence of occlusion.

5.10.4 Evaluation Result

The evaluation result is shown on the Table 5.1. From the table, we can notice that our proposed method successfully track the object with high detection rate and high accuracy. However, there is still miss-tracking on some frames of each experiment especially on the cases that the samples are initialized randomly or on the border of the expected region. Some of them are due to the object features used for the measurement of likelihood observation are still not represented the perfect object. This condition occurs because the object just enters the scene. Therefore only part of the object is measured. The other reason is the optimization done by particle filter algorithm did not go fast to adapt the change whether the object already appeared or not. The condition is illustrated on Fig. 5.22. The other miss-tracking occur when the particles appearance is still less than the threshold. On this condition, the particles are still not represented the predicted object state. It is because the limitation of placing the initial position of the particles. The condition is illustrated on Fig. 5.23. On those figure, the white dots represent particles state while red dot represents mean state of the particles or the predicted position of the tracked object.

#1 #5 #8

Figure 5.23 Miss-tracking on the unknown initial position tracking.

#1 #5 #7

Figure 5.24 Miss-tracking on tracking based on appearance condition.

Table 5. 1 Evaluation results.

Metric		Experiment 1	Experiment 2	Experiment 3	Experiment 4	Average [%]
Frame-based	TRDR [%]	91.57	100	100	95.87	96.86
	FAR [%]	0	0	0	0	0
	Detection Rate [%]	88.37	100	100	95.87	96.06
	Accuracy [%]	92.86	100	100	96	97.21
	FPR [%]	0	0	0	0	0
Object-based	TRDR [%]	89.13	100	100	84.23	93.34
	FAR [%]	0	0	0	0	0
	Detection Rate [%]	87.34	100	100	84.73	93.01
	Accuracy [%]	85.71	100	100	85.00	92.68
	FPR [%]	0	0	0	0	0

Experiment 1: object tracking with unknown initial position as shown in Fig. 5.16

Experiment 2: object tracking with known initial position as shown in Fig. 5.18

Experiment 3: object tracking with update scale as shown in Fig. 5.19

Experiment 4: object tracking based on appearance condition as shown in Fig. 5.20

5.11 CONCLUSIONS

This chapter presented a method to track the moving object employing particle filter in conjunction with color information based on known and unknown initial position of the object and object appearance. The robust color likelihood is used to evaluate samples properly associated to the target which present high appearance variability. We rely on Bhattacharya coefficient between target and samples histogram to perform this task. Model updating is carried to update the object in the presence of appearance change. The performance of the tracking algorithm was evaluated by the experiments. We analyzed the effect of the number of particles and number of histogram bins to the processing time and tracking accuracy. We obtained that the processing time is related to the number of particles and number of histogram although the tracking accuracy increase also. The experimental results show the algorithm can successfully track the single moving object based on known and unknown initial position and object appearance. Our system can track successfully the object in 4 conditions and can achieve the high detection rates more than 90 [%] in average.

CHAPTER VI

MULTIPLE OBJECTS TRACKING BASED ON PARTICLE FILTER ALGORTIHM

CHAPTER VI
MULTIPLE OBJECTS TRACKING BASED ON PARTICLE FILTER ALGORITHM

6.1 INTRODUCTION

In chapter 5, we presented particle filter algorithm to track the single moving object. We successfully tracked the single moving object in different conditions. In this chapter, we expand the algorithm by performing multiple objects tracking system in the presence of object occlusion. The multiple objects tracking has more challenging problem than the single object tracking due to the presence of objects occlusion, objects appearing and disappearing and so on. The object occlusions can cause a failure on the tracking process. They may cause inaccurate in estimation and updating of the state of the tracked object since the likelihood of the occluded object would be meaningless. The resampling phase would propagate the wrong random samples and quickly cause the loosing of the object.

To overcome the problem, various researches were proposed in the past. Among them, methods that are based on Bayesian filter framework with sequential Monte Carlo (particle filter) implementation are attracting substantial interest. Some of these methods are implemented on the single object state space [109, 110]. These methods approximated the mixture filtering distribution by particle filter to maintain multi-modality. However, a common limitation of these methods is that if objects are close to each other and particles from one specified object have very high weight, the particles representing the remaining objects are often suppressed. In addition, there are methods using a joint state space for object tracking [9, 107, 108]. In [9], the state vector denoting all the existing targets is augmented by a discrete random variable which represents the number of existing objects in a video sequence. They proposed the appearance and disappearance object based the on transitional probability matrix. The particle filter developed in [107] has multiple models for the object motion and comprises an additional discrete state component, denoting which of the motion models is active. In [108], the Bayesian Multiple-Blob Tracker (BraMBLe) presented a multiple persons tracking system based on statistical appearance models. The multiple blob tracking is managed by incorporating the number of objects present in the state vector and state vector is augmented as in [9] when a new object enters the scene. However, sampling particles from a joint state space can become inefficient as the dimension of the space increases. Although there are some attempts to reduce the number of particles such as [111], it is still computational demanding.

In this chapter, we propose a method for tracking multiple objects from video sequence using particle filter based on color measurements. We propose to handle object occlusion based on particle filter algorithm by redefining the resample and target update of the samples. The occlusion is predicted based on dynamical model and likelihood measurement. We divide the occlusion into two parts: partial and full occlusion. In partial occlusion, we propose to measure the likelihood in the region of each un-occluded object based on the predicted state. In this condition, the samples are still resampled and the target model is updated. In the case of full occlusion, likelihood of occluded object is not measured because it would be meaningless. In this condition, there is no resample and model updating; the samples are just propagated based on the system model. However, the likelihood of the samples of un-occluded object is still measured.

The entire procedure of our proposed method is described on Fig. 6.1. After objects are detected by the system, the system will start to track the objects based on likelihood measurement of each object. As long as the objects are not occluded, all of the particle filter steps; prediction, update and resample, are performed on each object. However, when the object occlusion occurs, measurement of likelihood in update step and resampling step are only performed on the un-occluded object. The samples of the occluded object are just propagated based on the system model. Then, the objects are tracked based on each condition.

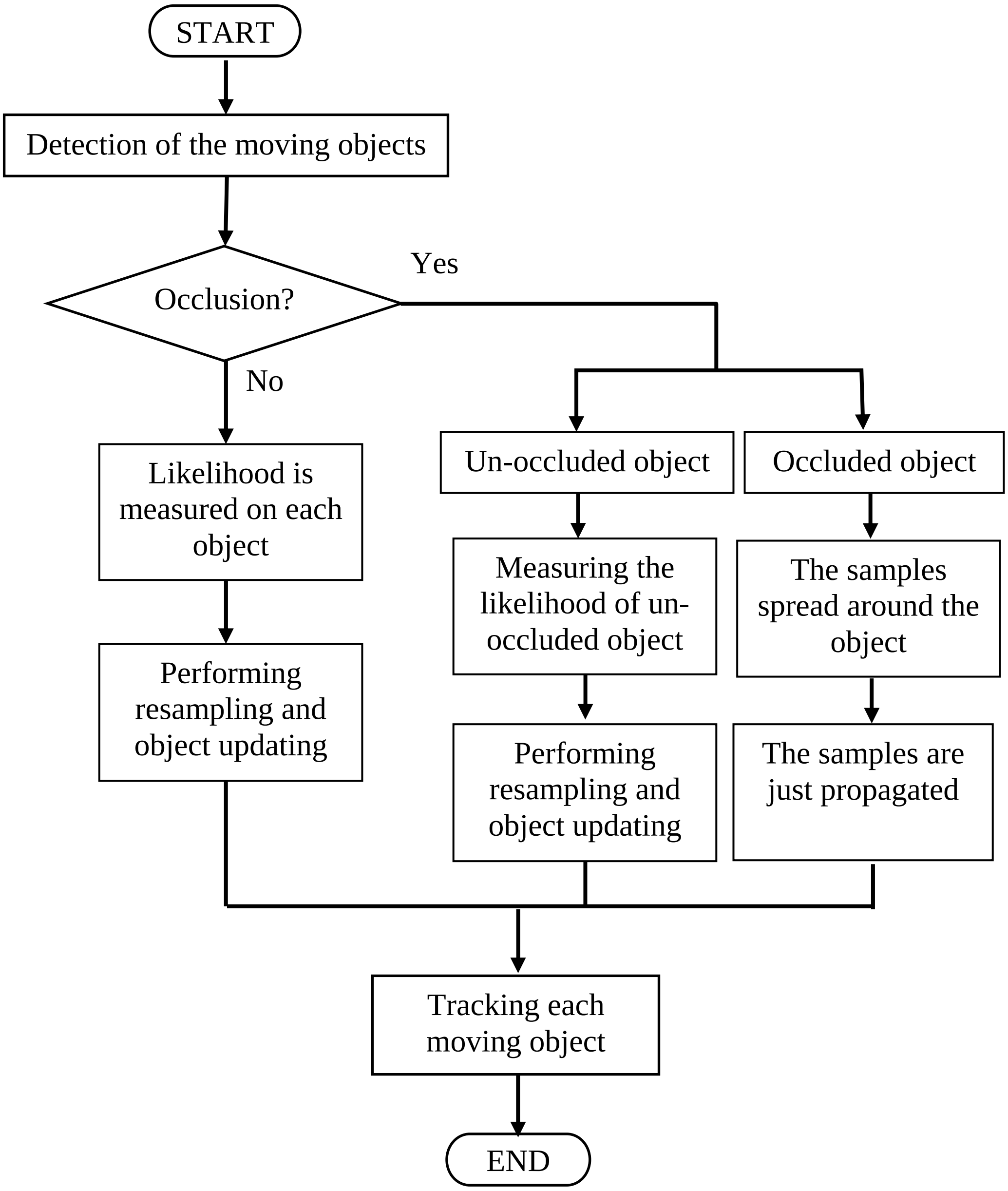


Figure 6.1. Entire procedure of occlusion handling.

6.2 OCCLUSION HANDLING ON MULTIPLE OBJECTS TRACKING

In multiple objects tracking, the problem of occlusions can cause a failure in the tracking process. They may cause inaccurate in estimation and updating of the position and size of the tracked object since the likelihood of the occluded target would be meaningless. The resampling phase would propagate the wrong random samples and quickly cause the loosing of the object. Therefore, a proper handling of occlusions is crucial.

In this chapter, the objects occlusion are predicted according to the dynamic models using the predicted distance between the objects such as,

$$(x^{i}_{m,k} - x^{i}_{n,k})^2 + (y^{i}_{m,k} - y^{i}_{n,k})^2 < R \qquad (6.1)$$

here x_m, y_m, x_n and y_n are the samples position of each object and R is a threshold, respectively. Occlusion also can be predicted using likelihood measurement by comparing it with the recent historical values.

The object occlusion is handled based on the two categories of occlusion: full occlusion and partial occlusion. When full occlusion is detected as shown in Fig 6.2, the occluded object (B) status turns into occluded object. This status involves several changes in the normal development of the process such as the adaptation rates (explained in Chapter 5) $\alpha_{\mathrm{x}}, \alpha_{\dot{\mathrm{x}}}, \alpha_{\mathrm{w}}, \alpha_{\dot{\mathrm{w}}}$ are set to zero and the object estimated speed is kept constant and the position is updated only according to its speed. In addition, no size or appearance adaptation is performed. Finally, those samples belonging to the occluded object are not resampled according to their weights but they are just propagated. As a result, samples spread around the object, because of the uncertainty predictions terms. The other object (A) samples are normally resampled but they cannot be assigned to the occluded target. When the occlusion is no longer predicted or sample likelihood exceeds the value of previous likelihood of the occlusion object, the object status turns into not occluded, which immediately implies the samples to be resampled again. In addition, position and speed are again updated.

