TRACKING A TENNIS BALL
USING IMAGE PROCESSING TECHNIQUES

A Thesis Submitted to the College of
Graduate Studies and Research
In Partial Fulfillment of the Requirements
For the Degree of Master of Science
In the Department of Computer Science
University of Saskatchewan
Saskatoon

By

JINZI MAO

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Master degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis. Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
University of Saskatchewan
Saskatoon, Saskatchewan (S7N 5C9)

ABSTRACT

In this thesis we explore several algorithms for automatic real-time tracking of a tennis ball. We first investigate the use of background subtraction with color/shape recognition for fast tracking of the tennis ball. We then compare our solution with a cascade of boosted Haar classifiers [68] in a simulated environment to estimate the accuracy and ideal processing speeds. The results show that background subtraction techniques were not only faster but also more accurate than Haar classifiers. Following these promising results, we extend the background subtraction and develop other three improved techniques. These techniques use more accurate background models, more reliable and stringent criteria. They allow us to track the tennis ball in a real tennis environment with cameras having higher resolutions and frame rates.

We tested our techniques with a large number of real tennis videos. In the indoors environment, We achieved a true positive rate of about 90%, a false alarm rate of less than 2%, and a tracking speed of about 20 fps. For the outdoors environment, the performance of our techniques is not as good as the indoors cases due to the complexity and instability of the outdoors environment. The problem can be solved by resetting our system such that the camera focuses mainly on the tennis ball. Therefore, the influence of the external factors is minimized.

Despite the existing limitations, our techniques are able to track a tennis ball with very high accuracy and fast speed which can not be achieved by most tracking techniques currently available. We are confident that the motion information generated from our techniques is reliable and accurate. Giving this promising result, we believe some real-world applications can be constructed.

ACKNOWLEDGMENTS

Thanks for the help given by my supervisors: Dr. Sriram Subramanian and Dr. David Mould and volunteers who participated in the evaluation.

TABLE OF CONTENTS

page

LIST OF TABLES

LIST OF FIGURES

Figure page

CHAPTER 1

INTRODUCTION

The advent of high-speed computers and high-resolution/frame rate cameras has renewed research interest in computer vision algorithms for applications such as security, military, and sports. Within these applications, one of the most frequently performed tasks is determining the number, position, and movement of various objects. The motion analysis and tracking information provided by computer vision techniques can be used for purposes such as tactical analysis in different sports, including baseball [26], soccer [2], American football [3], and tennis [64]. Motion information also provides better comprehension of the game for resolving doubtful referee decisions. Examples include an automatic line-call system QUESTEC [53], a system named *Hawk-Eye* [29] which helps the commentary team to analyze the play, and an automatic off-side detection system [39]. In addition, some training systems, such as the system introduced by Theobalt et al. [65] can also be constructed with this information.

1.1 TRACKING A TENNIS BALL

Detecting and tracking objects during a sporting event is an active research area. There have been successful attempts such as a soccer ball tracking system [16] based on a modified Hough transform technique, and a baseball tracking system [65] using color-based region detection technology. However, there are not many reports of successful deployment of such systems in real-world tennis applications. Tracking the ball in a tennis game poses many challenges due to the ball's small size and high speed [52]. Tennis ball tracking enables virtual replays, new game statistics, and other visualizations which result in novel ways of experiencing and analyzing tennis matches. Several other applications, including computer-assisted refereeing and player training, can also benefit from real-time tracking of the tennis ball.

Tracking the tennis ball is challenging due to the small size of the ball (67 mm in diameter), the relatively large size of the court (the diagonal length of the court is over 26 m), the high speeds at which it travels (the fastest serves are over 225 km/h), changing lighting conditions (especially in an outdoors environment), and varying contrast between the ball and the background across the scene.

Object-based and non-object based detection and tracking algorithms, such as the circular Hough transform [16], the *closed-world* tracking system introduced by Intille et al. [32], and the *self-windowing* technique introduced by Ishii et al [33], are not suitable for this application. In particular, the objects that these systems work with usually move much slower than the tennis ball. In addition, images of fast moving objects are normally blurred, which makes object recognition more difficult. The objects they track are much larger than the tennis ball. In addition, most cameras used by these techniques have low resolutions and frame rates. The fast processing speed achieved by these methods depends on the small size and number of images. If high resolution/frame rate cameras are selected in order to have a better view of the tennis ball, their processing speed will be strongly affected. Recently only a few successful deployments of tennis ball tracking systems, such as Sudhir et al. [64] and Pingali et al. [52], were reported in real-world applications. However, they did not provide systematic analysis of performance and errors in a real-world environment. In addition, player-ball interaction and occlusion of the ball due to players were not addressed by these techniques.

In this thesis we present the investigations into various computer vision algorithms for tracking the tennis ball. This system is designed to satisfy the following requirements:

Fast processing speed. This system needs to execute efficiently, i.e., the speed of image processing should catch up with (or be close to) the speed of the video input. This is challenging if a high resolution, high frame rate camera is selected. For example, if we use a color camera with resolution 640x480 and a frame rate of 50 fps, the amount of data that needs to be processed is about 44 MB/sec. To process such a huge rate of image data quickly, an efficient tracking algorithm is required.

Compatibility with various visibility levels. The visibility of the tennis ball among all frames varies due to several reasons:

- High speed of the tennis ball. The speed of the tennis ball during a match can easily reach 225 km/h: as a result, the contour of the tennis ball appears very blurred in some video frames.
- Environment variations. Due to the variations in the environment, such as the lighting, contrast and shadow, the shape and gray level of the tennis ball are very unstable across video frames.
- Presence of other moving objects. There are also other moving objects which look similar to the target objects, such as certain commercial symbols, which create ambiguity when we try to discriminate the ball from all candidates.
- The image area of the tennis ball ranges from 5 to 60 pixels due to different distance to the camera. As a result, feature information contained in the tennis ball, such as color, shape, and size, varies depending on the image size of the tennis ball.

To compromise with various visibility levels, a robust object recognition technique is required.

Proper handling of occlusion. Tennis balls are often occluded by other objects, such as the tennis players and tennis rackets. When occlusion happens, the system state needs to be updated properly.

1.2 OUR SOLUTION

In trying to provide a system that complies with the previous requirements, we first adapted a background subtraction technique to track the tennis ball. Our technique works in three stages. First, a background model is created. Next, current images are compared with the background model. The differences between them represent foreground objects. Based on either the color or the shape, objects which look similar to the tennis ball are considered as possible occurrences of the tennis ball. After verifying against the predefined size and aspect ratio, the location of the tennis ball is finally determined. To reduce noise, we use edge images instead of actual images to create the background model and find foreground objects. Noise is usually not as prominent in the edge images, as noise edges are very weak.

We then compared our approach to a technique based on Haar classifiers. The latter technique uses a cascade of boosted classifiers trained with Haar-like features to detect the target objects. First, a large number of positive and negative images are collected. The images are then fed into a classifier trainer which creates a cascade of boosted classifiers. With these classifiers, an object can be detected by sliding a search window through the image and deciding whether that search window contains the tennis ball. The technique of Haar classifiers is a general object detection algorithm, and by making the comparison we hope to identify positive and negative aspects of our background subtraction solutions.

To focus on the feasibility of background subtraction technique, we tested our solution in a simple controlled environment where a person tossed a tennis ball in front of a camera. Our analysis shows that background subtraction techniques were not only faster but also more accurate than Haar classifiers for the purpose of tracking the tennis ball. The result leads us to believe that background subtraction is a feasible solution for our problem. However, to achieve desirable performance, our technique requires very stringent conditions, such as proper lighting, the slow moving tennis ball, and a short distance between the ball and camera. These requirements can not be met in real-world applications. Also, the processing speed is not fast enough to handle high resolution/frame rate cameras: the maximum achievable processing speed was only 8 fps. To overcome these limitations, we further improved our solution and developed three improved techniques:

Background subtraction with verification. We first create the background model by averaging a number of background images. Then the differences between the current and the average background image are calculated. All the negative results are discarded. The idea behind this is that the gray value of the tennis ball is usually higher than that of the dark background due to the wavelength of tennis ball's color matches to the peak sensitivity of our cameras. The differences between the area covered by the tennis ball and the same area in the background image are positive. The area of the player is also detected. Candidates inside the area of the player are removed. The remaining candidates are further verified based on different criteria.

Image differencing between the current and the previous images. Ball candidates from the current image are first detected by image differencing between the current and the previous images. However, this operation results in a problem called double detection: positions of the ball appearing in both the previous and the current images are detected. To eliminate the candidates originating from the previous image, we first subtract the background image from the current image. A logical AND operation is then performed between the result of background subtraction and image differencing. The region of the ball generated from the previous frames is removed. Instead of using the simple average image, this technique uses a different background model which simulates the value of each background pixel as a single Gaussian distribution. This background model is more accurate than the simple average image. Once the ball candidates in the current image are found, the region of the player is detected. Ball candidates within the region of the player are removed. After verifying with shape and dynamics information, the tennis ball is detected.

Adaptive background modeling using a mixture of Gaussians. This technique is similar to the first technique but it uses a more sophisticated and accurate background model. Instead of using a single Gaussian, this technique uses a mixture of Gaussians to model the value of the each background pixel. In addition, changes to the environment can be gradually adopted into the existing background model which makes this technique more dynamic and flexible.

The main differences in these three techniques are the background model they use and the way foreground objects are detected.

To increase the processing speed of our techniques, we developed an optimization strategy which is based on the Kalman filter [35]. The Kalman filter is an algorithm which provides an efficient computational means to estimate the states of a process. This filter is very powerful in several aspects: it supports estimations of past, present, and even future states. Based on the estimation generated from the Kalman filter, object detection is performed only in the predicted region instead of the whole image. The amount of image data needed to be processed is reduced and as a result, the processing speed is increased.

We evaluated the performance of our improved techniques on a real tennis scene captured with a high resolution and high frame rate camera. The performance is analyzed based on three figures: true positive rate, false alarm rate, and image processing speed. The true positive rate is the rate of positive responses in the presence of instances of the feature. The false alarm rate is the rate of positive responses in the absence of the feature. Image processing time is measured as the frame rate.

To test the robustness of our techniques, more than sixty videos were recorded in both an indoors and an outdoors court. The results show that the overall performance of our techniques is improved compared with our first attempt. For the indoors environment, we achieved about a 90% true positive rate, less than a 2% false alarm rate, and a speed of 20fps on average. Our original approach could only achieve about a 20% true positive rate, a 65% false alarm rate, and a speed of 10fps on average. Due to the complexity and instability of the outdoors environment, the performance of our techniques was not as good as the indoors case. The average true positive rate was less than 40% and average false alarm rate was greater than 25%. Our original approach achieved about a 12% true positive rate, and a 70% false alarm rate. The processing speed of all techniques in the outdoors environment was slightly less than that of the indoors case.

Based on the analysis, we found the performance of our techniques is strongly affected by the robustness of the background model. As a result, the technique using a more robust background model achieves higher detection accuracy. In addition, the test results are also affected by external factors such as the stability of the background environment and the image size of the tennis ball. As a result, the outdoor results were worse than indoors because of the instability of the outdoors environment due to various reasons such as clouds, wind, and trees.

The rest of this thesis is organized as follows: Chapter 2 reviews previous work. Chapter 3 describes the technique of background subtraction with color/shape recognition. Chapter 4 describes the improved background subtraction techniques. Chapter 5 describes the results of the evaluation with real-scenes and discusses its implications for future investigations. Finally, conclusions and future work are presented in chapter 6.

CHAPTER 2

STATE OF THE ART

A large number of object detection and tracking techniques have been developed in the last two decades. They can be roughly classified into two categories: object based and non-object based. Object based approaches are suitable for detecting and tracking objects that are different from others in the video frames, i.e., the target objects can be recognized easily. Most of these techniques involve creating a model representing the target object, retrieving object properties, and discriminating target objects. If the properties used for object detection and tracking are not from the object itself, non-object based approaches can be employed.

Object-based approaches can be further divided into three classes: feature based, model based, and motion based. In feature based algorithms, the object feature states, such as the shape and the color, are used to differentiate target objects from others in a frame. Model based algorithms use high-level semantic representations and domain knowledge to differentiate target objects from other objects. Motion based algorithms analyze the sequence of video frames and extract and interpret motion consistencies over frames in order to discriminate moving objects.

If there is virtually no property available to distinguish the target objects from others, the detection and tracking is called an object undistinguishable problem. To solve this type of problem, a number of non-object based algorithms have been developed. This type of algorithm does not evaluate whether a sole object is the target object; instead, it evaluates some other properties, such as context information which includes the information on location, time, people's activity, and surrounding environment.

The rest of this section is organized as follows: Section 2.1 explains some image processing techniques used in this thesis. Section 2.2 describes the feature-based approaches. Section 2.3 describes the model-based approaches. Section 2.4 describes the motion-based approaches. Section 2.5 describes the non-object based approaches.

2.1 IMAGE PROCESSING TECHNIQUES

Throughout this thesis, a number of image processing techniques are mentioned. They serve different purposes such as image smoothing, edge detection, and contour detection. In the following section, these techniques are briefly described.

2.1.1 MEDIAN FILTER

The Median filter, as described by Boyle [8], is a commonly used technique for reducing small noise in an image. Small noise normally appears very distinct and its gray value is quite different from its neighbors. This technique eliminates the noise by changing its gray value to the median of neighboring pixel values. An example is given in Figure 2.1: the values of the neighboring pixels are 115, 119, 120, 123, 124, 125, 126, 127, and 150. The median value is 124. The value of the current pixel is replaced with 124. In this example a 3x3 neighborhood is selected. Increasing the size of neighborhoods will generate more severe smoothing.

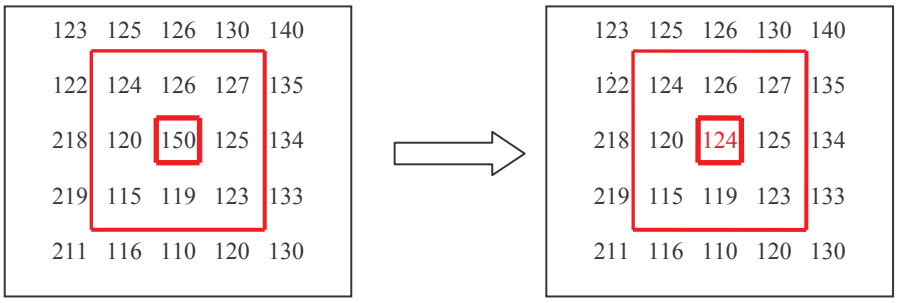


Figure 2.1: An example of the Median filtering. The value of the current pixel is replaced by the median value in its neighbors.

The computation cost of the Median filter is relatively high since it needs to sort all the values in the neighborhood.

2.1.2 LAPLACE OPERATOR

Gradient images illustrate the changes in gray value. They are often used for edge detection and can be generated by some gradient generators such as the Laplace operator [23]. The Laplace operator generates the gradient magnitude image using four convolution masks which approximate the second derivative. These masks are:

$$h=\begin{bmatrix}0&1&0\\1&4&1\\0&1&0\end{bmatrix}\quad h=\begin{bmatrix}1&1&1\\1&-8&1\\1&1&1\end{bmatrix}\quad h=\begin{bmatrix}2&-1&2\\-1&-4&-1\\2&-1&2\end{bmatrix}\quad h=\begin{bmatrix}-1&2&-1\\2&-4&2\\-1&2&-1\end{bmatrix}$$

To create a gradient magnitude image, different mask is applied to an input image and the value of the current pixel is replaced by the result generated from a convolution operation. One example is given in Figure 2.2: the value of the current pixel is changed from 100 to 179, i.e. $1\times115+1\times119+1\times120+1\times123+1\times124+1\times125+1\times126+1\times127-8\times100=179$.

$$h=\begin{bmatrix}1&1&1\\1&-8&1\\1&1&1\end{bmatrix}$$

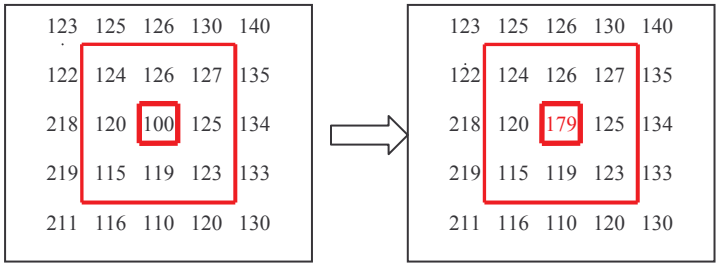


Figure 2.2: An example of applying the Laplace operator. The value of the current pixel, i.e. 100, is replaced by 179 generated from the convolution operation.

2.1.3 MORPHOLOGICAL OPERATORS

Dilation $\oplus$ is one of the two basic morphological operators. It combines two set of vectors using vector addition. Figure 2.3 gives an example of the dilation operation. The effect of the dilation operation is to expand the boundaries of foreground objects. As a result, areas of foreground objects are increased and holes within those objects become smaller.

$A=\{(1,0),(1,1),(1,2)\}$

$B=\{(-1,-1),(-1,0),(-1,1),(0,-1),(0,0),(0,1),(1,-1),(1,0),(1,1)\}$

$A\oplus B=\{(0,-1),(0,0),(0,1),(0,2),(0,3),(1,-1),(1,0),(1,1),(1,2),(1,3),(2,-1),(2,0),(2,1),(2,2),(2,3)\}$

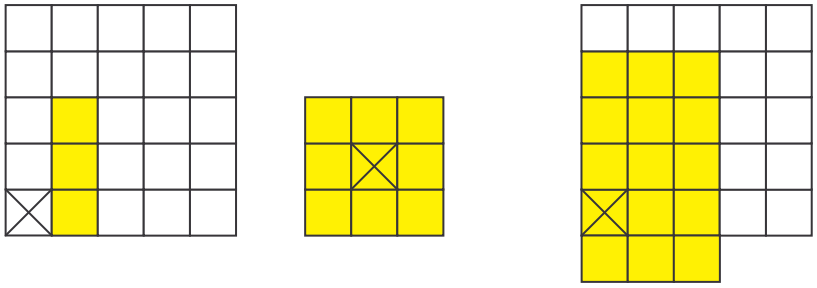


Figure 2.3: An example of the dilation operation.

As opposed to the dilation, the erosion Θ is another morphological operator which combines two sets using vector subtraction. The result of this operator is to wear away the boundaries of foreground objects. Thus areas of foreground objects are reduced, and holes within those areas become larger. Figure 2.4 gives an example of the erosion operation.

$$A = \{((1,0),(1,1),(1,2),(1,3)(2,0),(2,1),(2,2),(2,3),(3,0),(3,1),(3,3)\}$$
$$B = \{(-1,-1),(-1,0),(-1,1),(0,-1),(0,0),(0,1),(1,-1),(1,0),(1,1)\}$$
$$A\Theta B = \{(1,1),(1,2)\}$$

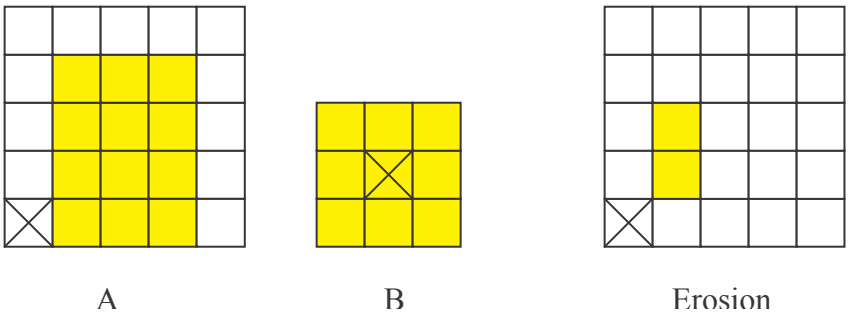


Figure 2.4: An example of the erosion operation.

2.1.3 CANNY EDGE DETECTOR

Canny edge detector introduced by Canny [11] is a well-known technique for edge detection. It works as follows: first, an image is smoothed to remove small noise. Based on this image, two gradient images are generated on both vertical and horizontal directions using one of the gradient operators such as the Laplace operator. An edge magnitude and direction images are calculated from these two gradient images. After thresholding the edge magnitude image, the edge breadth is reduced by a non-maxima operation which removes edges with a magnitude less than their neighbors. The result is

then thresholded by two thresholds: T1 and T2 where T1<T2. Edges with a magnitude less than T1 are removed, while those greater than T2 are detected as real edges. Finally, edges with a magnitude between T1 and T2 are also detected as edges if they connect to an edge pixel. The details of Canny edge detector are given in Figure 2.5.

1. An input image is first smoothed to remove small noise.
2. Two gradient images on both vertical and horizontal directions are created using one of the gradient operators. The results are denoted as $G_x(m,n)$ and $G_y(m,n)$ where m and n are the pixel coordinate.
3. An edge magnitude image is created as:

   $$M(m,n) = \sqrt{G_x(m,n)^2 + G_y(m,n)^2} \ . \qquad (2.1)$$

   An edge direction image is created as:

   $$\theta(m,n) = \tan^{-1}\left|\frac{G_y(m,n)}{G_x(m,n)}\right| . \qquad (2.2)$$

4. Threshold the magnitude image $M(m,n)$. set pixels to 0 if their value below a predefined threshold. The result is denoted as $M_T(m,n)$.
5. To reduce the edge breadth, a non-maxima operation is performed on $M_T(m,n)$: the non-zero pixels in $M_T(m,n)$ are set to 0 if their values are not greater than their neighbors along the direction indicated by $\theta(m,n)$.
6. Threshold the previous result by two thresholds: T1 and T2 where T1<T2. Pixels with values greater than T2 are detected as edges. Pixels with values less than T1 are detected as non-edges. The result is then processed again: pixels with values between T1 and T2 are also detected as edges if they connect to an edge pixel.

Figure 2.5: Canny edge detection

2.1.4 SHAPE DESCRIPTION

The shape of objects can be represented by various features, such as the aspect ratio, compactness, and roughness.

The aspect ratio is the ratio between the width and height of a blob. For circular objects such as the tennis ball, the difference between its height and width is very small. Therefore, the aspect ratio of the ball is close to one.

The compactness measures how close all particles in a blob are from one another. It is measured as the ratio of the perimeter to the area of the blob. A circular blob is the most compact and is defined to have a compactness of 1.0 (the minimum). More convoluted shapes have larger values.

The roughness measures how smooth a blob's surface is. It is a ratio of the perimeter to the convex perimeter of the blob. The convex perimeter is the perimeter of the smallest convex that encloses the blob. Smooth convex blobs have a roughness of 1.0 while rough blobs have a higher value as the blob's perimeter is larger than their convex perimeter.

2.2 FEATURE BASED APPROACHES

Feature based approaches use information such as shape and color distribution to determine the target object from image frames. Four examples of this approach are given in this section. These examples were chosen such that each demonstrates one possible usage of typical feature information, such as object shape, gray value, and color.

2.2.1 HOUGH TRANSFORM

The Hough transform [17] technique detects objects whose shapes can be parameterized in a Hough parameter space. Such objects include straight lines, polynomials, and circles, etc. The peaks detected in the Hough parameter space is used to describe the object shapes.

To illustrates the concept of the Hough transform, an example of line detection is given as follows: line segments can be described using parametric notion:

$$x\cos\theta + y\sin\theta = r,$$

where r is the length of a normal from the origin to this line and θ is the orientation of r with respect to the X-axis, as illustrated in Figure 2.6. For all points on the line, the value of r and θ is constant.

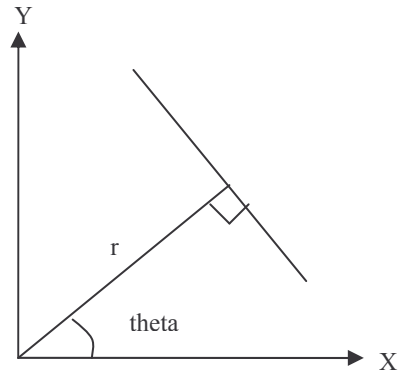


Figure 2.6: A normal representation of a straight line

Given this normal representation, we transform the points on the line to curves in a Hough parameter space whose coordinates represent the normal length and orientation. In this Hough parameter space, points which are on the line generate curves intersecting at a common point (r,θ).

The same idea can be applied to circle detection. The parametric representation of a circle can be defined as:

$$(x-a)^2+(y-b)^2=r^2, \tag{2.3}$$

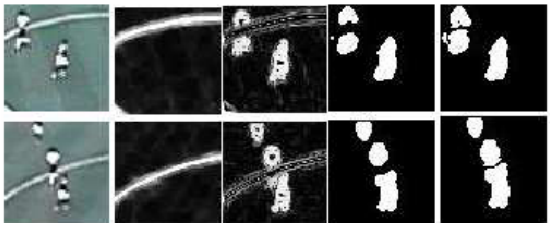
where (a,b) is the circle center and r is the radius. For all points on the circle the value of (a,b,r) is the same. We transform points on the circle to 3D curves in a Hough parameter space whose coordinates represent the circle center and the radius. In this Hough parameter space, points which are on the circle generate curves intersecting at a common point (a,b,r).

Given the circular shape of the tennis ball, it seems possible to adopt the Hough Transform in our solution. However, a tennis ball does not always appear a circular shape due to environment variations. For instance, a tennis ball might appear a crescent moon shape because of the self-shadowing effect created by lighting. Furthermore, this circle detection technique is implemented using a three-dimensional array which represents the center and radius of all possible circles. Therefore its processing speed is $O(n^3)$ where n is the total number of pixels in the image and is too slow for real-time object tracking. To adopt this technique in our solution, further modification and optimization are required.

2.2.2 TRACKING MULTIPLE SOCCER PLAYERS

Müller and Anido [46] presented multiple a technique which is able to track multiple soccer players in real-time. In this technique, object detection is only performed in a portion of the entire image. As a result, the amount of data to be processed is reduced and the processing speed is improved.

An example of player detection is illustrated in Figure 2.7. This technique uses gradient reference frames to detect players. The gradient reference frame is a gradient image of the field when there are no objects present. An example of such an image is shown in (b). A number of gradient reference frames are produced which together cover the entire field. For each queried image, for example (a), its gradient image is first calculated, as shown in (c). Then this gradient image is subtracted by the gradient reference image corresponding to that field location. The pixels whose values are bigger than a given threshold are detected as an object. The result is shown in (d). To eliminate shadow, morphological cleaning operations are applied. The final result is shown in (e).



(a) (b) (c) (d) (e)

Figure 2.7: The process of player detection presented by Müller and Anido [46].

After obtaining each player's location, the next step is to determine which one is the tracked player. It is done by proximity: the player whose location is closer to that of the player in the previous frame is considered to be the tracked player.

The authors presented one possible solution for detecting and tracking the human body. In a tennis event, players are also part of the problem domain. We can improve our solution by taking account of player information. However, the static background model used by this technique is not very robust because it is not compatible with environment variations, such as lighting and contrast changes. It is possible to modify this technique using a more reliable and robust background model. Furthermore, the idea of dividing the entire image into sub-images and processing only a portion of the entire image each time is a good optimization strategy to increase the processing speed.

2.2.3 TRACKING HIGH-SPEED MOTION WITH MULTI-EXPOSURE IMAGES

Theobalt et al. [65] described a technique for tracking high-speed motion sequences using low-cost digital cameras and strobe lights. In this paper, the authors applied their technique to baseball tracking. First, a few colored markers are attached to the surface of a baseball. Using the technique of color segmentation, the projected ball markers are detected. The center of the ball is then approximated by the center of the bounding box which encloses all detected markers.