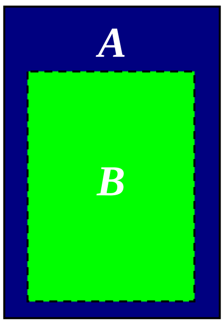
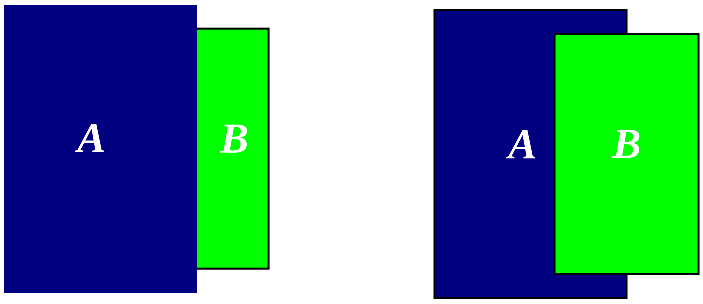


Figure 6.2. Object *A* fully occludes *B*



(a) (b)

Figure 6.3. Partial occlusion between object A and B

(a) Object A occludes B (L_{AB}), (b) Object B occludes A (L_{BA})

On the other hand, when partial oclusion occurs as shown in Fig. 6.3, the samples belonging to un-occluded area are still resampled according to object likelihood. We form the likelihood by marginalizing out all the possible dept ordering for each non-empty region intersection. For instance, for two occluded object A and B the likelihood is defined by averaging the likelihood as following,

$$L = 0.5(L_{AB} + L_{BA}) \tag{6.2}$$

where

$$\begin{aligned} L_{AB} &= L(Object_A\ region_A + Object_B\ region_{B-A}) \\ L_{BA} &= L(Object_B\ region_B + Object_A\ region_{A-B}) \end{aligned} \tag{6.3}$$

here, L_{AB} is likelihood under hypothesis object A occludes B and L_{BA} is likelihood under hypothesis object B occludes A. $Object_A\ region_A$ is likelihood calculation on the region of object A and $object_B\ region_{B\text{-}A}$ is likelihood calculation on the un-occluded region of object B as shown in Fig. 6.3 (a), respectively. These likelihoods are calculated under hypothesis object A occludes object B. $Object_B\ region_B$ is likelihood calculation on the region of object B and $object_A\ region_{A\text{-}B}$ is likelihood calculation on the un-occluded region of object A as shown in Fig. 6.3 (b), respectively. These likelihoods are calculated under hypothesis object B occludes object A.

6.3 EXPERIMENTAL RESULTS

In order to evaluate our proposed method, we have done the experiments to track the multiple objects in outdoor environment. The experiments are implemented on Pentium 4 with 2.53 [GHz] CPU and 512 MB [RAM]. The color histogram is calculated in RGB space with 8×8×8 bins. We implement our proposed algorithm to handle occlusion occurring in multi objects tracking. We predict the occlusion based on the estimated distance between the objects and likelihood measurement. We determine the object occlusion based on Eq. (6.1). When the distance between the object is less than the threshold, the system recognizes the occlusion between objects occurs. We did the experiments in three conditions.

On the first experiment, the objective is to track two moving objects when they move from different side of the scene. The resolution of the image is 320×240 [pixels]. The first object comes from the left side of the scene and the second object comes from the right side of the scene. Each object is associated with individual template and moves independently. Here, we successfully track each object by our proposed method. The tracking performance is shown in Fig. 6.4. Firstly, the first object appears on the left side of the scene and the second object appears afterward on the right side (frame #10). After several frames, the first object occludes the second object in the middle of the scene (frame #20 – frame #35). The object is tracked successfully although partial occlusion (frame #22 and frame #35) and complete occlusions (frame #26 – frame #30) occurred. The system successfully recovers the object from occlusion (frame #40). After the occlusion is no longer occurred, each object samples is normally resampled and the target appearance is update again.

The object occlusion is correctly detected using likelihood measurement shown in Fig. 6.5 (a) and Fig. 6.5 (b) and the estimated distance between object shown in Fig. 6.6. We determined the distance threshold for occlusion is 40. With the assumption that objects move on horizontal direction, the distance threshold is obtained from the minimum wide of the objects. From Fig. 6.6, we can understand that the estimated distance of the object is less than threshold between frame #20 and frame #35. We determined that the occlusion occurred between those frames. Moreover, the likelihood measurement also showed the similar result (Fig. 6.5 (a) and Fig. 6.5 (b). Based on this figure, we can see that the occluded object has lower likelihood compared with the likelihood of the recent historical value (previous frames). It occurred between frame #20 and frame#35. These results are similar with the prediction based on estimated distance.

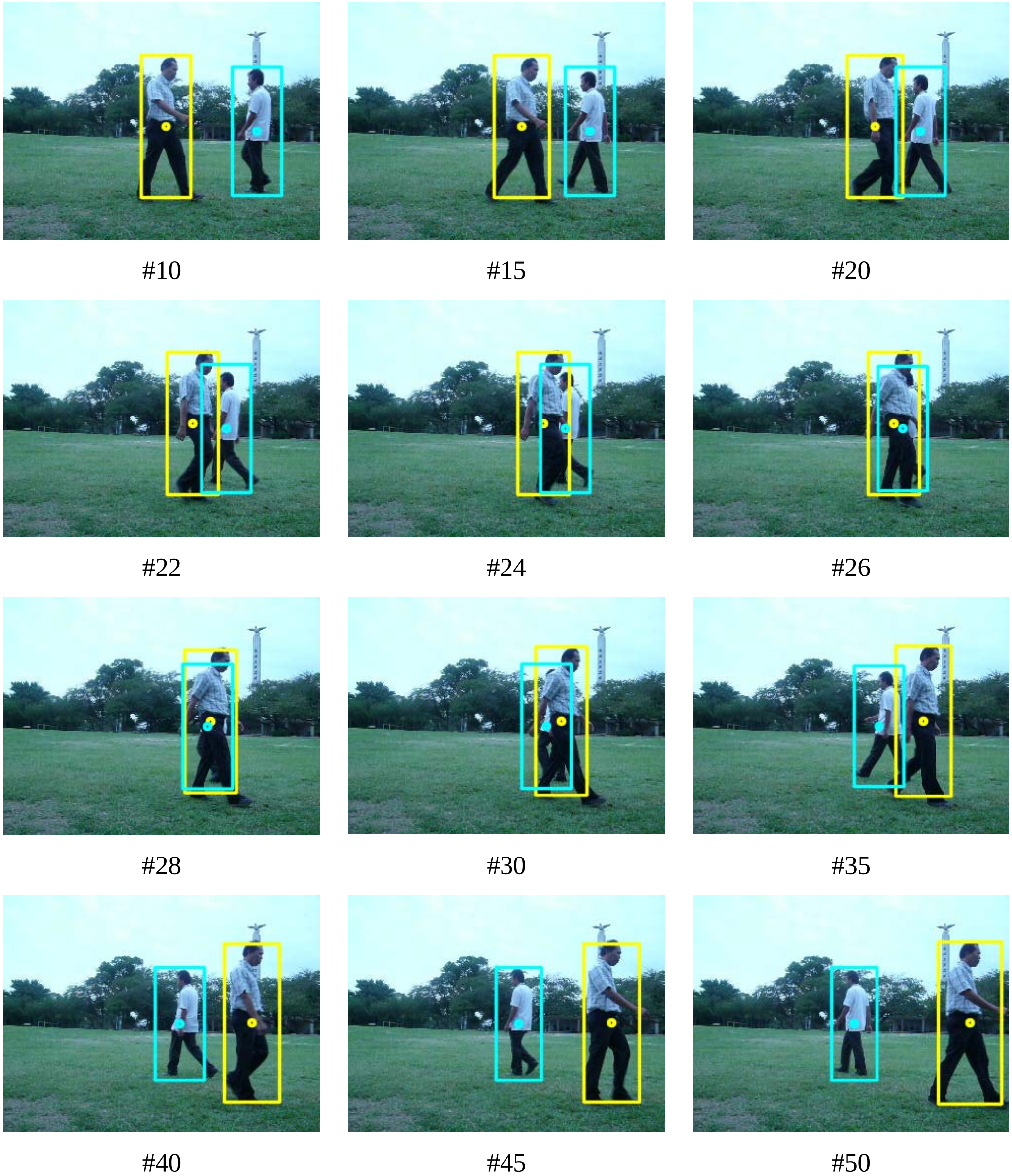
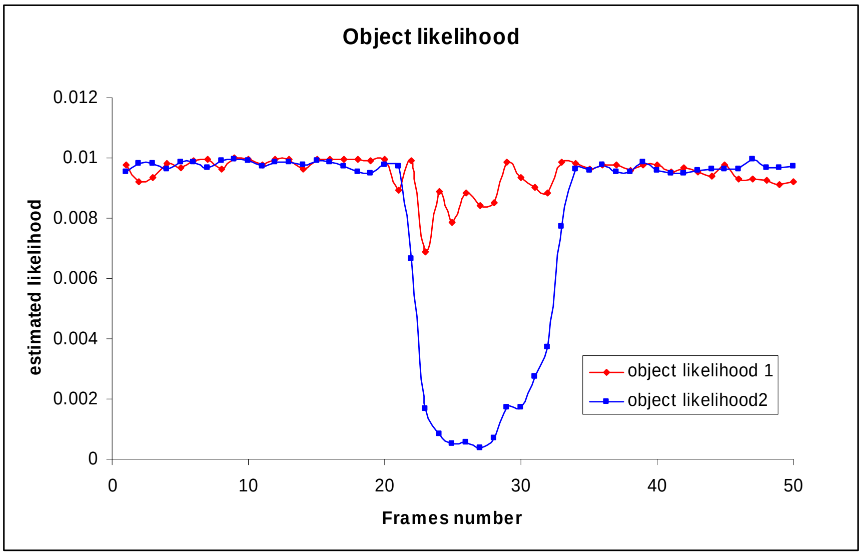
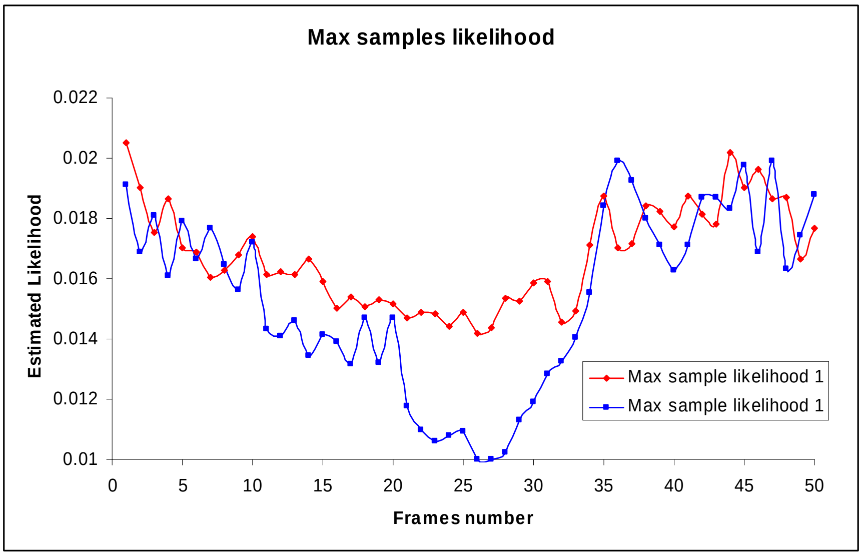


Figure 6.4. Multiple objects tracking in the presence of occlusion when they move in the opposite direction.



(a)



(b)

Figure 6.5. Object likelihood measurement.

(a) estimated object likelihood, (b) maximum object likelihood

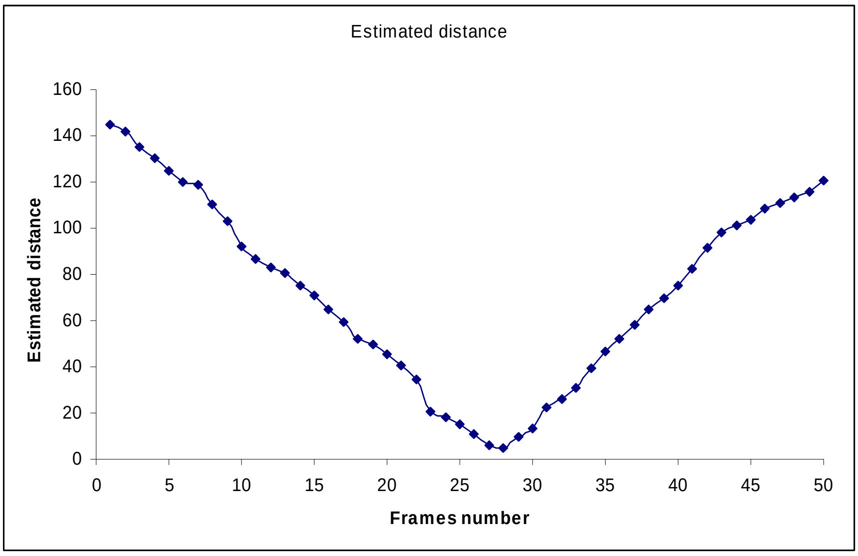


Figure 6.6. Estimated distance between predicted object positions.