The authors demonstrated the use of color information to segment objects from the background. The yellow green color of the tennis ball has very good visibility. Tennis balls show up well on video sequence. It is probable to detect and track the tennis ball based on the color information. However, this technique was only tested in a uniform background. For a real-world environment its performance needs to be verified.

2.2.4 ADAPTIVE BACKGROUND ESTIMATION USING THE KALMAN FILTERING

Adaptive background modeling, as described by McIvor [44], is able to update the current background model by adapting changes in the illumination and motion. Therefore, the resulting background model is more robust and dynamics than static ones. However, a common problem associated with adaptive background modeling is the problem of double detection, i.e., when foreground objects move slowly or stay at the same position for a period of time, their images will be adapted into the background model. If the objects move, the regions that are occupied previously and currently by the objects are all detected as foreground objects.

To solve this problem, Ridder et al. [55] presented a solution that uses the Kalman filter [35] which is an efficient state estimation mechanism: a pixel is detected as the foreground if the difference between its measured pixel value and its estimated pixel value generated by the Kalman filter is greater than a threshold. If a pixel is considered as a foreground pixel, the contribution of its measured pixel value is small when estimating the pixel value for the next time step. On the other hand, if a pixel is

considered as a background pixel, the contribution of its measured pixel value is big. Therefore, the influence that foreground objects make to the background model is smaller and the adaptation of moving objects into background model is slower. A problem with this technique is that when foreground objects cover the background for a period of time, it is impossible to get information about the changes in the background.

The author illustrated one possible usage of the Kalman filter to estimate the background pixel value. It is also applicable to the tennis ball tracking problem. For instance, it is possible to estimate the location or gray value of the tennis ball in the image sequences based on the Kalman filter. The estimated information can help us to further improve the detection accuracy and efficiency.

2.3 MODEL BASED APPROACHES

Model based algorithms use high-level semantic representations and domain knowledge to detect objects. Object detection and tracking are then performed based on these created models. Four examples are described in this section and they demonstrate two typical ways of constructing object models based on low-level information, such as features and object color distribution. Using this idea, we can also detect the tennis ball based on an object model which systematically describes the tennis ball. However, a single model is not able to completely describe underlying objects since the same object can appear very differently under various conditions. For example, the appearance of the tennis ball varies under different lighting conditions, speed, and distance. We can create a number of models which fully describe the tennis ball in all possible circumstances.

2.3.1 OBJECT DETECTION USING A CASCADE OF BOOSTED CLASSIFIERS

Viola et al. [68] introduced an object detection algorithm based on features shown in Figure 2.8. The value of these features can be computed very efficiently using an integral image. The value of integral image at any location (x, y) is the sum of pixels above and to the left of (x, y) inclusively in the original image.

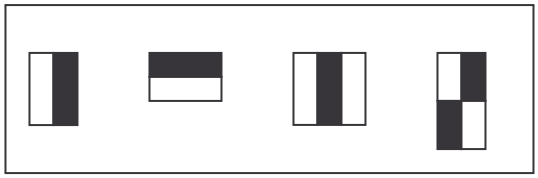


Figure 2.8: The features used by Viola et al. [68].

Each image contains enormous number of features. Even each feature can be computed quickly with the integral image, the computational cost is still too high to generate the whole set of features. The authors assumed that only a portion of the entire features are sufficient to describe the target object and developed a feature selection algorithm. During each iteration of this algorithm, a large number of weak classifiers are trained based on every feature f_j. Each weak classifier h_j is defined as:

$$h_j(x) = \begin{cases} 1 & if\ p_j f_j(x) < p_j \theta_j \\ 0 & otherwise \end{cases}, \tag{2.4}$$

where f_j is the feature associated with the current classifier, x is a sample image, θ_j is a predefined threshold, and p_j is the parity indicating the direction of the inequality sign. If the value of its corresponding feature is above the threshold, the classifier generates 1 indicating the current image x looks like the target object. If the result generated from this classifier is wrong, an error is created. The weak classifier which generates the least number of errors is then chosen for current iteration. This procedure is repeated a large number of times resulting in a final set of selected features and their corresponding weak classifiers. The details of this algorithm are given in Appendix A

To further reduce the processing time, this paper contributes the idea of classifier cascades. Instead of using a single classifier which works with all selected features, features are organized into a cascade of smaller classifiers. Each classifier works with a small portion of the features. An image is detected as the target object if it is approved by all the classifiers. A large portion of the candidates are rejected in the early stages that use the simplest features. Therefore, most of the detection time is spent on the real targets and the processing time is reduced.

Given a number of sample images, we can create our own classifiers that are able to recognize the tennis ball. Once these classifiers are available, they can be simply applied to our problem: for each image, these classifiers determine the position and the size of

the ball. However, this technique has five drawbacks: first, the performance of this technique strongly depends on the completeness of sample images used for the training. The appearance of the tennis ball varies in a real-world environment due to various lighting conditions, and moving speeds. It is difficult to collect a complete set of sample images which contains all possible appearances of the tennis ball. Second, this technique can only detect the target object with a size within a limited range. In a tennis match, the size of the tennis ball size ranges from 5 to more than 60 pixels. This technique is not able to detect the tennis ball with all possible sizes. Third, when the image size of the ball is too small, these classifiers are not reliable because the ball might look no different to other small objects. Four, training classifiers requires a very long time. Retraining does not gain any benefit from previously created classifiers. Five, the processing speed of this technique is not fast enough for real-time object tracking. Therefore, we think this technique is not very robust for tennis tracking. To adapt this technique in our solution, further improvement is required.

2.3.2 AN EXTENDED SET OF HAAR-LIKE FEATURES FOR RAPID OBJECT DETECTION

Viola et al. [68] introduced a rapid object detection scheme based on a boosted cascade of Haar-like features. Lienhart and Maydt [40] improved this technique by adding a set of rotated Haar-like features as shown in Figure 2.9 which dramatically enrich the set of simple features.

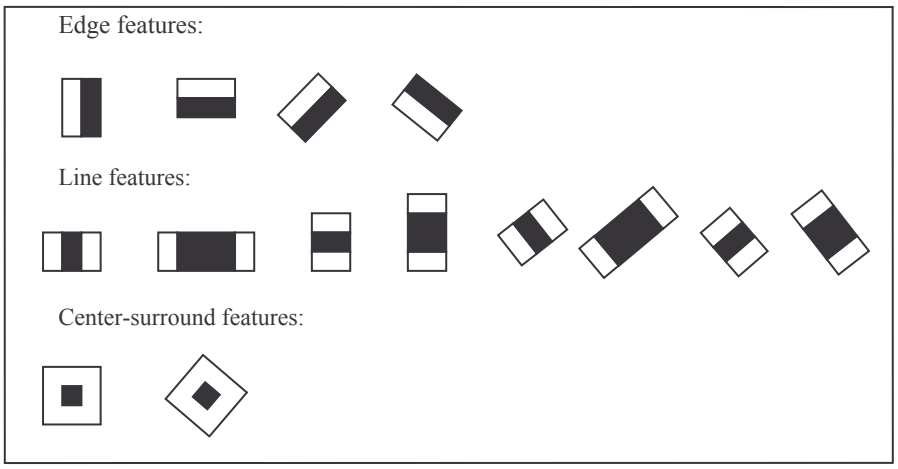


Figure 2.9: The features used by Lienhart and Maydt [40].

At a given hit rate, this technique shows on average a 10% lower false alarm rate compared with the technique introduced by Viola et al. [68]. This extended set of Haar-like features can substantially improve detection accuracy and reduce false alarm of our tennis ball classifiers.

2.3.3 CAMSHIFT

CamShift [72] stands for the "Continuously Adaptive Mean Shift" algorithm. This technique is able to track objects based their color.

First, the Mean Shift algorithm was developed which detects the centre and size of the object in an image. The detail of this algorithm is illustrated in Figure 2.10:

1. Set the size of the search window.
2. Set the initial location of the search window.
3. Compute the location of the centroid within the search window based on the 0th and first moment.
4. Center the search window at the centroid.
5. Repeat Steps 3 and 4 until it has moved for a distance less than the preset threshold.

Figure 2.10: The Mean Shift algorithm.

Based on the Mean Shift algorithm, the CamShift algorithm works as follows: first, a color histogram model of the target object is created. An image is then converted to a color probability distribution image using histogram back projection which replaces the color of each pixel with the probability of that color appearing in the color histogram model. The center and size of the object are found by the Mean Shift algorithm from the color probability image. The size and location of the tracked object are propagated to the next frame. An overview of this algorithm is given in Figure 2.11.

1. Set the whole image as the calculation region of the probability distribution.
2. Select the initial location of the search window.
3. Produce a color probability distribution within the calculation region.
4. Execute the Mean Shift algorithm described above to find the centroid and size of the object.
5. For the next frame, place the search window at the centroid found in Step 4 and reset the window size based on the size found in Step 4.

Figure 2.11: The CamShift algorithm.

For fast moving objects such as a tennis ball, the performance of this algorithm is not good because objects can easily move out of the calculation region. Further this tracking algorithm depends on the object's color and is strongly affected by lighting conditions. The tennis ball appears white when it is exposed to strong lighting and gray when it is in the dark environment. In addition, the color of the tennis ball tends to adapt to the background when it is further away from the camera. Therefore, this technique is not suitable for tracking the tennis ball.

2.3.4 BACKGROUND MODELING USING A MIXTURE OF GAUSSIANS

Background subtraction is a common method to track moving objects. The major problem with this technique is that they do not effectively adapt to the environmental changes, such as changes of the lighting condition and shadow. To solve this problem, Stauffer et al. [61] introduced a background modeling technique which models the value of each background pixel as a mixture of Gaussian distributions which covers all possible appearances of the target object. Pixel values which do not fit into all the background distributions are considered as the foreground. Foreground pixels are then grouped into connected components. Finally, the connected components are tracked from frame to frame using a multiple hypothesis tracker based on the Kalman filter [35].

We can model the background of a tennis event using a mixture of Gaussians. Based on this model we can determine pixels belonging to foreground objects. Given this robust background model, we can expect more accurate detection of foreground objects.

However, this technique requires more processing time than traditional background modeling techniques. Therefore, an optimization method is required.

2.4 MOTION BASED APPROACHES

Motion based algorithms detect an object based on the motion information extracted from a sequence of images. This type of approaches does not necessarily require explicit construction of the object models. In this section, three techniques are discussed which demonstrate two typical ways to construct the motion information either based on a state estimation mechanism or currently available measurements. These examples also illustrate the use of motion information to detect objects and verify object candidates. Motion information demonstrated during a tennis event is very obvious. The ball constantly moves across the field following physical patterns. If the motion information of the tennis ball can be retrieved, we are able to improve the detection accuracy. However, to achieve the best performance, it is not sufficient solely to adopt motion based approaches. In the case of tennis tracking, the motion information contained in image sequence can be very complex and confusing because there can be many moving objects such as players, crowds, commercial symbols, and trees.

2.4.1 KALMAN FILTER

The Kalman filter [35] is a state estimation algorithm based on a feedback control mechanism: this filter predicts the process state and then obtains the feedback from the measurements. The equations for the Kalman filter are organized into two groups: time update equations and measurement update equations. The time update equations predict the current state and error covariance. The output of these equations is a state prediction for the next time step. The measurement update equations incorporate a new measurement into the prior state prediction. The output is an improved estimation. This process is illustrated in Figure 2.12.

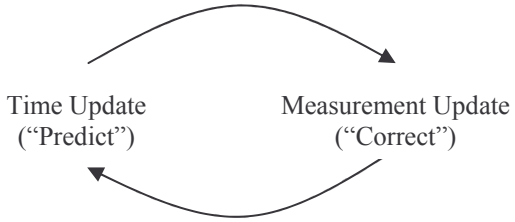


Figure 2.12: A visual illustration of the Kalman filter described by Kalman [35].

Motion information contained in video frames can be expressed as a sequence of locations, speed, and acceleration. Given the capability of predicating the system state, motion information can be constructed based on the Kalman filter. For example, we can construct a sequence of estimated locations where the object might appear in images. Dynamic information such as the speed and the acceleration can also be estimated.

Following this idea, we can apply the Kalman filter to the problem of tennis ball tracking. For instance, locations where the tennis ball might appear in the image can be predicted. The predicted information can help us to improve the detection accuracy by eliminating objects that are not located at or near the predicted location. It can also improve the efficiency of our detection algorithm by concentrating on the sub-region around the predicted location. However, the Kalman filter is not very accurate since it is based on the assumption that the process noise and the measure noise are normally distributed. In reality we often find processes with non-linear, non-Gaussian noise, multi-modal distributions. Furthermore, the quality of camera also affects the prediction accuracy of the Kalman filter. Particularly for low frame rate cameras, the time interval between two consecutive frames is too long. For fast moving objects such as tennis balls, changes in speed, acceleration, and direction can be very dramatic during such a long period of time. The Kalman filter is not fast enough to respond to constant and sudden changes of system state.

2.4.2 K- ZONE

To determine if a pitch is a strike or a ball during a baseball match, Ren et al. [54] created a system called K ZONE which is based on the trajectory information. This system consists of three parts: the camera *pan-tilt-zoom* encoding subsystem, the

measurement subsystem and the graphic overlay subsystem. The part we are interested in is the measurement subsystem which detects the trajectory of the baseball.

To track the baseball, this system exploits the kinematical properties of the baseball's flight, instead of using pattern matching. The first step is to verify the potential ball position in the image. This algorithm qualifies any potential ball positions based on the size, shape brightness and color for a number of adjacent pixels. More importantly, these adjacent pixels must be dramatically different from what occupied the same location in the previous images.