On the second experiment, we track the moving objects when they move in the same direction and occlude in the middle of the scene. The resolution of the image is 320×240 [pixels]. The first object comes from the left side of the scene and the second object comes from the same side of the scene. The second object moves faster than the first one. Each object is associated with individual template and moves independently. Here, we successfully track each object by our proposed method. The tracking performance is shown in Fig. 6.7. The partial occlusion is detected on frame #40. At that time, the samples within the occluded region are not resampled but they are just propagated, while the samples within un-occluded region are still resampled. The complete occlusion is occurred on frame #70. The second complete occlusion then occurs on frame #140 as the second object change the motion direction to the left side of scene (frame #115). When full occlusion is detected, the samples of occluded object are not resampled. They are just propagated based on the dynamical model and spread around the object. The system successfully recovers the second object from occlusion (frame #161). After the occlusion is no longer detected, the samples of each object are normally resampled and the object appearance is update again.

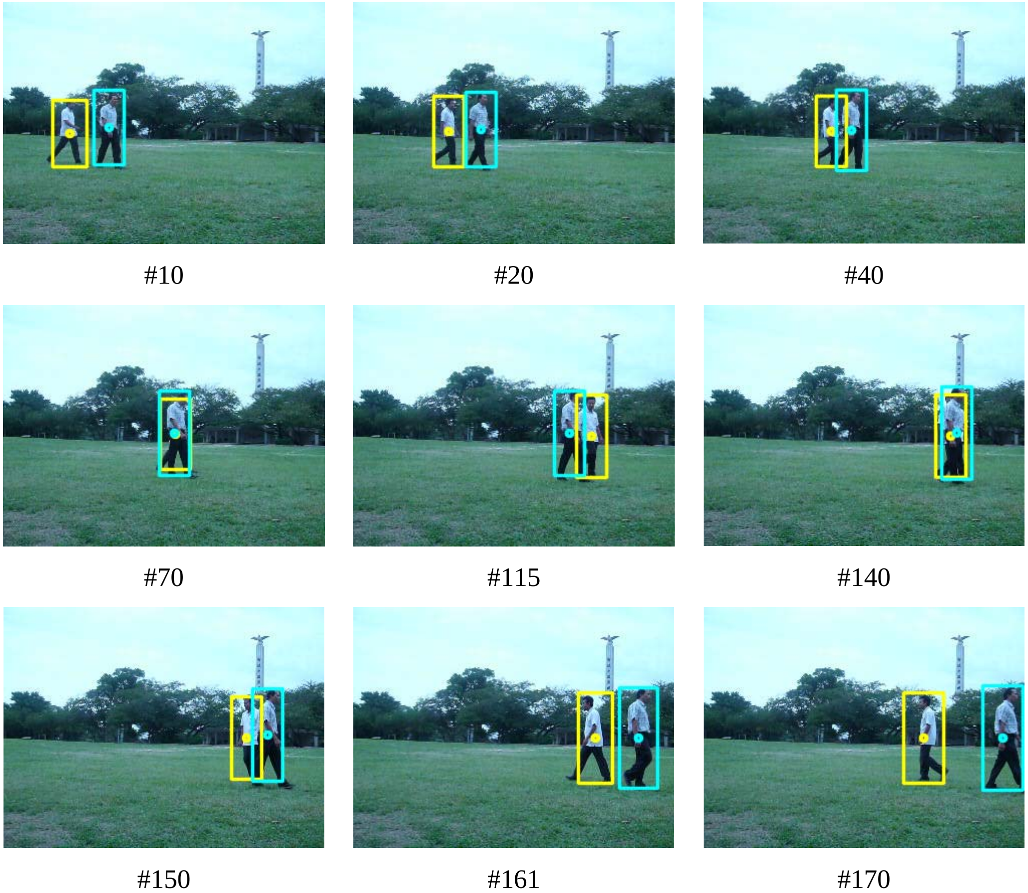


Figure 6.7. Multiple objects tracking in the presence of occlusion when they move in the same direction.

On the third experiment, the objective is to track the moving objects of the players of the team in red and white shirt on the football sequence. The video sequence is downloaded from http://www.youtube.com/watch?v=zZ5tt6_guR4. The resolution of the image is 480×360 [pixels]. Each object is associated with individual template and moves independently. Here, we successfully track each object by our proposed method. Fig. 6.8 shows the tracking performance of this video sequence. The tracking result is described in different colored rectangular.

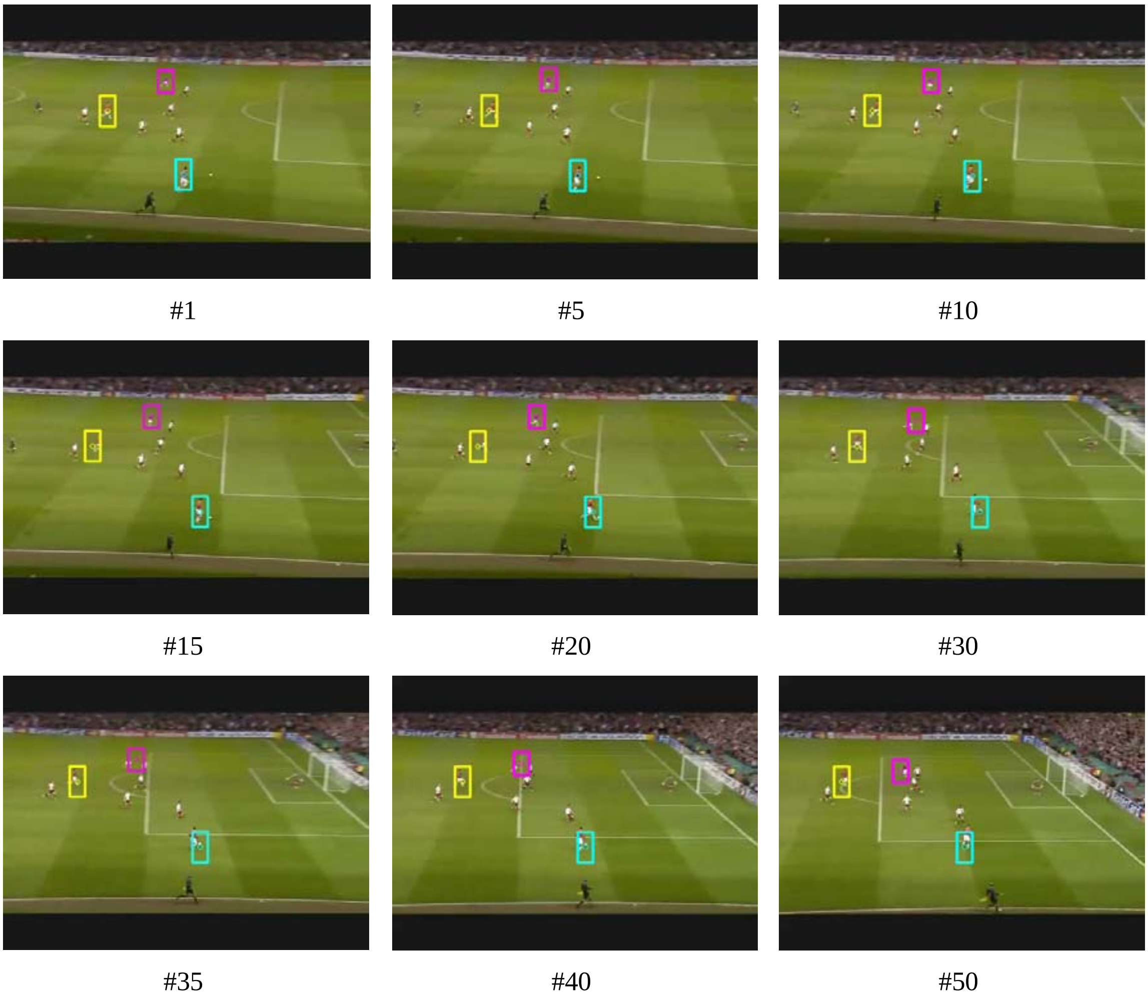


Figure 6.8. Multiple objects tracking on football sequence.

The processing time of the proposed method is related to the number of the objects to be tracked. It is because each object has individual template and move independently. So the particle filter should estimate each object based on their template and their motion model. Most of the time in our method is spent on building the color histogram whose complexity is proportional to the number of particles and size of the object regions. In our case, our implementation can run at 15~16 frames/s in the first and the second experiment. However, in the football sequence showed in the third experiment, we can track three objects that run at 8 frames/s. Thus the main drawback of the proposed method is that the increasing of number of objects makes the processing time become longer.

6.4 CONCLUSIONS

This paper presented a method to track multiple objects employing color-based particle filter in the presence of occlusion. The performance of the tracking algorithm was evaluated by experiments on several sequences. From the experimental results, the proposed algorithm can successfully track the moving objects. Our algorithm can handle partial and fully object occlusion by redefining the resample and target update step of the occluded object. The occlusion is predicted based on the dynamical model and likelihood measurement. The robust color likelihood is used to evaluate samples properly associated to the target which present high appearance variability. We rely on Bhattacharya coefficient between target and sample histogram to perform this task. Model updating is carried to update the object in the presence of appearance change.

However, from the experiments, we find most of computations are spent on building color histogram of the object. The processing time becomes a problem need to be overcome. It can be improved in several ways. For example, the number of required particles could be reduced by adopting a better proposal density for existing particles and a better prior density of appearing objects. The probabilistic exclusion principle proposed by Mac Cormick and Blake [72] is very useful for tracking multiple objects, which employs the partition sampling to reduce the high computational cost from fully coupled systems. A combination of Mac Cormick and Blake's technique and ours will make the multiple objects tracking more efficient and more robust. Currently the orientation of the targets is fixed during the tracking. The possible solution is to assign an orientation to the tracked objects using the edge orientation histograms as in [39]. These are remained for our future works.

CHAPTER VII

CONCLUSIONS AND FUTURE WORKS

CHAPTER VII
CONCLUSIONS AND FUTURE WORKS

7.1 CONCLUSIONS

In this thesis we have focused on study on object tracking in outdoor environment using video camera. Several aspects of object tracking algorithm have been considered in this thesis: (1) detection and tracking the interest object appear on the scene or field of view of camera, (2) tracking of the interest object employing multi-camera (3) tracking the moving object based on particle filter and (4) expanding the particle filter algorithm to track multiple objects with occlusion appear on the scene. Our tracking system is based on two category of object tracking: the deterministic and stochastic method. In the deterministic method, we applied frame difference method on low resolution image and PISC image to detect the moving object. The object tracking is performed using block matching technique. In the stochastic method, we applied particle filter algorithm. We performed the particle filter algorithm by integrating it with the color feature. We also performed the multiple objects tracking using the particle filter algorithm.

We start our thesis by investigating the related object tracking system methodology and the future development of the tracking system. We have presented various recent methodology of the tracking system including object detection, object tracking and object recognition. On each methodology, we presented detail methodology with the weakness and the strong points. Furthermore, we presented also the future development of the tracking system.

We proposed a method to track the interest object and its identification method. Our proposed method is implemented on the three steps of the tracking system. First, we proposed low resolution image to remove the fake motion on the background. Second, we proposed block matching based on PISC image to track the interest object and finally we proposed spatial and color feature to identify the interest object. Our proposed method successfully track the interest object among two and three objects appear in the scene. We achieve satisfactory results with identification rate of 92.1 [%] in average.

To overcome the limitation of object tracking using single camera, we propose to expand the algorithm by implementing multi-camera to track the interest object. On this algorithm, we utilize the PISC image to detect the moving object and the interest object is tracked using block matching technique. The multi camera system is implemented under LAN environment which each camera is connected to each PC. The communication between PCs is done using socket programming. We successfully track the interest object using two cameras and achieve the detection rates of 97.23[%] and accuracy of 96.98[%] in average.