Because the background changes over time due to moving shadow and lighting variations, a mixture of Gaussian distributions centered about recently observed pixel values for a given location is used to keep track of the background. If the current pixel value deviates significantly from a predicted value, the system records that change as motion. The group of pixel locations that a ball occupies satisfies this criterion.

After applying these constraints to an input image, a number of candidates will be qualified as the ball. In order to discover the ball position, the authors developed a finite state machine with two states. In the first state, this algorithm looks for feasible trajectories. Once the trajectory is detected, all candidates are verified based on this trajectory. After the second state, all ball positions are found.

The authors demonstrated the use of motion information to resolve doubtful candidates. The same idea can be used in our solution. The authors also illustrated the use of size, shape, brightness and color information to qualify ball candidates. Given the similarity of tennis balls, these criteria are also applicable for tennis ball detection. In addition, the authors demonstrated the use of a mixture of Gaussians to model the background which makes the background model more robust and dynamic.

2.4.3 OBJECT TRACKING BASED ON TRAJECTORY PROPERTIES

Gueziec [25] presented a trajectory-based algorithm for detecting and tracking a soccer ball in broadcast videos. This algorithm is based on the trajectory of the ball and includes four components: ball size estimation, candidate detection, candidate trajectory generation and trajectory processing.

In the ball size estimation component, the ball size is estimated from that of salient objects such as the goalmouth, the ellipse and the people. The size of the ball varies from frame to frame. Its size is estimated based on these reference objects. In the candidate detection component, some non-ball objects are removed from the image. In the candidate trajectory generation component, a set of candidate feature images are created from all candidates in the sequence of frames. Each candidate feature image produces a candidate trajectory set. In the trajectory processing component, the trajectory of the ball is detected from the set of candidates. Other trajectories overlapped with that of the ball are removed. The ball trajectory and the remaining trajectories are extended using a model matching procedure. Most of the objects contained in the ball trajectory are the ball. On the other hand, most objects contained in the non-ball trajectories are not the ball.

The idea of using trajectory information to detect the ball can be very useful for tracking a tennis ball since a tennis ball is constantly moving and exhibits similar kinematic properties. If the trajectory of the tennis ball is available, we can improve the detection accuracy by eliminating objects which do not reside on the trajectory.

2.5 NON-OBJECT BASED APPROACHES

If there is virtually no property available to distinguish the target objects from others or the target objects are difficult to model, a number of non-object based algorithms can be employed. This type of algorithm evaluates some other properties such as the context information which could include the location, time, people's activity and surrounding environment. In a tennis event, the shape of the tennis ball demonstrated in video images is not very strong and varies dependent on different lighting, speed, and distance. There are some cases where the ball looks similar to other objects such as some commercial symbols. As a result, object-based techniques described previously are not sufficient to detect all occurrences of the tennis ball. For those occurrences which can not be recognized by object-based techniques, we need to investigate information other than the ball itself. The techniques described in this section give us some idea about how to use the non-object properties to detect objects indirectly.

2.5.1 CLOSED-WORLD TRACKING

Intille et al. [32] presented a technique to track weakly modeled objects (football player) using rich context information. As mentioned by the authors: a context would be something like "a region of the field near the upper hash mark on the 50 yard line that contains two players, one offensive and one defensive". The specific context can be used to determine which image processing tools can be selected.

To use context effectively, the notion of *closed-world* is presented. A *close*-world is a region of space and time in which the specific context is adequate to determine all possible objects residing in that region. Each pixel in this region belongs to either one of the objects in the current context. Given the context information, context-specific features (pixels) which represent a player are selected. These features are then matched to the next frame using correlation.

The authors exhibited the use of context information for object tracking. For small objects such as tennis balls, the context information (e.g. the players, the tennis court, the field marks, the net, the commercial symbols etc) is far richer than the ball itself. We can take advantage of this abundant information to improve detection accuracy. However, the context information is specific to a particular site. If we move to another tennis court, the context might be different and our solutions may not work properly. Therefore, we should avoid context information which is too specific.

2.5.2 SELF-WINDOWING FOR HIGH SPEED VISION

With a high-speed vision system such as the Vision Chip [34], images can be captured at extremely high speed. As a result, the difference between adjacent frames is very small. By taking account of this information, Ishii et al. [33] presented an algorithm called *self-windowing* which is dedicated to solve the image segmentation and matching problem for high-speed vision system. Its idea is quite simple: first the image of an object in the current frame is dilated. Then the image of the object in the next image can be detected as an intersection of the dilation result with the next image.

By using Vision Chip and special hardware, this technique significantly simplifies the problem of object tracking by solely using a morphological operation. In the future,

image processing techniques might evolve in this direction. However we are not able to test this method since the equipment is not available.

2.6 LIMITATIONS OF THESE TECHNIQUES

In this section, we have investigated various techniques for object detection and tracking. We conclude that each technique has its own advantages. It is possible to adopt their ideas in our solution. However, due to the challenges associated with tennis ball tracking, a single technique is not sufficient to solve the whole problem since these techniques were not originally designed to work with tennis balls. Many problems are raised when they are applied to the tennis tracking system:

- The computation costs of most techniques are very high. The processing speed for these referred techniques can not guarantee high processing speed, especially when higher resolution and faster frame rate cameras are used.
- Generally, the objects these techniques try to track do not move very fast. In the case of the tennis ball, its speed can reach 225 km/h which causes the appearance of the tennis ball to be very blurred.
- The sizes of the objects they track are much bigger than that of tennis balls. For small objects such as the tennis ball, the feature space contained in such a small region is limited and insufficient to describe the target object.

Therefore, a wiser strategy is to build a system which combines advantages offered by these techniques. Based on our analysis, we found that background subtraction is a feasible approach because it does not need explicit modeling of target objects. In addition, it is simple to implement and can operate very efficiently. However, this technique can not solve all the problems, for example the elimination of noise. A number of feature based techniques such as the Hough transform for circle detection can be used to improve detection accuracy. We can also utilize the motion and dynamic information presented by the tennis ball. In addition, the context information contained in a tennis event can also help us to eliminate suspicious candidates. One example of such the information can be described as: the distance between the positions that the

tennis ball appears in two consecutive frames is usually very far due to the fast speed of the tennis ball.

In the next chapter, we are going to present one solution which is based on the various ideas learned from this chapter. We also modified these techniques to make them suitable for our problem.

CHAPTER 3

BACKGROUND SUBTRACTION

In this chapter we present one solution for tennis ball tracking. This technique was designed to meet the requirements of high accuracy, fast processing speed, and compatibility with various lighting conditions. It is based on background subtraction and color segmentation/shape recognition. First, a background model is created. Next, current images are compared with the background model. The differences between them represent foreground objects. Based on either the color or the shape, objects which look similar to the tennis ball are considered as possible occurrences of the tennis ball. After verifying against the predefined size and aspect ratio, the location of the tennis ball is finally determined. This approach is simple to implement, and the detection accuracy is very high when certain conditions are satisfied such as slow motion and large image size of balls.

The name "background subtraction" comes from the simple technique of subtracting the observed image from the background image. As discussed by McIvor [44], it is a commonly used technique for segmenting moving objects in a scene for applications such as the surveillance and sports analysis. The area of the image plane where there is a significant difference between the observed and the background image indicates the location of moving objects. The technique is particularly appealing for real-time applications due to its simple implementation and limited computational requirements.

As we discovered: in the context of tracking a tennis ball, noise is a big issue, since the ball consists of so few pixels. Due to the bad quality of the webcam, noise appears very frequently among images, which interferes with the process of object detection. Traditional background subtraction approach is not capable of eliminating the majority of noise and they usually require additional operations. We used a modified background subtraction approach to overcome any limitations due to the bad quality of the captured images. To reduce noise, we used edge images instead of actual images to create the

background model and find foreground objects. As discussed by Xie et al. [74], noise is usually not as prominent in the edge images, as noise edges are very weak.

Our technique works as follows: first a background model is created from a collection of background images. For each queried image, edges of foreground objects are detected by background subtraction. The tennis ball is then identified by color/shape segmentation. An overview of our technique is illustrated in Figure 3.1.

The rest of this chapter is organized as follows: section 3.1 describes the process of background model creation. Section 3.2 describes the process of foreground object detection. Section 3.3 describes the color/shape segmentation. Section 3.4 analyzes the suitability of our technique. To verify the feasibility of our technique, we also implemented another technique based on boosted classifiers. The detail of this technique is described in section 3.5. Finally, section 3.6 compares the test result of our solution with that of boosted classifiers. The feasibility of our technique is analyzed.

- Given a number of background images, a background model is created which consists of an average background edge image and a standard deviation image.
- For each queried image:
    1. An edge image is created from this image.
    2. Edges of foreground objects are detected by background subtraction.
    3. Edges of the tennis ball are segmented from the result generated in the previous stage. Two approaches have been implemented. One is based on color segmentation. Another is based on shape recognition.
    4. The area of the tennis ball is then dilated and subjected to a size and aspect ratio check. The area of the tennis ball is finally detected.
    5. Finally, the location of the tennis ball is detected.

Figure 3.1: Background subtraction with color/shape segmentation.

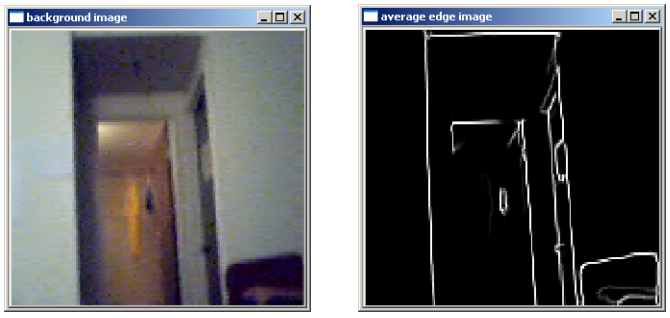
3.1 BACKGROUND MODEL CREATION

To create the background model, we first collected a number of background images. These background images were recorded in an empty court where the tennis match happens, from the same view point that the match is recorded from. Once these images are available, the background model is created as follows: first, every background image

is converted to the gray scale image. The resulting image is smoothed to remove small noise. An edge image is then created from the smoothed image. Once we have processed all background images, the background model is created which includes two components: an average background edge image as shown in Figure 3.3 (b), and a standard deviation for each pixel location. The detail of background model creation is illustrated in Figure 3.2.

- For each background image:
  1. Convert the image to a gray scale image. To convert an image from RGB color to gray scale, the following equation is used:
     Grayscale Intensity = 0.299R + 0.587G + 0.114B, (3.1)
     where 'R', 'G', and 'B' are the red, green, and blue channel respectively.
  2. Smooth the grayscale image by the Median filter.
  3. Create an edge image from the smoothed image. Edges are detected by the Canny operator. Based on our experiment, the lower and upper threshold values used by the Canny operator are set to 50 and150 respectively.
- Perform a static calculation. A background model is created which includes: ***B*** (average background edge image), and ***D*** (standard deviation image).

Figure 3.2: Background model creation.



(a) (b)

Figure 3.3: An example of the background and background edge images.

Given this background model, foreground object can be detected by background subtraction. Details are discussed in the next section.

3.2 FOREGROUND OBJECT DETECTION

Once we have the background model, edges of foreground objects can be detected as follows: from each queried image, an edge image is first created. Edges of foreground objects are then detected by background subtraction. Details of foreground object detection is illustrated in Figure 3.4.

1. For a given image, convert it to a gray scale image using the same method described in the previous section.
2. Smooth the grayscale image by using the Median filter.
3. Create an edge image ***F*** from the smoothed image. Edges are detected by the Canny operator. The parameters of the Canny operator are the same value as those in the previous section.
4. Subtract the average background edge image ***B*** from the current edge image ***F***. Label pixels as background if they meet the requirement of

$$\left|B(p) - F(p)\right| \le 2D(p), \tag{3.2}$$

where ***p*** represents each pixel location, and ***D*** is the standard deviation at position p. Those pixels not satisfying this condition are labeled as foreground objects.

Figure 3.4: Foreground object detection.

An example of foreground object detection is given in Figure 3.5: a grayscale image (b) is first created from (a). An edge image (c) is then generated from (b) by the Canny operator. Finally, edges of foreground objects are detected by background subtraction.

After this stage, edges of foreground objects are detected. Next we need to segment out edges representing the tennis ball from current result. Details are discussed in the next section.

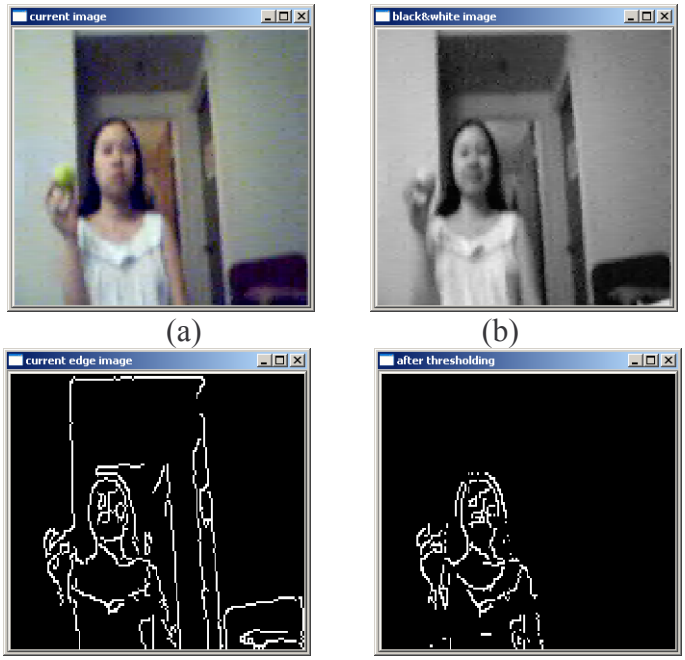


Figure 3.5: An example of foreground object detection.

3.3 OBJECT SEGMENTATION

The goal of this stage is to detect edges of the tennis ball from the result generated in the previous section. Two approaches have been taken to achieve this objective. One is based on object color. It only works with the color webcams. The other technique is based on object shape, in particular the circular shape of balls. It works for both the color webcam and the monochrome SONY camera.

3.3.1 COLOR SEGMENTATION APPROACH

This approach first removes edges whose color is significantly different from that of the tennis ball. The area of the tennis ball is dilated so that blobs can be detected. After verified with the predefined size and aspect ratio, the tennis ball is finally detected. The detail of this approach is illustrated in Figure 3.6.

An example of this approach is given in Figure 3.7: (a) after color segmentation, tennis ball edges are detected. (b) The area enclosed by the tennis ball edges is filled by the dilation. (c) After further verification, the contour representing the tennis ball, as

indicated by the red rectangle, is detected. (d) The contour of the tennis ball is superimposed on the original input image.

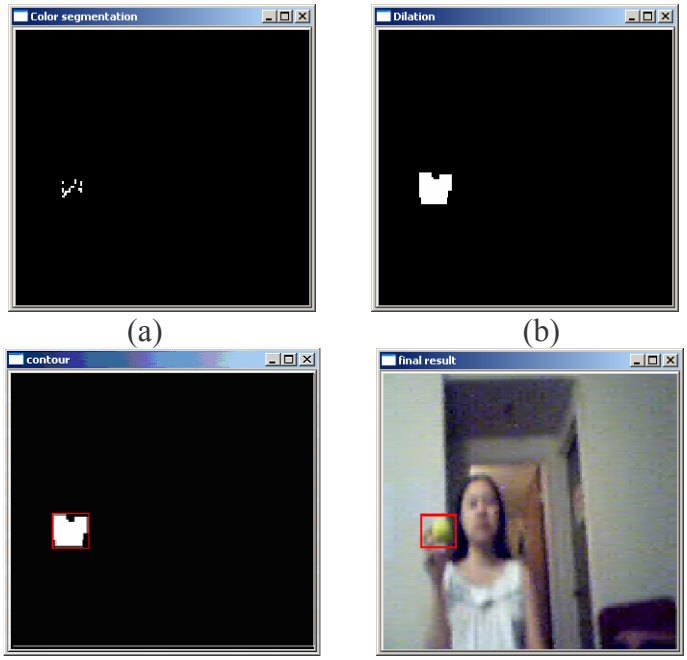
1. Eliminate foreground edges whose color is significantly different from that of the tennis ball. An edge can be classified as a tennis ball edge if the color of any one of its neighbors or itself is close to the color of the tennis balls. The typical color of the tennis ball in RGB space is (154, 205, 50).
2. Fill the area of the tennis ball by the dilation. After two or more iterations of dilation with a 3x3 mask, the area of the tennis ball enclosed by the tennis edges is filled up.
3. Contours are then detected based on the technique described by Williams and Shah [72]. The implementation of the contour detection is available in OpenCV [31].
4. The results are subject to the predefined size and aspect ratio check. Based on our experiments, we found the size of most tennis ball is between 10 and 50 pixels and the aspect ratio is between 0.7 and 1.4. Contours which do not meet these requirements are removed.
5. The remaining blob is detected as the tennis ball.

Figure 3.6: The color segmentation approach.

3.3.2 SHAPE RECOGNITION APPROACH

In most images, the tennis ball appears in a circular shape. Object detection can be achieved by shape recognition. Hough Transform [17] is an effective method which detects analytically defined shapes, such as lines and circles, and is able to recognize partial or slightly deformed shapes, therefore behaving very well in recognizing partially occluded objects.

Our technique works as follows: apply the Hough Transform for circle detection to the foreground edge images generated from the previous stage. The implementation of the Hough circle detection is available in OpenCV [30]. Subject all circular shape objects to size and/or color check. The size of the tennis ball is defined between 10 and 50. Superimpose the result to the original image.

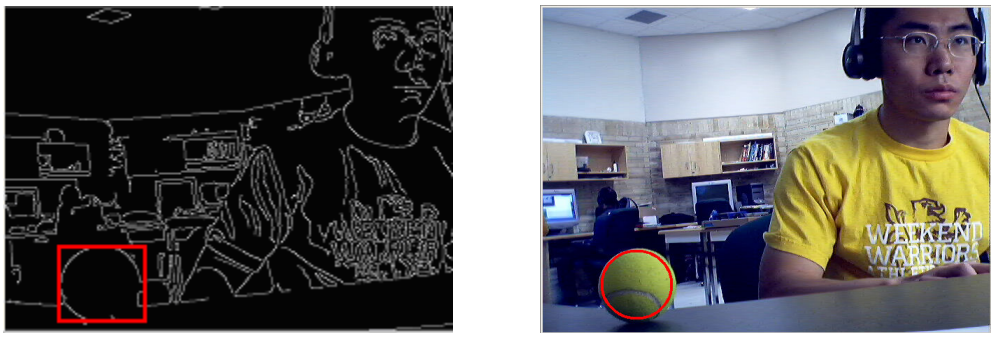


(a) (b)

(c) (d)

Figure 3.7: An example of the color segmentation approach.

An example of the shape recognition approach is illustrated in Figure 3.8. To track the tennis ball, this technique is applied to every frame in a video sequence.



(a) (b)

Figure 3.8: An example of the shape recognition approach.

3.4 SUITABILITY ANALYSIS

In order to evaluate the suitability of background subtraction approaches for tennis ball tracking we tested our techniques in a controlled setting. We also implemented a ball detection technique using a cascade of boosted classifiers described by Viola et al. [68] to compare our results against. Based on test results, we found our solutions can successfully track tennis ball under strict conditions including proper lighting, slow motion of tennis ball, and close distance between tennis ball and cameras. Background

subtraction approach is shown to be a feasible solution for tracking tennis ball. However, the conditions required by our techniques are too stringent to be satisfied in real-world applications. In addition, the algorithm's processing speed is slow when high resolution and frame rate cameras are used. Further improvement is needed in order to make them work for real applications.

3.5 BALL DETECTION USING A CASCADE OF BOOSTED CLASSIFIERS

To test the feasibility of our approaches, we also implemented another technique using a cascade of boosted classifiers introduced by Viola et al. [68] and improved by Lienhart and Maydt [40]. This technique uses feature information to detect objects. First, an integral image which evaluates features at constant time is created. Then, a small portion of features which are sufficient to distinguish objects from others is selected. Finally, a boosted classifier is constructed which further reduces computation time. Once the classifier is ready, they can be simply applied to region of interest. They respond positively if the object is contained in that region. This technique has been widely used for detecting objects such as the face, eye, and logo. The implementation of this technique is available in OpenCV [31].

This technique works in two stages: first, a boosted of classifiers is created from a large number of sample images. After a classifier is created, it can be applied to any region of interest in an input image. The classifier responds positively if the region is likely to contain the tennis ball and responds negatively otherwise. The detail of this technique is discussed in the following sections. Section 3.5.1 discusses the process of classifiers creation. Section 3.5.2 discusses the detection technique based on these classifiers.

3.5.1 CLASSIFIER CREATION

The classifier can be created in four stages: first, a large number of training images are collected. Samples of the target object are then extracted from these positive images. When the preparation is completed, the training process is started. Once the classifier is

available, its performance is tested. The detail of classifier creation is illustrated in Figure 3.9.

1. Prepare sample images for classifier training. We collected 5000 positive images and 7000 negative images. Positive images contain a tennis ball. Negative images are any arbitrary images which do not contain a tennis ball.
2. Create samples. Build a vec-file from the positive samples using the *createsamples* utility included in OpenCV [31].
3. Run the utility named *haartraining* included in OpenCV [31] to create the classifiers. The training procedure may take several days to complete even on a fast machine.
4. A classifier can be tested with the *performan*ce utility included with OpenCV [31] or directly via a "live" test if a detailed report is not necessary

Figure 3.9: Classifiers creation.

Once the classifier is ready, it can be applied to tennis ball detection. Details are discussed in the next section.

(a) (b)

Figure 3.10: An example of negative and positive images.

3.5.2 OBJECT DETECTION

After a cascade of classifiers is created, it can be applied to any region of interest (of the same size as used during the training) in an input image. The classifier outputs a "1" if the region is likely to show the object, "0" otherwise. To search for the object in the whole image the search window is moved across the image. Every location is checked using the classifier. In order to find an object of an unknown size, the scan procedure

should be done several times at different scales. An input image and the detection result are shown in Figure 3.11.

(a) (b)

Figure 3.11: Detecting the tennis ball using the classifiers.

3.6 OPTIMIZATION

Further optimization can be made by invoking the Kalman filter [35]. It is used to predict the location of the tennis ball where it might appear in the next frame. The object detection is then performed in the predicted regions instead of the entire image. Therefore, the amount of image data needed to be processed is reduced. Consequently the processing speed is increased. The optimized tracking algorithm is illustrated as Figure 3.12. This optimization strategy is used for both the background subtraction as well as the Haar classifier techniques.

3.6 TEST RESULTS

To compare the solutions in a controlled environment, a person tossed the ball at different speeds in front of a webcam under different lighting conditions. The lighting conditions were indoors with diffused day-light, outdoors in day-light and indoors with no diffused daylight.

For each experimental condition, we recorded a video with a length of 5minutes (about 480 frames per minute, resolution of 640x480). We used detection rate and the processing speed to compare the performance of the different algorithms under different conditions. The detection rate is the ratio of correct detections to the total number of

frames. The processing speed is measured by the average amount of time spent in processing each frame.

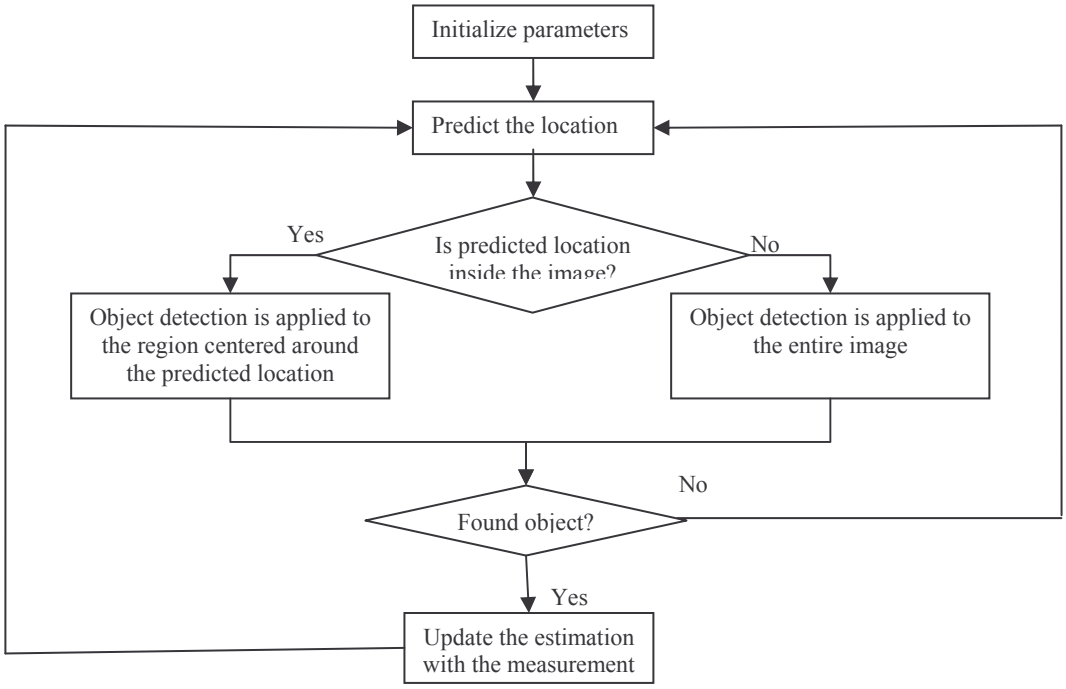


Figure 3.12: The optimization strategy.

We went through the processed image sequence for each condition and counted the number of correct detections. Table 3.1 summarizes the detection rate. Table 3.2 summarizes the average processing time for each frame. All these test results were achieved under strict conditions, such as proper lighting and slow motion.

Table 3.1: Test results using the three different approaches

	Indoors – day	Outdoors – day	Indoors - night
Background subtraction With color segmentation	90%	92%	85%
Background subtraction With shape recognition	89%	87%	82%
A cascade of boosted classifiers	84%	81%	82%

Table 3.2: Average processing time for each frame

	Background subtraction with color segmentation	Background subtraction with shape recognition	Cascade of boosted classifiers
Average processing time	0.125s	0.213s	0.317s

Overall background subtraction techniques were more accurate than the Haar classifier.

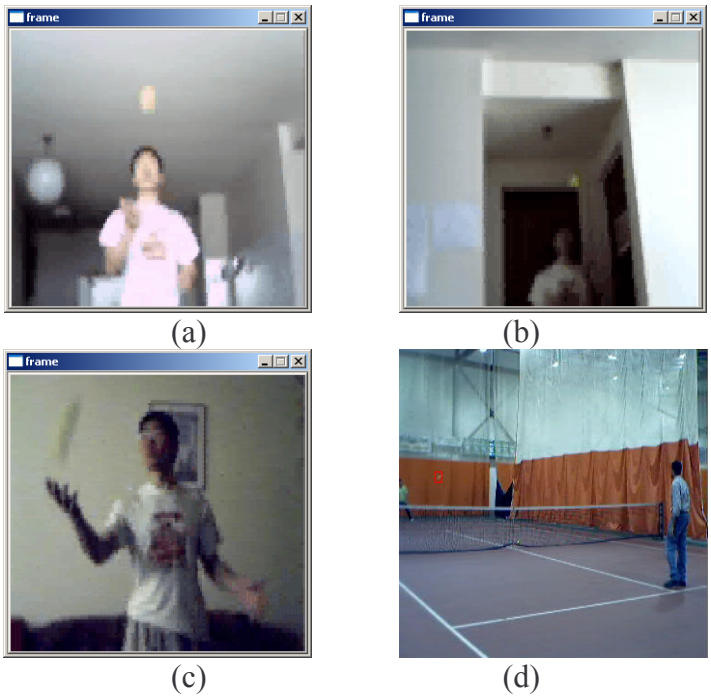
Haar Classifier

The Haar classifier could successfully detect about 84% of the time, vs. about 90% success with the background subtraction techniques. The background subtraction techniques were also faster than the Haar-classifier. The problem with the boosted classifier approach is that it can only detect objects with a limited range of sizes. These ranges need to be pre-coded in the classifier at the training stages. Building/rebuilding the boosted classifiers is very time-consuming. In our case, it takes about a whole week to create such classifiers.