We presented also object tracking method based particle filter algorithm. We proposed to integrate the color feature measurement to the particle filter algorithm. In addition, to handle the appearance change and background clutter, we proposed also model updating. We analyzed the effect of the number of particles and number of histogram bins to the processing time and tracking accuracy. We obtained that the processing time is related to the number of particles and number of histogram although the tracking accuracy increase also. The experimental results show the algorithm can successfully track the single moving object based on known and unknown initial position and object appearance.

Finally, we proposed to expand the color-based particle filter algorithm to track multiple objects in the presence of occlusion. We proposed to handle occlusion by redefining the resample and model update step. The occlusion itself is predicted based on the estimated position and likelihood measurement. We implemented our proposed algorithm to track two and three moving object which has individual template and independent motion and the satisfactory results are achieved. We found also that the processing time is related to the number of the objects to be tracked.

7.2 FUTURE WORKS

As mentioned in the previous chapters, our tracking system still has many problems need to be solved in the future. Several points are described below as the future researches.

The first aspect of our future work is related to the tracking of the interest object using single camera. Firstly, the low resolution image can successfully remove the fake motion on the background. However, when the interest object becomes small, the low resolution image cannot distinguish between the interest moving object and fake motion object. To solve this problem, we need to add another feature or method such as skin color detection to distinguish the moving object (person) from other non-interest object. Secondly, the block matching technique has successfully tracked the interest moving object in the presence of occlusion by the assumption that the first object will be tracked as the interest object. However, when the moving objects appear in the same time, we cannot judge any object to be an interest object. Moreover, when the interest moving object is covered by the occluded object, the image information of the interest moving object cannot be obtained. We can solve those problems by adding other information to the interest moving object such as flow of moving object based on optical flow, dimension or another feature. We can also add the color feature model to each object. So whenever the objects appear, they have their own model that different from each other. Thirdly, color and spatial information method show the high correct identification rate. However, the system still cannot identify the objects sometimes when they are just entering or leaving the scene. The system also has limitation when the object is moving slowly. In this condition, the inter-frame difference image of the object will become smaller and we will get smaller bounding box and less moving pixels. This problem can be improved by a better feature selection method for example using SIFT features [39]. The method can be used for object recognition by matching individual features to a database of features from known objects using a fast nearest-neighbor algorithm, followed by a Hough transform to identify clusters belonging to a single object and finally performing verification through least-squares solution for consistent pose parameters.

The second aspect is object tracking using multi-camera. We successfully get the wider FOV. However, the proposed tracking system still has some miss-tracking. One of the solutions is implementing the wider range of tracking camera enabled by using more cameras connected together or by using wireless cameras. And also improving the tracking algorithm, such as tracking multi-object by implementing the statistical method, could lead to some improvements in the tracking system.

The third aspect is related to particle filter-based tracking algorithm. In this algorithm, we still found miss-tracking on some frames. Some of them are due to the object features used for the measurement of likelihood observation are not represented the perfect object. This condition occurs when the object just enters the scene. Therefore only part of the object can be measured. The other reason is the optimization done by particle filter algorithm did not go fast to adapt the change whether the object already appeared or not. We can improve the algorithm by redefining the object feature for likelihood measurement. We can add other features such as edge, texture and so on.

The fourth remaining works is related to the multiple objects tracking algorithm. We successfully track up to three objects by our particle filter algorithm. However, we found the processing time becomes a problem need to be solved. The processing time is related to the number of the object to be tracked. We can improve this problem in several ways. For example, the processing time could be reduced by applying another feature to the likelihood model and the number of required particles could be reduced by adopting a better proposal density for existing particles and a better prior density of appearing objects.

And finally, we need to develop a tracking system that not only detect and track the moving objects, but also recognize and analyze their actions. The main idea is to construct such a surveillance system, which detects moving objects and tracks them. Furthermore, it also recognizes and analyzes some interactions between the objects and the real environment. Actions such as walking, stopping, meeting and waiting are easily interpreted from trajectories and time lines. By learning the expected actions, the non-expected actions and behaviors could be recognized automatically. This can be also applied to smart rooms, intelligent communication and intelligent transport system (ITS) and so on. We believe that the system we have developed in this thesis can be applied in those systems and can make the society become better.

REFERENCES

REFERENCES

1. M. Black and A. Jepson, “A probabilistic framework for matching temporal trajectories: condensation-based recognition of gestures and expressions”, *Proc of the European Conference on Computer Vision*, pp. 909-924, 1998.
2. B. Menser and M. Brunig, “Face detection and tracking for video coding applications”, *Asilomar Conference on Signals, Systems, and Computers*, pp. 49-53, 2000.
3. D. Beymer, P. McLauchlan, B. Coifman and J. Malik, “A real-time computer vision system for measuring traffic parameters”, *Proc of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 495-501, 1997.
4. M. Greiffenhagen, V. Ramesh, D. Comaniciu and H. Niemann, “Statistical modeling and performance characterization of a real-time dual camera surveillance system”, *Proc of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 335-342, 2000.
5. A. Utsumi, H. Yang and J. Ohya, “Adaptive human motion tracking using non-synchronous multiple viewpoint observations”, *Proc. of the Int’l Conf. on Pattern Recognition*, pp. 607-610, 2000.
6. A. Yilmaz, O. Javed and M. Shah, “Object tracking: a survey”, *ACM Computing Survey*, vol.38, no.13, pp.1-45, 2006.
7. W. Hu, T. Tan, L. Wang and S. Maybank, “A survey on visual surveillance of object motion and behaviors”, *IEEE Trans. on System, Man, and Cybernetics*, vol. 34, no. 3, pp. 334-352, 2004.
8. K. Nummiaro, K. Esther and L. Gool, “Object tracking with an adaptive color-based particle filter”, *Proc. of Symposium for Pattern Recognition of the DAGM*, pp. 353-360, 2002.
9. J. Czyz, B. Ristic and B. Macq, “A particle filter for joint detection and tracking of color objects”, *Image and Vision Computing*, vol. 25, no. 8, pp. 1271-1281, 2007.
10. G. Paschos, “Perceptually uniform color spaces for color texture analysis: an empirical evaluation”, *IEEE Trans. on Image Proces*sing, vol. 10, no.6, pp. 932-937, 2001.
11. P. Perez, C. Hue, J. Vermaak and M. Gangnet, “Color-based probabilistic tracking”, *Proc. of the 7th European Conference on Computer Vision*, pp. 661-675, 2002.
12. T, Nakamura and T, Ogasawara, “On-line visual learning for color image segmentation and tracking based on fuzzy ART model”, *Proc. of the Int’l conf. on Intelligent Robot and System*, vol. 1, pp. 222-228, 1999.
13. J. Canny, “A computational approach to edge detection”, *IEEE Trans. on Pattern Analysis and Machine Intelligent*, vol. 8, no. 6, pp. 679-698, 1986.
14. B. Horn and B. Schunk, “Determining optical flow”, *Artificial Intelligent*, vol.17, no.1, pp. 185–203, 1981.
15. B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision”, *Proc of the Int’l Joint Conference on Artificial Intelligence*, pp. 674-679, 1981.
16. M. Black and P. Anandan, “The robust estimation of multiple motions: Parametric and piecewise smooth flow fields”, *Computer Vision and Image Understanding*, vol. 63, no. 1, pp. 75-104, 1996.

17. R. Szeliski and J. Coughlan, “Spline-based image registration”, *Int’l J. of Computer Vision*, vol. 22, no.3, pp. 199-218, 1997.
18. R. Haralick, B. Shanmugam and I. Dinsten, “Textural features for image classification”, *IEEE Trans. on System, Man and Cybernetic*, vol. 33, no. 3, pp. 610-622, 1973.
19. S. Mallat, “A theory for multi-resolution signal decomposition: The wavelet representation”, *IEEE Trans. on Pattern Analysis and Machine Intelligent*, vol. 11, no. 7, pp. 674-693, 1989.
20. H. Greenspan, S. Belongie, P. Perona, S. Rakshit and C. Anderson, “Over complete steerable pyramid filters and rotation invariance”, *Proc of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 222-228, 1994.
21. M.J. Ferreira, C.P. Santos and J. Monteiro, “Texture cue based tracking system using wavelet transform and a fuzzy grammar”, *Proc. of the 5th IEEE Int’l Conf on Industrial Informatics*, pp. 393-398, 2007.
22. C. Shen, A. Hengel and A. Dick, “Probabilistic multiple cue integration for particle filter based tracking”, *Proc of the Digital Image Computing: Techniques and Applications*, pp. 399-408, 2003.
23. N. Ki and E. Delp, “New models for real time tracking using particle filter”, *Proc. of the SPIE Conf. on Visual Communications and Image Processing*, vol. 7257, 12 pages, 2009.
24. P. Brasnett and L. Mihaylova, N. Canagarajah and D. Bull, “Particle filtering with multiple cues for object tracking in video sequences”, *Proc. of the SPIE Conf. on Electronic Imaging, Science and Technology*, vol. 5685, pp. 430-441, 2005.
25. Q.Cai, A. Mitiche, J.K. Aggarwal, “Tracking human motion in indoor environment”, *Proc of the Int’l Conf. on Image Processing*, pp. 215-218, 1995.
26. Q Cai, and J.K Aggarwal, “Tracking human motion using multiple cameras”, *Proc. of the Int’l. Conf. on Pattern Recognition*, pp. 68-72, 1996.
27. D. Beymer and Konolige, “Real-time tracking of multiple people using continuous detection”, *Proc. of the Int’l Conf. on Computer Vision*, 8 pages, 1999.
28. C. Pai, H. Tyan, Y. Liang, H. Liao and S. Chen, “Pedestrian detection and tracking at crossroads”, *Proc. of the Int’l Conf. on Image Processing*, pp. 101-104, 2003.
29. Z. Qui, D. Yao, Y. Zhang, D. Ma and X. Liu, “The study of the detection of pedestrian and bicycle using image processing”, *Proc. of the IEEE Intelligent Transportation Systems*, vol.1, pp.340-345, 2003.
30. B. Sugandi, H. Kim, J. K. Tan and S. Ishikawa, “Tracking of moving persons using multi-camera employing peripheral increment sign correlation image” *ICICIC Express letter*, vol. 1, no.2, pp. 177-183, 2007.
31. R. Cucchiara, “Multimedia surveillance systems”, *Proc. of the 3rd ACM Int’l Workshop on Video Surveillance & Sensor Networks*, pp. 3-10, 2005.
32. A. Prati, R. Vezzani, L. Benini, E. Farella and P. Zappi, “An integrated multi-modal sensor network for video surveillance”, *Proc. of the 3rd ACM Int’l Workshop on Video Surveillance & Sensor Networks*, pp. 95–102, 2005.

33. Y. Wang, Y. Chang, and K. Cheng, "A video analysis framework for soft biometry security surveillance", *Proc. of the 3rd ACM Int'l Workshop on Video Surveillance & Sensor Networks*, pp. 71-78, 2005.
34. D. Fidaleo and M. Trivedi, "Manifold analysis of facial gestures for face recognition", *Proc. of the ACM SIGMM Workshop on Biometrics Methods and Applications*, pp. 65-69, 2003.
35. J. Carranza, Ch. Theobalt, M. A. Magnor and H.-P. Seidel, "Free-viewpoint video of human actors", *ACM Trans. on Graphics*, vol. 22, no.3, pp. 569-577, 2003.
36. R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto and Hasegawa, "System for video surveillance and monitoring", *Technical Report CMU-RI-TR-00-12*, Robotics Institute, Carnegie Mellon University, 2000.
37. H. Moravec, "Visual mapping by a robot rover", *Proc. of the Int'l Joint Conference on Artificial Intelligence*,, pp. 598-600, 1979.
38. J. Shi and C. Tomasi, "Good features to track", *Proc of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 593-600, 1994.
39. D. Lowe, "Distinctive image features from scale-invariant key points". *Int'l. J. of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.
40. K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector", *Proc of the European Conference on Computer Vision*, pp. 128-142, 2002.
41. A. M. McIvor, "Background subtraction techniques", *Proc. of the Image and Vision Computing*, 6 pages, 2000.
42. S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld and H. Wechsler, "Tracking groups of people", *Computer Vision and Image Understanding*, vol. 80, no. 1, pp. 42-56, 2000.
43. C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking", *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 246-252, 1999.
44. J. Heikkila and O. Silven, "A real-time system for monitoring of cyclists and pedestrians", *Proc. of the 2nd IEEE Workshop on Visual Surveillance*, pp. 74-81, 1999.
45. Y. Satoh, S kaneko and S Igarashi, "Robust object detection and segmentation by peripheral increment sign correlation image", *System and Computer in Japan*, vol. J84-D-II, no 12, pp. 2585-2594, 2001.
46. A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving target classification and tracking from real-time video", *Proc. of the IEEE Workshop Applications of Computer Vision*, pp. 8-14, 1998.
47. B. Sugandi, H. Kim, J. Tan and S. Ishikawa, "Real time tracking and identification of moving object by using a camera in outdoor environment", *Int'l Journal of Innovative Computing, Information and Control (IJICIC*), vol. 5, no. 5, pp. 1179-1188, 2009.
48. W. Hu, L. Wang and T. Tan, "Recent developments in human motion analysis", *Pattern Recognition*, vol. 36, no. 3. pp. 585-601, 2003.
49. D. Harwood, I. Haritaoglu and L. S. Davis, "W4: Real-time surveillance of people and their activities", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809-830, 2000.