Background subtraction

Background subtraction approaches are strongly dependent on the color and shape of the tennis ball. The color and shape are the crucial components of the image. They are seriously distorted if any of the following conditions are not met. First, proper lighting is required. Second, the tennis ball can not move too fast. Third, the tennis ball can not appear too far away from the camera. As shown in Figure 3.13 (a) and (b), if the lighting is too strong or dark, the color of the tennis ball appears different from its actual color. The shape of the tennis ball is also hard to recognize under dark conditions. When the tennis ball moves, its image becomes blurred, see Figure 3.13 (c). Its shape and color are badly distorted. The distance between the camera and the tennis ball also affects the performance of these solutions because the tennis ball is hard to recognize when it is further away, see Figure 3.13 (d). To preserve the color and shape, we require that the tennis ball be clearly visible and properly lit. These requirements are too stringent to be

satisfied in a real-world environment. In addition, the technique of color segmentation can not distinguish non-target objects which have similar color and size to the tennis ball. A further problem is that the shape of the tennis ball is changed if it contacts other objects, e.g. hands or a tennis racket. These problems are more evident due to the limited capabilities of webcams. We expect that when using cameras with higher resolution and frame rates the problems with visibility and shape distortion would be reduced.



(a) (b)

(c) (d)

Figure 3.13: Bad visibility due to poor lighting, the motion, and far distance.

The Kalman filter

The low frame rate of the webcam also affects the performance of the Kalman filter. Because the time interval between two consecutive frames is too long, there is a good chance that the tennis ball changes its direction, i.e., the uncertainty is great. We first tested the Kalman filter by moving the tennis ball smoothly by hand. The experiment results show that the correct prediction rate is 36%. But if we use the video recorded from a real tennis match, the correct predict rate is only 18% because the tennis ball changes its direction and speed constantly. Therefore, most of the time, the image detection is still performed to the entire image. The gain from prediction is not significant. In addition, the Kalman filter is not accurate enough to predict system states since it is based on the assumption that the process noise and measure noise are normally

distributed. The model used for detecting the ball did not take into account any drag forces that influence the ball dynamics. This also limits the ability of the Kalman filter to make good predictions.

Processing Speed

When using background subtraction with color segmentation the average processing time for about 0.125s which is about 8 frames per second (fps). This was equal to the frame rate of the recorded video. However, in the case of Haar classifiers the average processing speed translates to a frame rate of about 3.2 fps.

These speeds are not sufficient for tracking the ball in a real application. To be able to see the tennis ball in a real application we need a higher resolution camera with higher frame rates. The results suggest that for such demanding applications background subtraction techniques offer the most promise.

3.7 CONCLUSION

Overall, the background subtraction based solutions perform better than the technique of boosted classifiers. This technique is shown to be a feasible solution for tracking the tennis ball. Despite the limitations associated with our techniques, we explored several important design aspects. These techniques are a good starting point, and serve as system prototypes. In the next chapter, we will introduce the extended background subtraction techniques which overcome most of the problems described above. By using an industrial camera, the extended techniques can achieve very high detection rate with low processing time.

CHAPTER 4

EXTENDED BACKGROUND SUBTRACTION

As discussed in the last chapter, background subtraction is a feasible solution for tracking fast moving objects. However, there are still some problems associated with our first approach such as the requirements of proper lighting, and a slow moving tennis ball. It is not practical to adopt our solution in real-world applications. Therefore, we extended our first approach and developed three other techniques which were designed to track the tennis ball with higher accuracy, faster processing speed, as well as being compatible with various conditions. In addition, we tried to reduce the influence of external factors, such as lighting conditions, moving speed, and the image size of the tennis ball. These techniques are: background subtraction with verification; image differencing between the current and previous frames; and background modeling using a mixture of Gaussian distributions. A review of our extended solutions is presented in Figure 4.1. The main differences in these three methods are the background model they use and the way foreground objects are detected.

The rest of this chapter is organized as follows: section 4.1 reviews the camera details. Section 4.2 describes the technique of background subtraction with verification. Section 4.3 describes the technique of image differencing between the current and previous frames. Section 4.4 describes the technique of background modeling using a mixture of Gaussians.

- First, a background model is created from a number of background images.
- Then, foreground objects are detected.
- The player is also detected. Candidates in the area of the player are removed.
- Finally, candidates are subjected to the shape and dynamic verifications.
- The remaining candidates are detected as the tennis ball.

Figure 4.1: A review of our extended solutions.

4.1 CAMERA DETAILS

To obtain better images of the tennis ball and retrieve more accurate motion information from a video sequence, we used a specialized high-speed monochrome camera (SONY XC-HR58). This camera can capture images at a resolution of 782x582 at a frame rate of 50 fps. We decided to use a monochrome camera for two main reasons: to limit the processing burden on the software, and to minimize the effect of changing lighting condition. With the ability of adjusting the shutter speed, this camera can clearly record objects moving at various speeds. We eliminated any blurring effect due to the high speed motion by selecting a higher shutter speed (500ns). At the same time, a faster shutter speed also results in lower illumination because of the shorter exposure time. Therefore, the scene appears dark when a higher shutter speed is selected. By analyzing videos, we found the tennis ball still shows up very well with dim illumination. According to the camera specification, this camera is the most sensitive to light with wavelength of 500nm (see Figure 4.2). The wavelength of the yellow green color of the tennis ball is 550 nm, which is close to the highest sensitivity that this camera responds to, thus making these cameras highly sensitive to the tennis ball. These properties are common to most high speed cameras.

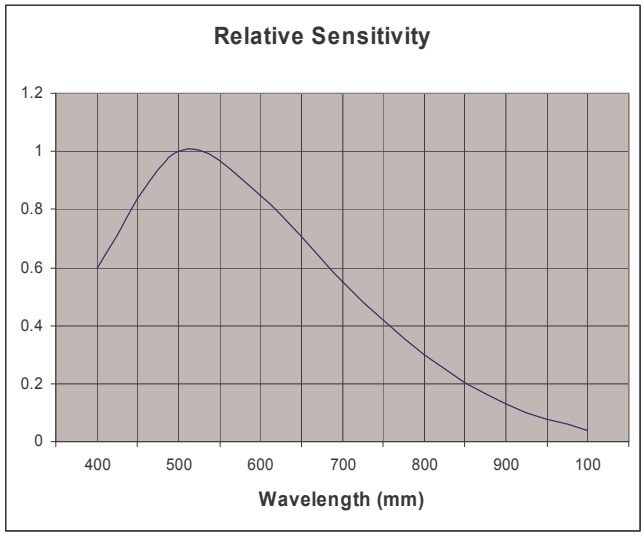


Figure 4.2: Various sensitive levels based on different wavelengths.

Because the image sizes of the tennis ball are small, and contours of the tennis ball are not very strong, edges of the tennis ball are difficult to detect. The technique

introduced by our first approach, which works with the edge instead of actual images, is no longer necessary. The techniques introduced in this chapter work directly with the actual gray images.

4.2 BACKGROUND SUBTRACTION WITH VERIFICATION

Our first technique works as follows: a background model is created from a collection of background images. For each queried image, foreground objects are first detected by background subtraction. The area of the player is also detected. Foreground objects residing inside the area of the player are removed. The remaining candidates are further verified. An overview of this technique is illustrated in Figure 4.3 (details will be presented in the following sections). The system flow is shown in Figure 4.5.

- Given a number of background images, a background model is created. This background model is represented by an average background image.
- For each queried image:
    1. Tennis ball candidates are detected by background subtraction.
    2. The player is detected by a modified background subtraction. The area of the player includes the part of the human body and the tennis racket. Ball candidates within the region of the player are discarded.
    3. The remaining ball candidates are subjected to the size, compactness, aspect ratio, roughness, and dynamics verifications.
    4. After the ball is detected, the system dynamics is updated based on the measurement.
    5. Finally, the location of the ball in the current image is detected.

Figure 4.3: Background subtraction with verification.

The rest of this section is organized as follows: section 4.2.1 describes the process of background model creation. Section 4.2.2 describes the process of ball candidate detection. Section 4.2.3 describes the process of the player detection. Finally, further verification is discussed in section 4.2.4.

4.2.1 BACKGROUND MODEL CREATION

To create the background model, first the background images need to be smoothed in order to remove noise. Once a number of smoothed background images are collected, an average calculation is performed on each pixel location. The result is an average background grayscale image. An example of such an image is shown in Figure 4.4.

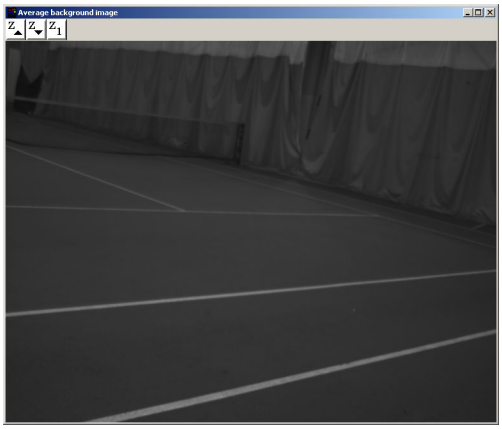


Figure 4.4: An average background image.

Once the background model is created, the tennis ball can be detected by the technique described in the following sections.

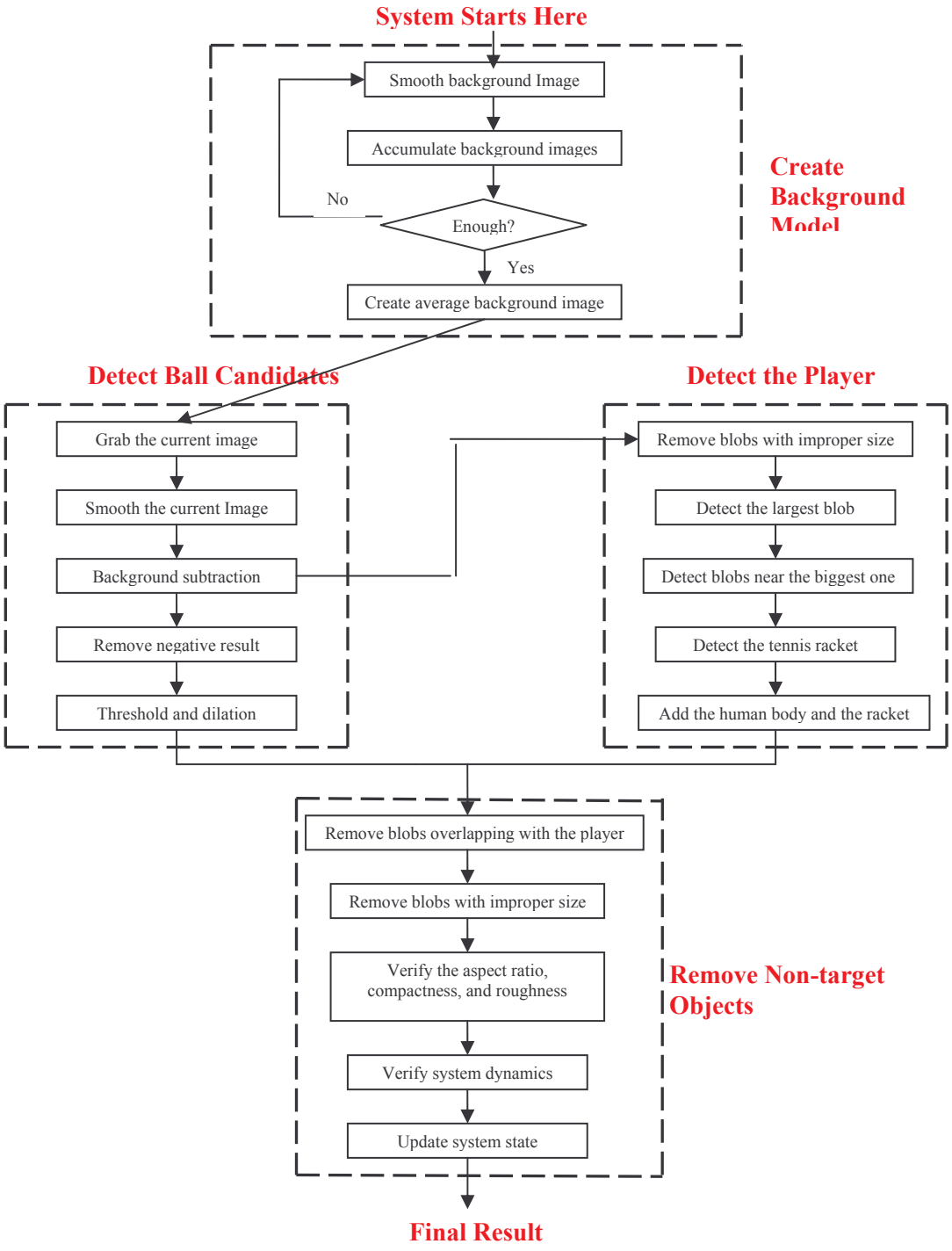


Figure 4.5: The system flow of our first technique.

4.2.2 BALL CANDIDATE DETECTION

To detect the tennis ball from an image, we first identified ball candidates by the technique of background subtraction. The detail of detecting ball candidate is illustrated in Figure 4.6.

1. For a given image, small noise is removed by the Median filter.
2. Subtract the average background image from the smoothed image.
3. Pixels with a negative result are removed. Because the tennis ball has high visibility, its gray value is usually higher than that of the dark background. The result of subtraction between the region of the tennis ball and the same region in the background is usually positive.
4. Remaining pixels are thresholded and dilated.
5. The result represents the ball candidates.

Figure 4.6: Ball candidate detection.

An example of ball candidate detection is provided in Figure 4.7: for a given image (a), it is first smoothed to remove small noise. The result is shown in (b). Figure 4.7 (b) is then subtracted by the background image Figure 4.4. The result is shown in (c). Pixels with negative results are removed. The result is shown in (d). After performing the threshold and the dilation, ball candidates are detected as shown in (e).

Once ball candidates are detected, we remove those which do not represent the tennis ball. This is achieved in two stages: first, we remove candidates belonging to the player (Details are discussed in section 4.2.3). Then candidates which do not satisfy the predefined criteria and dynamic property are eliminated (Details are discussed in section 4.2.4).

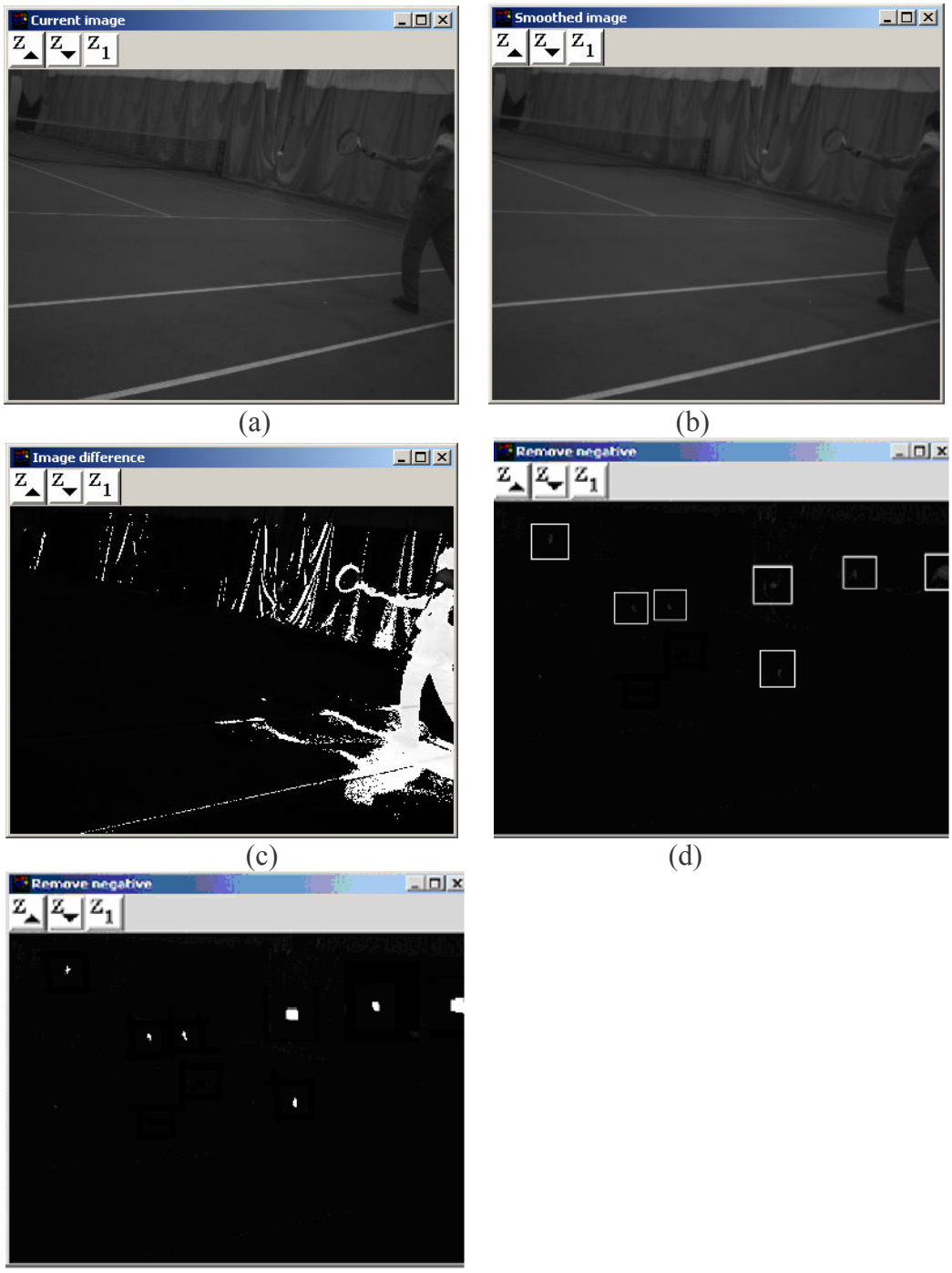


Figure 4.7: An example of ball candidate detection.

4.2.3 PLAYER DETECTION

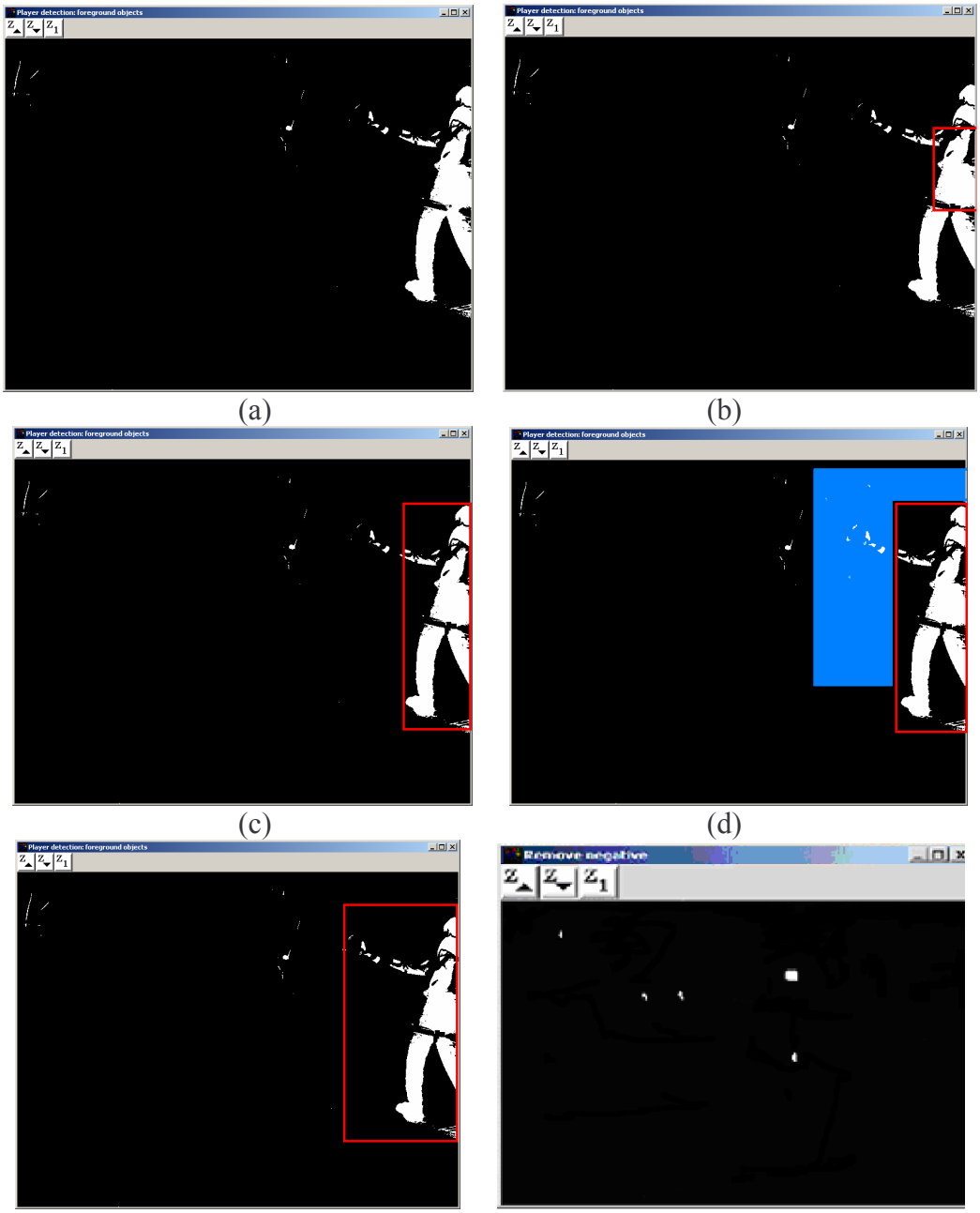
The player is detected as follows: first, the result of the background subtraction generated from the previous step, i.e. Figure 4.7(c), is preprocessed to remove small blobs. The largest blob is detected which belongs to part of human body. The other part

of human body is also detected based on the size and the location. The human body is then constructed from all the detected blobs. Next, the tennis racket is also detected from the blobs near the human body. Once the tennis racket is found, the player is detected which includes the part of human body and the tennis racket. The detail of player detection is illustrated in Figure 4.8.

1. Remove small blobs whose size is less than 10 pixels.
2. Detect the biggest blob. Normally the biggest blob belongs to a part of the human body.
3. Detect other part of the human body. The blobs which are near the biggest one and whose size is greater than 20 pixels are considered as a part of the human body.
4. Construct the human body. The human body is identified by a rectangle which covers all selected blobs.
5. Set the search region for detecting the tennis racket. The search region of the tennis racket is set to an area surrounding the human body. Suppose the bounding box of the human body is identified as (LeftX, LeftY, RightX, RightY), where (LeftX, LeftY) and (RightX, RightY) are the coordinates of the top-left and bottom-right corners of the rectangle. The search region is designed as: (LeftX – W, LeftY – 0.3*H, RightX + W, RightY – 0.2*H) excluding the area covered by the human body, where 'H' and 'W' are the height and the width of the human body rectangle. If some part of the search region is out of the image, the search region is limited by the boundary of the image.
6. Detect the tennis racket. Blobs in the search region are identified as part of the tennis racket if they satisfy the following condition: A/D > T, where 'A' is the size of the blob, 'D' is the distance between the center of the blob and the center of the human body rectangle, 'T' is the threshold value which depends on the location of the camera.
7. Finally, the player is detected by grouping the region of the human body and blobs of the tennis racket.

Figure 4.8: Player detection.

An example of player detection is presented in Figure 4.9: (a) Small blobs are first removed. (b) The biggest blob is identified by the red box. (c) The human body is detected by adding blobs near the biggest one. (d) The tennis racket is searched in the blue area. (e) The player is detected which includes the part of the human body and the tennis racket. (f) The candidates which are inside the region of the player are removed.



(a) (b) (c) (d) (e) (f)

Figure 4.9: An example of player detection.

After removing candidates residing within the region of the player, the rest of the candidates are further verified against various criteria. Details are discussed in the following section.

4.2.4 FURTHER VERIFICATION

The remaining candidates are further verified. Candidates with improper size, shape are removed. In addition, remaining candidates are verified with the dynamic information associated with the tennis ball. Details are given in Figure 4.10.

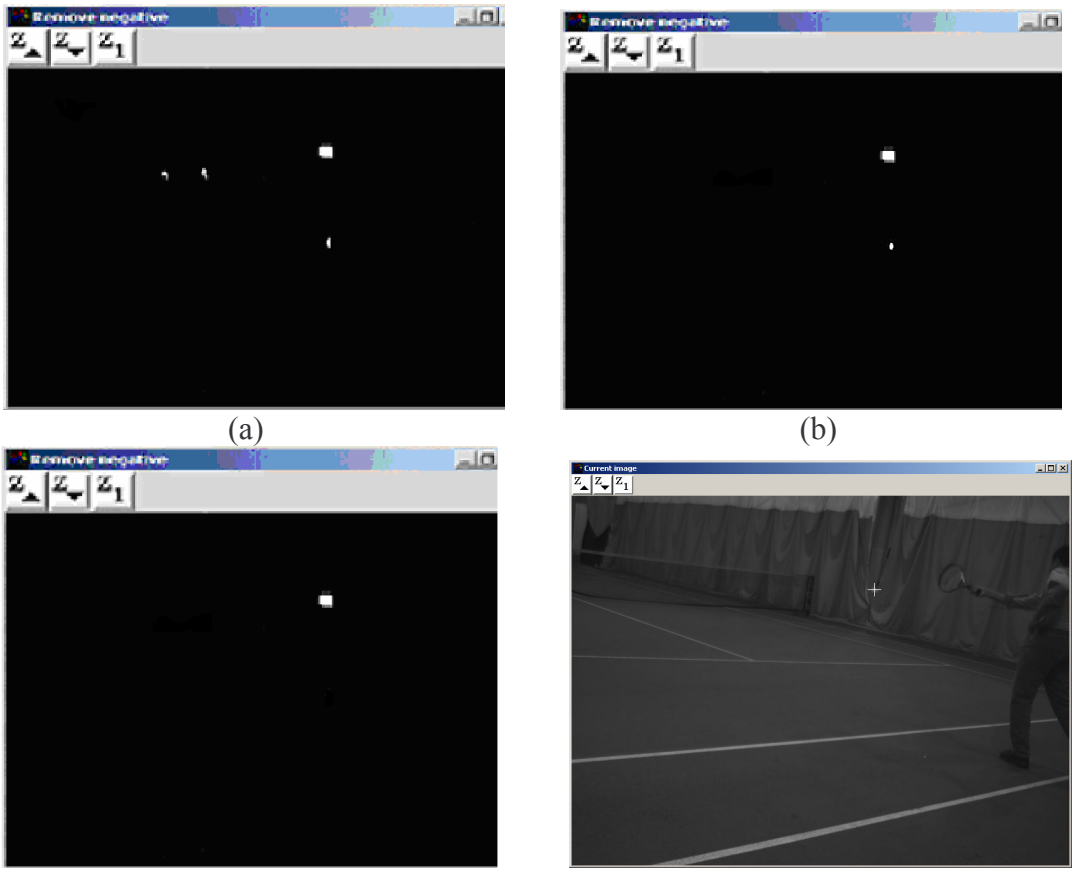
1. Normally the size of the tennis ball ranges from 5 to 60 pixels. Candidate with size out of this range are removed.
2. Candidates with improper shape are removed. Blobs with the aspect ratio, compactness, and roughness near to 1.0 are considered potential candidates of the tennis ball, others are removed.
3. The remaining blobs are subjected to the dynamics verification. This is implemented by using the Kalman filter [35]. The location of the ball is estimated by the Kalman filter. Candidates whose distance to the predicted location exceeds a predefined threshold are removed.
4. Finally, the remaining candidates are detected as the tennis ball.