50. C. Stauffer and W Grimson, “Learning patterns of activity using real-time tracking”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747-757, 2000.
51. D. Meyer, J. Denzler, and H. Niemann, “Model based extraction of articulated objects in image sequences for gait analysis”, *Proc. of the IEEE Int'l Conf. on Image Processing*, pp. 78-81, 1998.
52. J. Barron, D. Fleet, and S. Beauchemin, “Performance of optical flow techniques,” *Int'l J. of Computer Vision*, vol. 12, no. 1, pp. 42-77, 1994.
53. Y. Dai and Y. Nakano, “Face-texture model-based on SGDL and its application in face detection in a color scene”, *Pattern Recognition*, vol. 29, no. 6, pp. 1007-1017, 1996.
54. R. Hsu, M. Mottaleb and K. Jain, “Face detection in color images”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 696-706, 2002.
55. Garcia and G. Tziritas, “Face detection using quantified skin color regions merging and wavelet packet analysis”, *IEEE Trans. on Multimedia*, vol. 1, no. 3, pp. 264-277, 1999.
56. M. Jones and J. Rehg, “Statistical color models with application to skin detection”, *Int'l J. of Computer Vision*, vol. 46, no. 1, pp. 81-96, 2002.
57. M. Yang and N. Ahuja, “Gaussian mixture model for human skin color and its applications in image and video databases”, *SPIE Proceedings on Storage and Retrieval for Image and Video Databases VII*, pp. 458-466, 1999.
58. Q. Huynh-Thu, M. Meguro and M. Kaneko, “Skin color extraction in images with complex background and varying illumination”, *IEEE Workshop on Applications of Computer Vision*, pp. 280-285, 2002.
59. S. Phung, D. Chai and A. Bouzerdoum, “Adaptive skin segmentation in color images”, *Proc of the IEEE Int'l Conf. on Acoustics, Speech and Signal Processing*, vol. 3, pp. 353-356, 2003.
60. K. M. Cho, J. H. Jang and K. S. Hong, “Adaptive skin-color filter”, *Pattern Recognition*, vol. 34, no. 5, pp. 1067-1073, 2001.
61. A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training”, *Proc of the 11th Annual Conf. on Computational Learning Theory*, pp. 92-100, 1998.
62. A. Levin, P. Viola and Y. Freund, “Unsupervised improvement of visual detectors using co-training”, *Proc of the IEEE Int'l Conf. on Computer Vision (ICCV)*, pp. 626-633, 2003.
63. M. Kockelkorn, A. Luneburg and T. Scheffer, “Using transduction and multi-view learning to answer emails”, *Proc of the European Conference on Principle and Practice of Knowledge Discovery in Databases*, pp. 266-277, 2003.
64. G. Welch, B. D. Allen, A. Ilie and G. Bishop, “Measurement sample time optimization for human motion tracking/capture systems”, *Proc. of the Trends and Issues in Tracking for Virtual Environments, Workshop at the IEEE Virtual Reality 2007 Conference*, 6 pages, 2007.
65. J. T. Reindeers J. Veenman and E. Backner, “Resolving motion correspondence for densely moving points”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 1, pp. 54-72, 2001.
66. T. Broida and R. Chellappa, “Estimation of object motion parameters from noisy images”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, no. 1, pp. 90-99, 1986.

67. G. Kitagawa, “Non-Gaussian state-space modeling of non-stationary time series”, *J. of American Statistic Association*, vol. 82, no. 400, pp. 1032-1063, 1987.

68. Y. Shalom and T. Foreman, “Tracking and Data Association”, *Mathematics in science and engineering*, vol. 179, pp. 331-346, 1988.

69. M. Isard and A. Blake, “Condensation - conditional density propagation for visual tracking”, *Int. J. of Computer Vision*, vol. 29, no.1, pp. 5-28, 1998.

70. D. Comaniciu and Peter Meer, “Mean shift: A robust approach toward feature space analysis”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603-619, 2002.

71. D. Huttenlocher, J. Noh, and W. Rucklidge, “Tracking non-rigid objects in complex scenes”, *IEEE Int'l Conf. on Computer Vision (ICCV)*, pp. 93-101 1993.

72. Mac Cormick, and A. Blake, “Probabilistic exclusion and partitioned sampling for multiple object tracking”, *Int. J. of Computer Vision*, vol. 39, no. 1, pp. 57-71, 2000.

73. A. Yilmaz, X, Li, and M. Shah, “Contour based object tracking with occlusion handling in video acquired using mobile cameras”, *IEEE Trans. on Pattern. Analysis and Machine Intelligent*, vol. 26, no. 11, pp. 1531-1536, 2004.

74. M. Isard and A. Blake, “Contour tracking by stochastic propagation of conditional density”, *Proc. of the European Conference on Computer Vision*, pp. 343-356, 1996.

75. C. C. Fung and N. Jerrat, “A neural network based intelligent intruders detection and tracking system using CCTV images”, *Proc. of the TENCON*, vol.2, pp. 409-414, 2000.

76. Y. Matsushita, K. Nishino, K. Ikeuchi and M. Sakauchi, “Shadow elimination for robust video surveillance”, *Proc. of the Workshop on Motion and Video Computing*, pp. 15-21, 2002.

77. C. Micheloni and G. L. Foresti, “Fast good features selection for wide area monitoring”, *Proc. of the IEEE Conf. on Advanced Video and Signal Based Surveillance*, pp. 271-276, 2003.

78. Q. Wang, G. Xu and H. Ai, “Learning object intrinsic structure for robust visual tracking”, *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, vol.2, pp.227-233, 2003.

79. S. Park and J. K. Aggarwal, “Recognition of two-person interactions using a hierarchical Bayesian network”, *Proc. of the 1st ACM SIGMM Int'l Workshop on Video Surveillance*, pp. 65-76, 2003.

80. N. Gralewski, B. Campbell, C. Thomas, and D. Gibson, “Statistical synthesis of facial expressions for the portrayal of emotion”, *Proc. of the 2nd Int'l Conf. on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, pp. 190-198, 2004.

81. H. Gunes, M. Piccardi and T. Jan, “Face and body gesture recognition for a vision-based multimodal analyzer”, *Proc. of the Pan-Sydney Area Workshop on Visual Information Processing*, pp. 19-28, 2004.

82. M. Leo, T. D'Orazio and P. Spagnolo, “Human activity recognition for automatic visual surveillance of wide areas”, *Proc. of the ACM 2nd Int'l Workshop on Video Surveillance and Sensor Networks*, pp.124-130, 2004.

83. S. G. Gong and H. Buxton, “Editorial: understanding visual behavior”, *Image and Vision Computing*, vol. 20, no. 12, pp. 825-826, 2002.

84. D. Makris and T. Ellis, “Path detection in video surveillance”, *Image and Vision Computing*, vol. 20, no. 12, pp. 895-903, 2002.
85. J. Fernyhough, A. G. Cohn, and D. C. Hogg, “Constructing qualitative event models automatically from video input”, *Image and Vision Computing*, vol. 18, no. 9, pp. 81-103, 2000.
86. Y. Seto, “A trend and vision of biometric security technology”, *J. of Information Processing Society of Japan*, vol.47, no.7, pp. 571-576, 2006.
87. T. Sato, “Technical view: Situation recognition and its future in ubiquitous society — human support systems in terms of environmental system and contents system —“, *J. of Systems Control and Information*, vol.49, no.4, pp. 113-118, 2005.
88. F. Bartoleni, A. Tefas, M. Barni and I. Pitas, “Image authentication techniques for surveillance applications,” *Proc. of the IEEE on 3rd Generation of Surveillance System*, vol. 89, no.10, pp. 1403-1417, 2001.
89. B. Sugandi, H. Kim, J. Tan and S. Ishikawa, “A color feature-based method for object tracking employing a particle filter algorithm” *Proc. of the 2nd Global Conference on Power Control and Optimization*, pp. 206-211, 2009.
90. M. Ahad, T. Ogata, J. K. Tan, H. Kim and S. Ishikawa, “View-based Human Motion Recognition in the Presence of Outliers”, *Biomedical Soft Computing and Human Sciences*, vol. 13, no.1, pp. 71-78, 2008.
91. G. Kitagawa, “Monte Carlo filter and smoother for non-Gaussian nonlinear state space models”, *J. of Computational and Graphical Statistic*, vol. 5, no. 1, pp. 1-25, 1996.
92. N. Gordon, D. Salmon and A. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation”, *IEE Proc. of Radar and Signal Processing*, vol. 140, no. 2, pp.107-113, 1993.
93. F. Cheng and Y. Chen, “Real time multiple objects tracking and identification based on discrete wavelet transform”, *J. of the Pattern Recognition Society*, pp 1126-1139, 2006.
94. R. C. Gonzalez and R. E. Woods, “*Digital Image Processing*”, Addison-Wesley Longman Publishing Co., Inc.,Boston, MA., 2001.
95. Y. LIU, A. Haizho and G. Xu, “Moving object detection and tracking based on background subtraction”, *Proc. of the Society of Photo-Optical Instrument Engineers (SPIE),* vol. 4554, pp. 62-66, 2001.
96. S. M. Desa, and Q. A. Salih, “Image subtraction for real time moving object extraction”, *Proc. of the Int’l Conf. on Computer Graphics, Imaging and Visualization (CGIV’04)*, pp. 41-45, 2004.
97. E. Stringa, “Morphological change detection algorithms for surveillance applications”, *Proc. of the British Machine Vision Conf.*, pp. 402-412, 2000.
98. S. Khan, J. Omar and M. Shah, “Tracking in un-calibrated cameras with overlapping field of view”, *The 2nd IEEE Workshop on Performance Evaluation of Tracking and Surveillance* (*PETS2001*), 8 pages, 2001.
99. T. Chang and S.Gong, “Tracking multiple people with a multi-camera system”, *Proc of the IEEE Workshop on Multi-Object Tracking*, pp. 19-26, 2001.

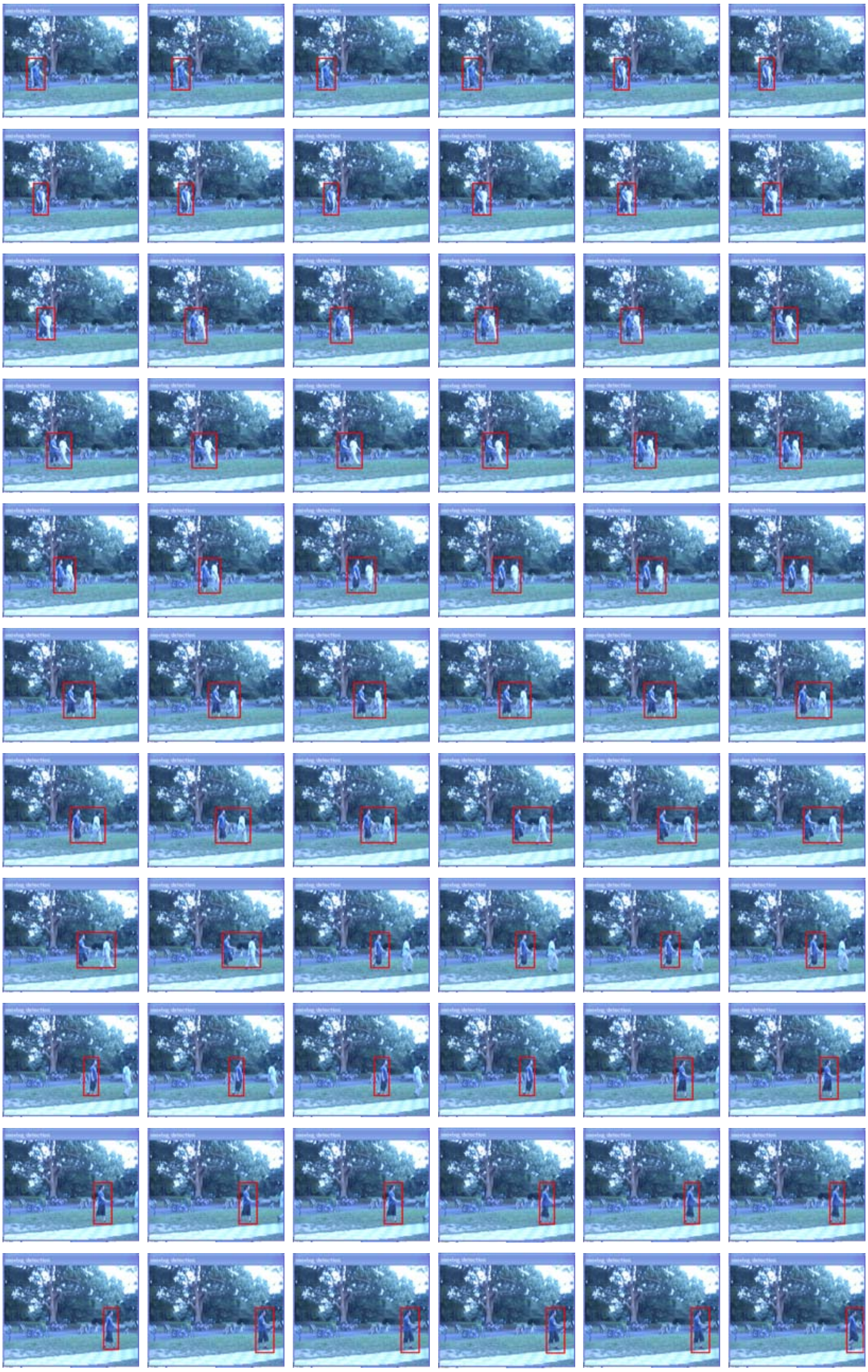
100. B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: particle filters for tracking applications*, Artech House, 2004.
101. S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A tutorial on particle Filters for on-Line nonlinear/non-Gaussian Bayesian tracking", *IEEE Trans. on Signal Processing*, vol. 50, no.2. pp. 174-188, 2002.
102. T. Broida and R. Chellappa, "Estimation of object motion parameters from noisy images", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, no. 1, pp. 90-99, 1986.
103. G. Kitagawa, "Non-Gaussian state-space modeling of non-stationary time series", *Journal of American Statistic Association*, vol. 82, no. 400, pp. 1032-1063, 1987.
104. D. Simon, *Optimal State Estimation*, Jhon Wiey & Sons, Inc, Publication, 2006
105. W. Freeman and E. Adelson, "The design and use of steerable filters", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891-906, 1991.
106. F. Bashir and F. Porikli, "Performance evaluation of object detection and tracking systems", *IEEE Int'l Workshop on Performance Evaluation of Tracking and Surveillance* (PETS), 8 pages, 2006.
107. M. Isard, and. A. Blake, "A mixed-state condensation tracker with automatic model-switching", *Proc. of the Int'l Conf. on Computer Vision*, pp. 107-112, 1998.
108. M. Isard, and J. MacCormick, "Bramble: a Bayesian multiple blob tracker", *Proc. of the Int'l Conf. on Computer Vision*, pp. 34-41, 2001.
109. J. Vermaak, A. Doucet, and P. Perez, "Maintaining multimodality through mixture tracking", *Proc. of the Int'l Conf. on Computer Vision*, pp. 1110-1116, 2003.
110. K. Okuma, A. Taleghani, N. de Freitas, J.J. Little, and D.G. Lowe, "A boosted particle filter: multi-target detection and tracking", *Proc. of the European Conf. on Computer Vision (ECCV)*, pp. 28-39, 2004.
111. K. Smith, D. Perez, and J. M. Odobez, "Using particles to track varying numbers of interacting people", *Proc of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 962-969, 2005.
112. F. Daum, J. Huang, "Curse of dimensionality and particle filters", *Proc. of the IEEE Aerospace Conference*, pp. 1979-1993, 2003.

APPENDIX

APPENDIX I

TRACKING THE INTEREST OBJECT AND ITS IDENTIFICATION METHOD

1.1. Tracking the interest object from two objects move in the same direction



1.2. Tracking the interest object from two objects move in different direction

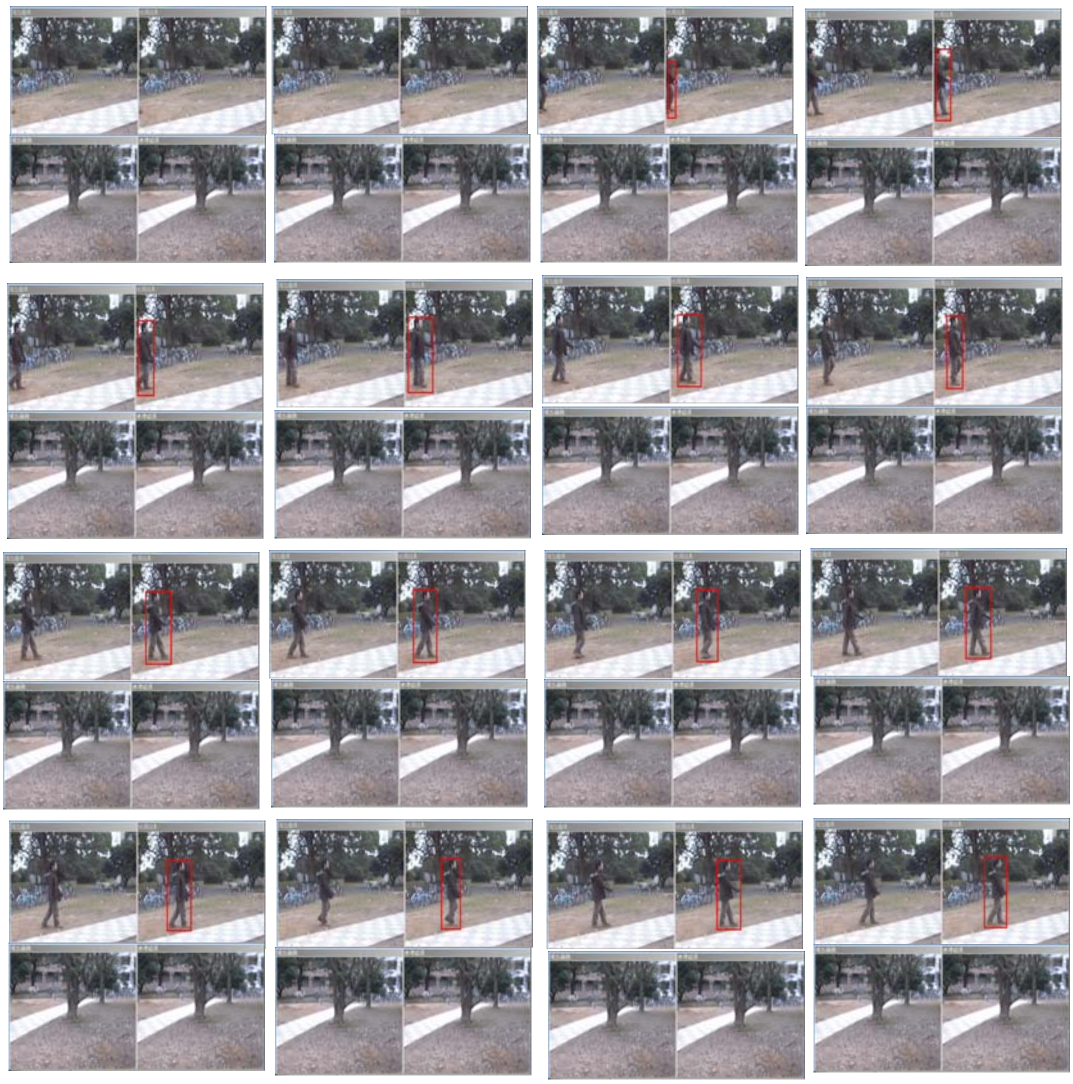
1.3. Tracking the interest object from three objects appear on the scene

APPENDIX II

OBJECT TRACKING USING MULTI-CAMERA UNDER OUTDOOR ENVIRONMENT

2.1. Tracking the interest object using multi-camera from two object move on different direction

C_1 : camera 1 and C2 : camera 2



C_1

C_2

C_1

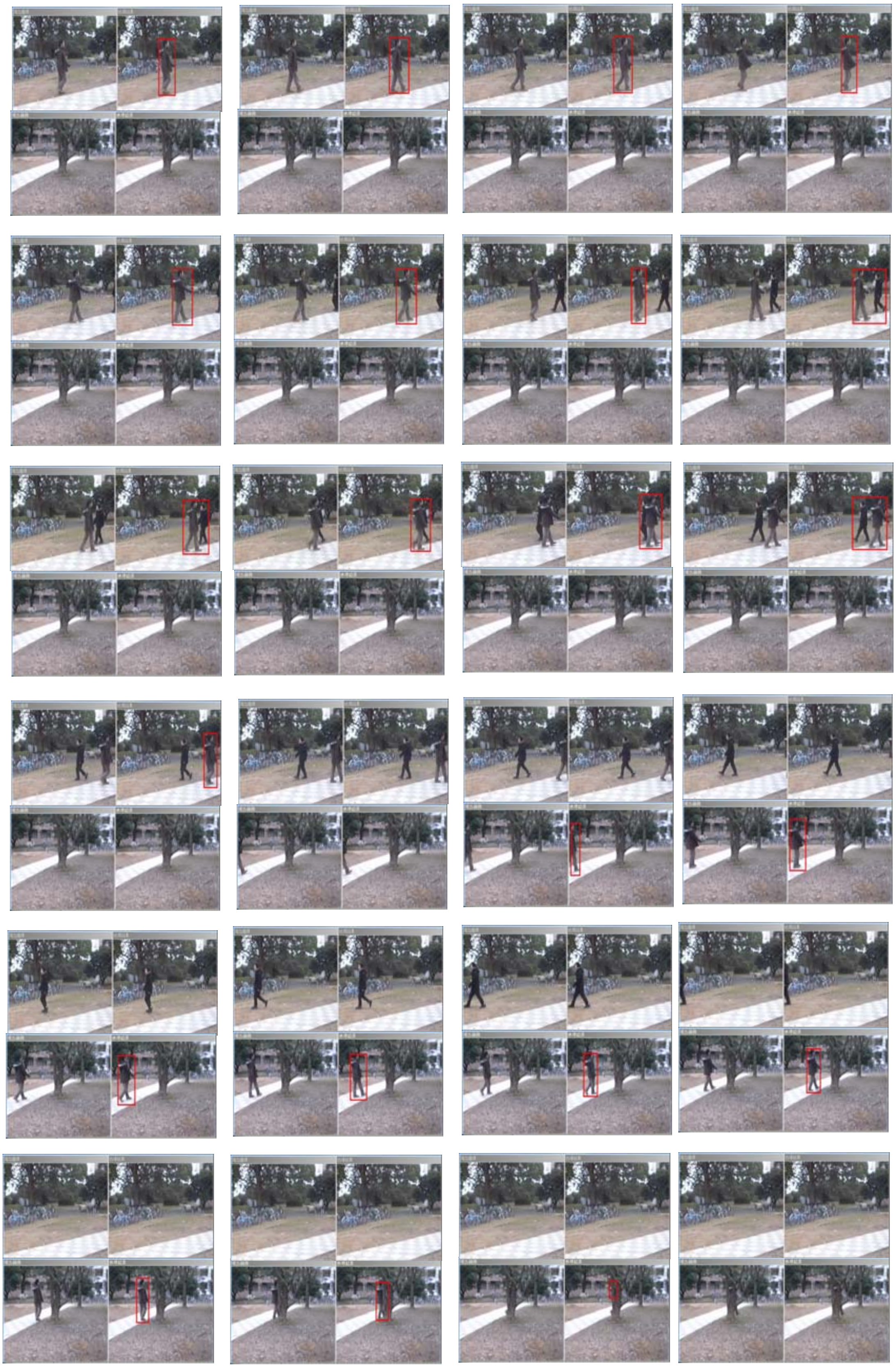
C_2

C_1

C_2

C_1

C_2



C_1

C_2

C_1

C_2

C_1

C_2

C_1

C_2

C_1

C_2

C_1

C_2

C_1

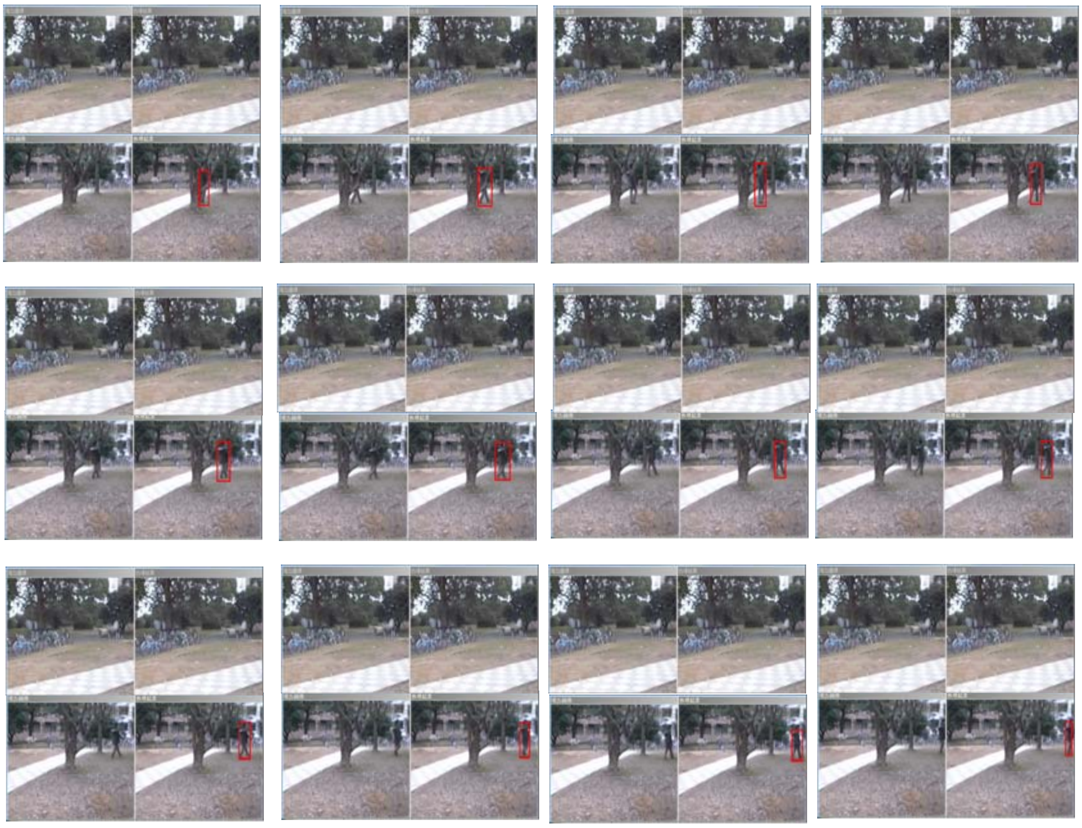
C_2

C_1

C_2

C_1

C_2



2.2. Tracking the interest object using multi-camera from two object move on same direction

C_1

C_2

C_1

C_2

C_1

C_2

C_1

C_2

C_1

C_2

C_1

C_2

C_1

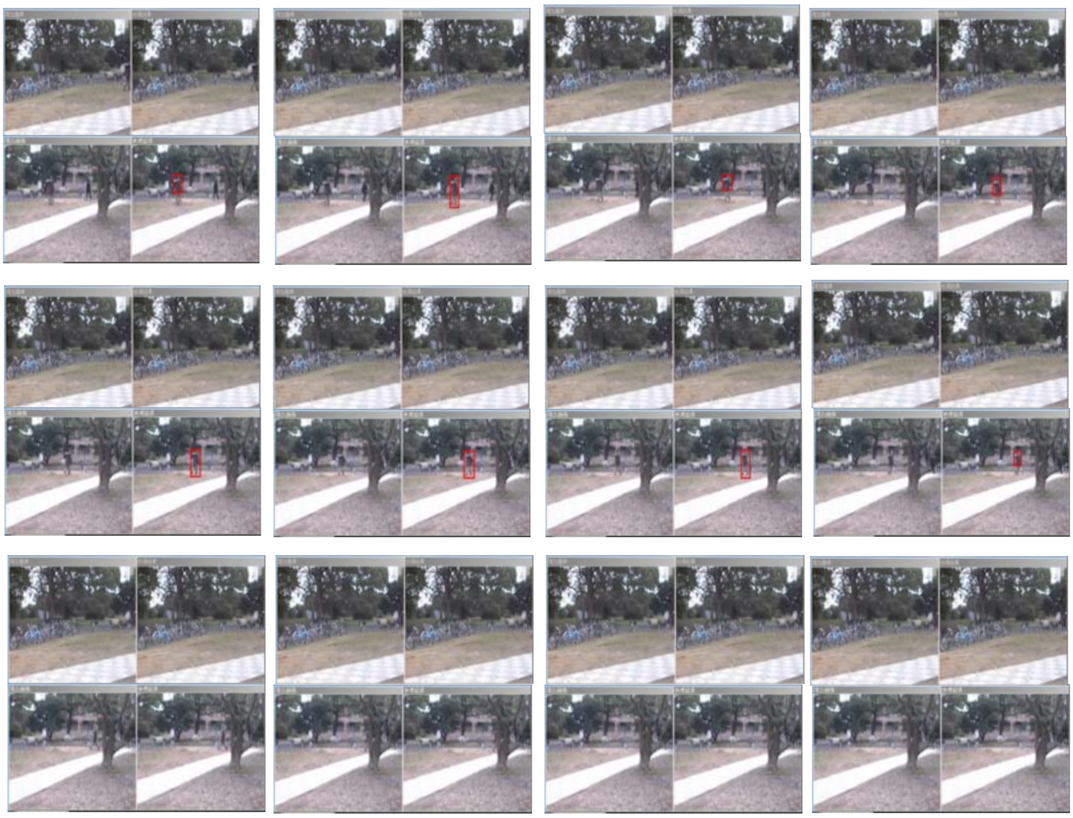
C_2

C_1

C_2

C_1

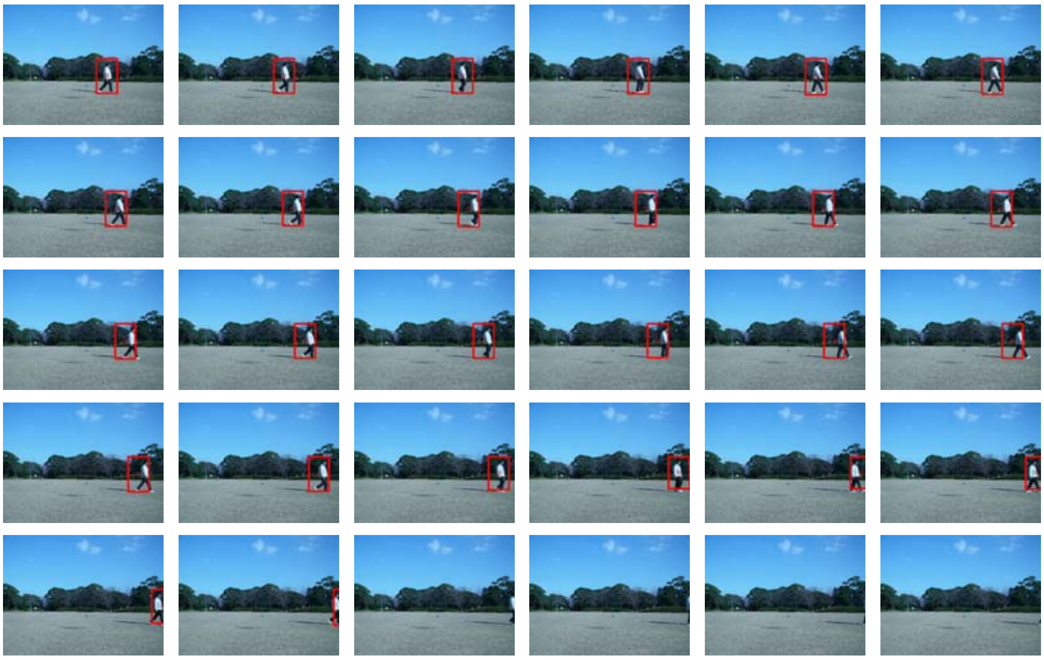
C_2



APPENDIX III

OBJECT TRACKING BASED ON PARTICLE FILTER ALGORITHM

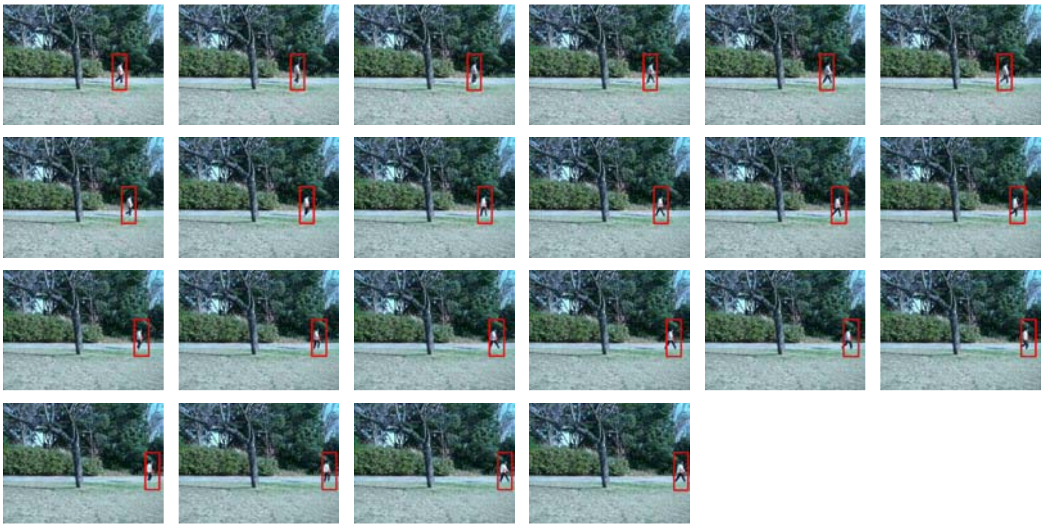
3.1. Object tracking with random initial samples position



3.2. Object tracking with known initial samples position

3.3. Object tracking with update scale

3.4. Object tracking in the presence of occlusion based on appearance condition



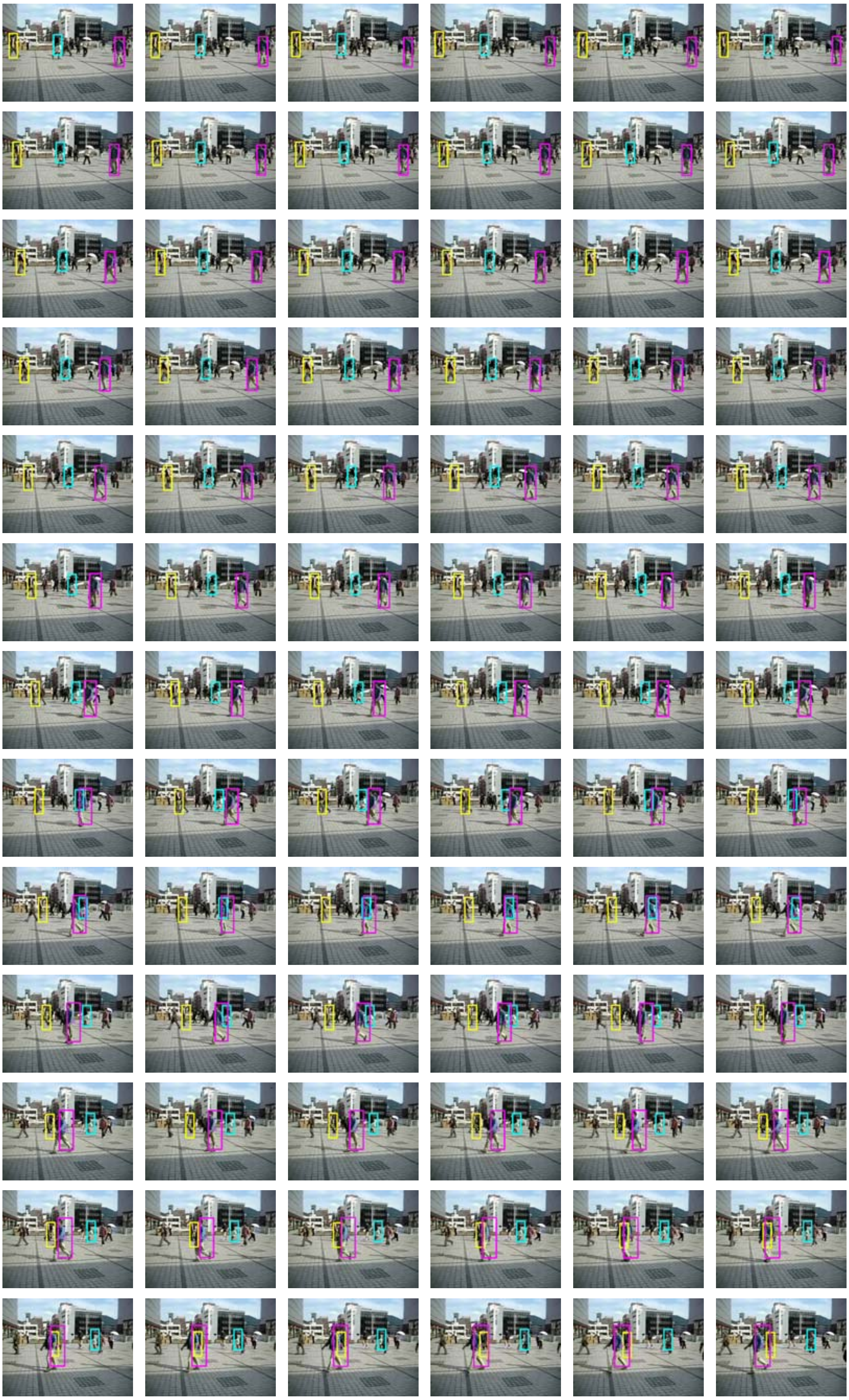
APPENDIX IV
MULTIPLE OBJECTS TRACKING BASED ON PARTICLE FILTER ALGORITHM

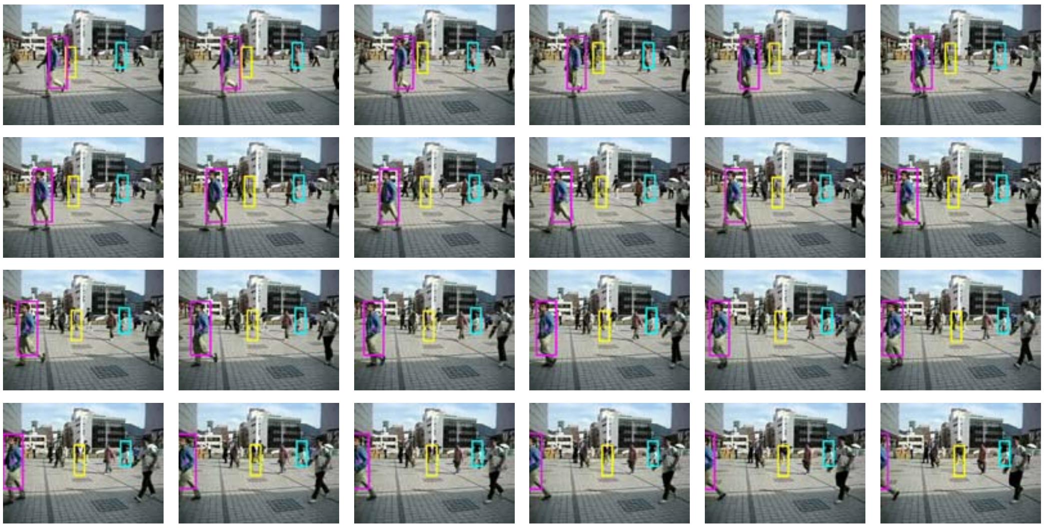
4.1. Multiple objects tracking in the presence of occlusion

4.2. Multiple objects tracking of soccer players

The video is downloaded from http://www.youtube.com/watch?v=zZ5tt6_guR4.

4.3. Multiple objects tracking in the complex environment





氏　　　名	ブディ スガンディ
学位の種類	博　士(工学)
学位記番号	工博甲第　号　(論博の場合は、工博乙第　号)
学位授与の日付	平成　23年　3月　25日
学位授与の条件	学位規則第4条第1項該当(論博の場合は「第2項」該当)
学位論文題目	Study on object tracking under outdoor environments using video cameras (ビデオカメラを用いた屋外環境下での物体追跡に関する研究)
論文審査委員	主　査　准 教 授　　金　　亨燮 教 授　　石川　聖二 教 授　　田川　善彦 教 授　　芹川　聖一

学 位 論 文 内 容 の 要 旨

近年，コンピュータビジョンや画像センシング分野においては，セキュリティの向上や運動解析などを目的とした物体の追跡に関する研究が活発に行われている。その中で，視覚情報を用いた追跡システムによる動画像からの移動物体（人物などを含む）の検出や追跡，認識に関する研究が盛んに行われており，一部においては商業化を念頭にしたシステムの開発が進められている。しかし，それらは屋内環境を対象としたものや限られた建物内での利用に関するものが多く，遮蔽や環境の変化にロバストな追跡結果が得られないなどの問題があり，その改善が求められている。本論文では，野外関係での連続するビデオ画像から，移動物体（人物）を自動的に検出，追跡・認識するための画像処理手法を提案し，実験によりその有効性を検討する。

本研究では具体的に，以下の四つの課題を取り上げ，その解析法について述べる。

① 対象物体の検出および追跡手法の開発

② 複数カメラによる対象物体の追跡手法の開発

③ 色特徴とパーティクルフィルタを併用した移動物体の追跡手法の開発

④ パーティクルフィルタを用いたオクルージョンを考慮した複数物体の追跡手法の開発

これらの画像処理技術を開発することにより，様々な状況下での追跡システムの実現が可能となる。また，監視や防犯システムなどのアプリケーションへの適用が可能であり，コンピュータ画像処理や画像計測などの様々な応用が期待できる。

本論文は，以下に示す全七章から構成されている。

第一章では，本論文の序論として研究背景や目的について述べる。

第二章では，移動物体の追跡手法の関連研究および本研究の位置づけについて述べる。現在，物体の検出，物体の追跡，物体の認識について様々な手法が提案されており，本章ではそれらの手法におけるメリットと問題点について詳しく述べる。

第三章では，連続するビデオ画像からの対象物体を自動的に追跡および移動物体の認識を行うための手法を提案する。提案法ではまず，雑音や細かな動きなどによる物体追跡時の精度低下を避けるため，低解像度画像を生成する。また，対象物体を精度よく追跡するため，画像の周辺分布を用いたブロックマッチング法を適用する。次に，画像の空間特徴と色特徴を用いた関心領域の識別を試みる。提案法を用いた認識実験では，連続するフレーム画像上を動く移動物体から関心領域を自動追跡することが可能で，平均 92.1[%]で正しく移動物体を識別することができた。

第四章では，複数カメラによる対象物体の追跡アルゴリズムを提案する。本手法では，移動物体を自動検出するため，画像の周辺分布とマッチング手法を用いる。複数カメラを用いた広範囲における移動人物を追尾するため，複数のカメラと PC で構成されるシステムを LAN 環境で実装し，各 PC 間でシステムを制御するためのソケットプログラミングを実装し, 2 組の PC とビデオカメラからの移動物体を追跡するためのシステムを構築した。実験では，移動物体を 96.98 [%]の精度で検出することが可能であった。

第五章では，パーティクルフィルタによる移動物体の追跡法を提案する。本論文では，パーティクルフィルタアルゴリズムにおける移動人物の追跡に，色情報を付加することにより，よりロバスト性を向上した追跡法を提案していう。また，移動物体の外観の変化，色情報が類似した背景と移動物体との分離にも対応するため，物体のモデル更新法を提案する。実験結果より，提案法は一人の移動人物を正しく追跡することが可能であった。

第六章では，オクルージョンを考慮した複数人物の追跡手法を開発するため，パーティクルフィルタアルゴリズムを改善し，リサンプリングや物体のモデルの更新を再定義する手法を提案した。本法では，オクルージョンの有無を，推定された各パーティクルの位置と尤度から予測する。提案法では，複数の移動人物を正しく追跡をすることが可能であり，実験によりその有用性を確認した。

第七章では，本論文で述べた手法をまとめ，今後の課題や展望などについて述べた。

学位論文審査の結果の要旨

近年，コンピュータビジョンや画像センシング分野においては，セキュリティの向上や運動解析などを目的とした物体の追跡に関する研究が活発に行われている。その中で，視覚情報を用いた追跡システムによる動画像からの移動物体（人物などを含む）の検出や追跡，認識に関する研究が盛んに行われており，一部においては商業化を念頭にしたシステムの開発が進められている。しかし，それらは屋内環境を対象としたものや限られた建物内での利用に関するものが多く，遮蔽や環境の変化にロバストな追跡結果が得られないなどの問題があり，その改善が求められている。本論文では，野外関係での連続するビデオ画像から，移動物体（人物）を自動的に検出，追跡・認識するための画像処理手法を提案し，実験によりその有効性を検討している。

本研究では具体的に，以下の四つの課題を取り上げ，その解析法について述べた。

① 対象物体の検出および追跡手法の開発
② 複数カメラによる対象物体の追跡手法の開発
③ 色特徴とパーティクルフィルタを併用した移動物体の追跡手法の開発
④ パーティクルフィルタを用いたオクルージョンを考慮した複数物体の追跡手法の開発

ビデオカメラからの移動人物を自動追尾するため，まず本論文ではdeterministic法とstochastic法の両面からのアプローチ法を試みている。Deterministic手法では，低解像度画像にフレーム間差分および周辺分布画像（PISC），ブロックマッチング法などを用いた物体の検出を行う手法を提案した。また，Stochastic手法として，パーティクルフィルタを用いた物体の検出を試みた。さらに，色特徴をパーティクルフィルタに適用することにより，オクルージョン問題を解決するためにリサンプリングや物体のモデルの更新を再定義し，オクルージョンの有無を推定した各パーティクルの位置と尤度から予測することにより，遮蔽が生じた場合の人物の追尾や複数人物の追跡を行うための手法を提案し，実験によるその有用性を検証した。

以上より，本研究ではビデオカメラ映像からの移動人物を自動で追跡するため，対象物体の自動抽出法，画像ノイズの低減法，複数PCとカメラ群からなる制御システムを構築するため，ネットワーク環境下での移動人物の追跡法の開発，さらに，色特徴をパーティクルフィルタに加味した人物の追跡法を構築し，野外での遮蔽が生じたビデオ映像からの追尾法を提案した。特に，野外での移動人物の自動追跡法は，複雑な環境下での画像解析分野への応用が可能であり，大きな学術的な知見を与えている。また，従来の単一カメラを用いた物体の追跡における制約を回避するため，複数カメラを用いた対象人物の追跡手法を提案しており，LAN環境下でのPC同士のコミュニケーションをソケットプログラミングにより実現し，広範囲におよ

ぶ追跡を可能にした。これらのことから，情報化社会におけるセキュリティシステムを構築する上で，実用化のための見通しを示している。

これらの画像処理技術を開発することにより，様々な状況下での追跡システムの実現が可能となる。また，監視や防犯システムなどのアプリケーションへの適用が可能であり，コンピュータ画像処理や画像計測などの様々な応用が期待できる。

本論文に関して，公聴会に出席した教員や審査委員から専門的な質問がなされ，いずれのも著者の適切な回答によって理解が得られた。また，公聴会後の審査会においても，本研究の専門分野との関連性や将来展望などの諮問がなされ，著者から適切な回答が得られた。

以上より，論文審査及び最終試験の結果に基づき，本学位論文審査委員会において慎重に審査した結果，本論文が博士（工学）の学位に十分値するものであると判断した。