Figure 4.10: Further verification.

Once the location of the tennis ball is detected, the dynamic information associated with the ball is updated by taking into account of the detected location as described by Kalman [35].

An example of further verification is presented in Figure 4.11: (a) Candidates with improper size are removed. (b) Candidates with improper shape are removed. (c) Candidates which are not close to the estimated location are removed. (d) The remaining candidate is detected as the tennis ball, as indicated by the white cross.

Several balls can move simultaneously in the scene. It is very difficult to detect which one the player currently focuses on. Therefore, our technique does no distinguish them and allows multiple occurrences of the ball.



(a) (b)

(c) (d)

Figure 4.11: An example of the further verification.

In this section, we presented our first technique of background subtraction with verification: first, a background model is created. For each queried image, ball candidates are detected by background subtraction. Candidates inside the area of the player are removed. The results are further verified with shape and dynamic information. Finally, the remaining candidate is detected as the tennis ball.

In the next section, we are going to present our second technique which models the value of each background pixel as a single Gaussian.

4.3 IMAGE DIFFERENCING BETWEEN THE CURRENT AND PREVIOUS FRAMES

Our second approach is inspired by the technique introduced by Pingali et al. [52]. The authors presented a real-time multi-camera tennis ball tracking system. Ball

detection is achieved by frame differencing between the current and previous images. The results are then verified against the size and shape (aspect ratio) parameters. The region which is close to the expected position is chosen in the case of multiple detections. But this technique has the problem of double detections. The authors solved this problem by first finding regions in the current image that lie in the expected intensity range for the ball. They then performed a logical AND operation between the results generated from image difference and the regions of the expected ball position. The occurrence of the ball representing the location in the previous image is wiped out.

However, determining the proper intensity range for the ball in the current image is tricky. In addition, the expected intensity range changes constantly across all images. Therefore, a very complex mechanism is required to estimate the expected intensity range for each image. Instead, we solved the problem of double detection using the technique of background subtraction.

Our second technique works as follows: first, a background model is created which models the value of each background pixel as a single Gaussian distribution. Ball candidates are detected from image differencing between the current and previous frames. Candidates generated from the previous image are removed. Candidates inside the region of the player are also removed. The results are further verified with shape and dynamic information. The remaining candidate is detected as the tennis ball. Finally, the background model is updated given the detected location. The detail of our second technique is illustrated in Figure 4.12. The system flow is presented in Figure 4.13.

- Given a number of background images, a background model is created. The value of each background pixel is modeled with a single Gaussian distribution.
- Given the current image ***A*** and the previous image ***B,*** the tennis is detected as follows:
    1. Ball candidates are detected by image differencing between ***A*** and ***B***. The result is denoted as ***C***.
    2. Perform background subtraction to find regions representing possible foreground objects in the current image. The result is ***D.***
    3. Perform a logical AND operation between ***C*** and ***D*** resulting in a number of ball candidates ***E***.
    4. Detect the player. Among ***E,*** ball candidates within the region of the player are removed.
    5. The remaining ball candidates are subjected to the size, compactness, aspect ratio, roughness, and dynamics verifications.
    6. Finally the location of the ball in ***A*** is detected. The background model is updated given the detected location.

Figure 4.12: Image differencing between the current and previous frames.

The rest of this section is organized as follows: section 4.3.1 describes the process of background model creation. Section 4.3.2 describes the process of ball candidate detection. Section 4.3.3 briefly describes the process of player detection. Finally, further verification and background model update are discussed in section 4.3.4.

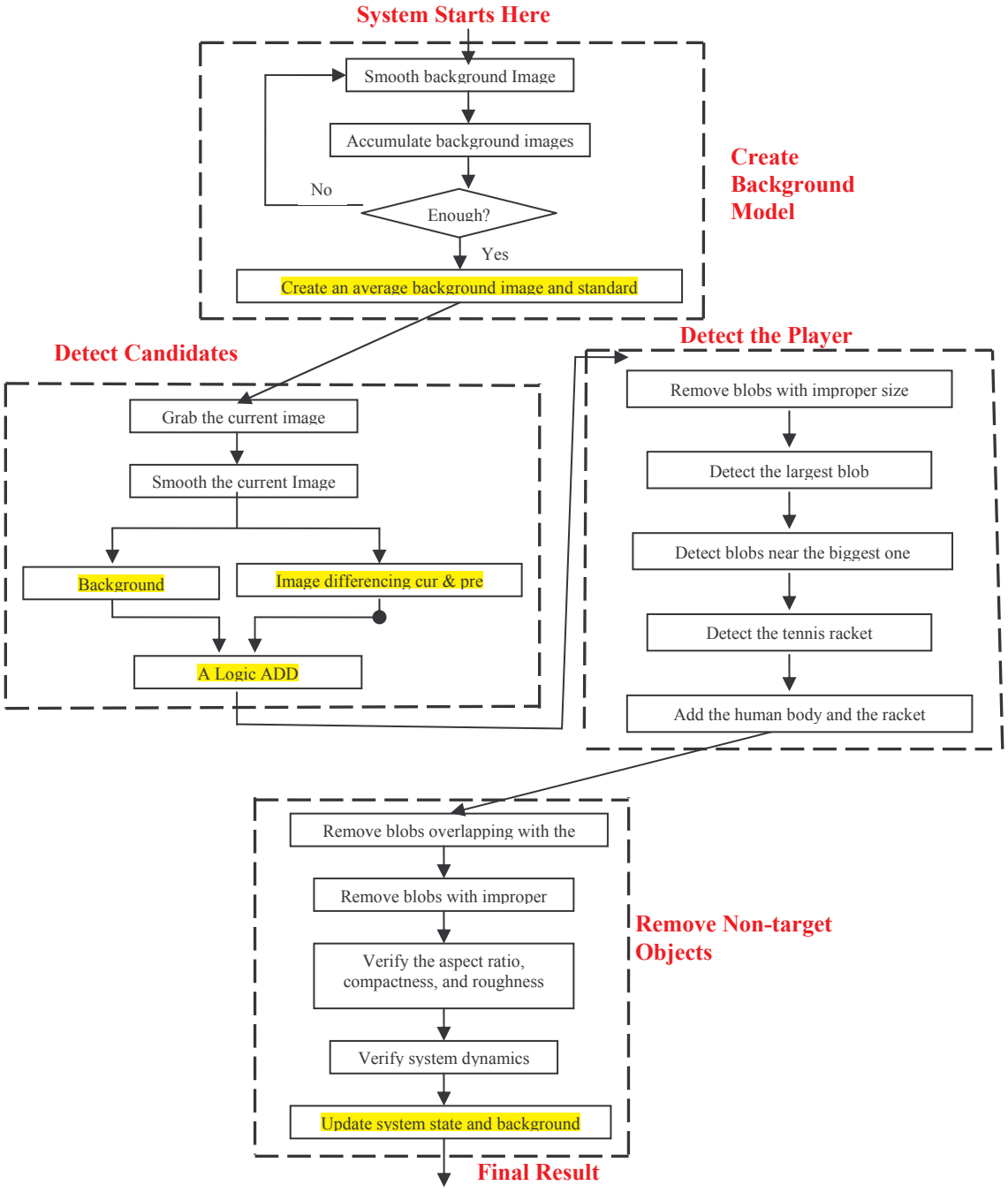


Figure 4.13: A visualization of our second technique.

4.3.1 BACKGROUND MODEL CREATION

The background model used in this technique consists of the average background image and standard deviation for each pixel location. The gray value of each background pixel is assumed to follow a Gaussian distribution. To create the background model, first the background images are smoothed by the Median filter. Once a number of smoothed

background images are collected, a background model consisting of an average background image and standard deviations is created. An example of the average background image is shown in Figure 4.14.

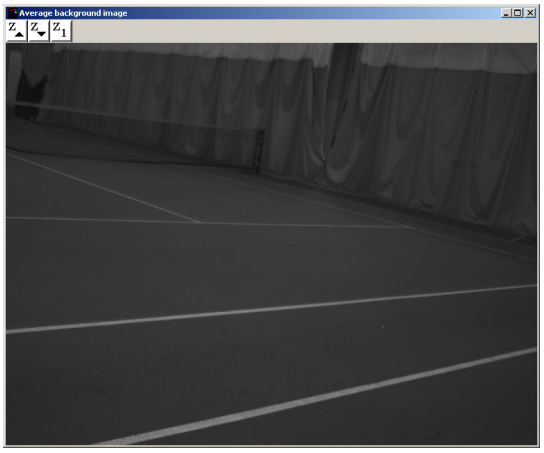


Figure 4.14: An average background image.

Once the background model is created, the tennis ball can be detected by the technique described in the following sections.

4.3.2 BALL CANDIDATE DETECTION

Given the current and previous images, ball candidates can be detected as follows: first, subtract the previous image from the current image. To remove candidates generated from the previous image, a background subtraction is performed between the current and the average images. A logical AND operation is then performed between the result of the background subtraction and the result of image differencing. Finally, ball candidates from the current image are detected. The detail of detecting ball candidates is illustrated in Figure 4.15.

Given the current image ***A*** and the previous image ***B***, ball candidates from ***A*** are detected as follows:

1. Subtract ***B*** from ***A***. The result is denoted as ***C***.
2. Subtract ***A*** by the average background image. Label pixels as the background if calculated result is less than two times standard deviation. Those pixels not satisfying this condition are labeled as foreground objects. The result is denoted as ***D***.
3. A logical AND operation is performed between ***C*** and ***D***. Results generated from two non-zero operands is 1 and 0 otherwise.
4. The results are the ball candidates.

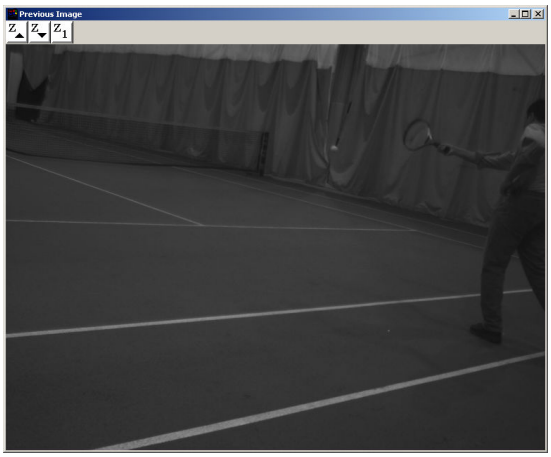
Figure 4.15: Ball candidate detection.

An example of ball candidate detection is given in Figure 4.16: (a) the previous image. (b) The current image. (c) The image differencing between the current and the previous images. (d) The foreground objects detected by background subtraction. (e) The ball candidates are found by a logical AND operation between (c) and (d).

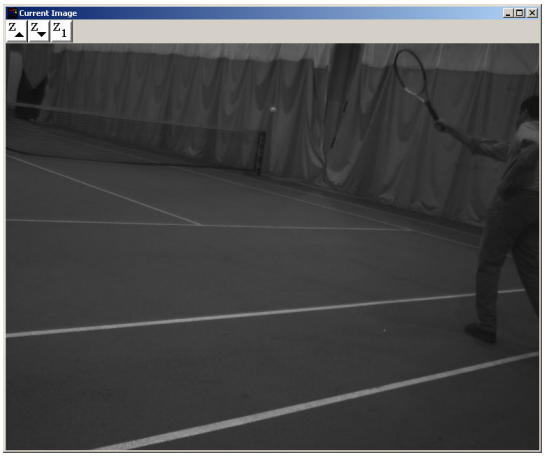
Once ball candidates are detected, we remove those which do not represent the tennis ball. This is achieved in two stages: first, we remove candidates belonging to the player (Details are discussed in section 4.3.3). Then candidates not satisfy the predefined criteria and dynamic property are eliminated (Details are discussed in section 4.3.4).

4.3.3 PLAYER DETECTION

The player is detected using the same approach as described in our first technique. The difference is that our second technique uses a different background model. The detected result is shown in Figure 4.17 (a).



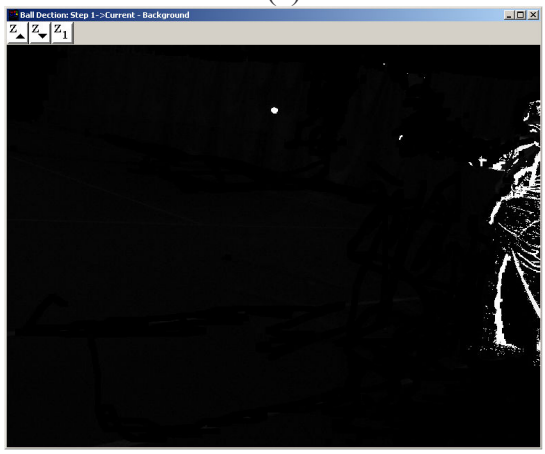
(a)



(b)

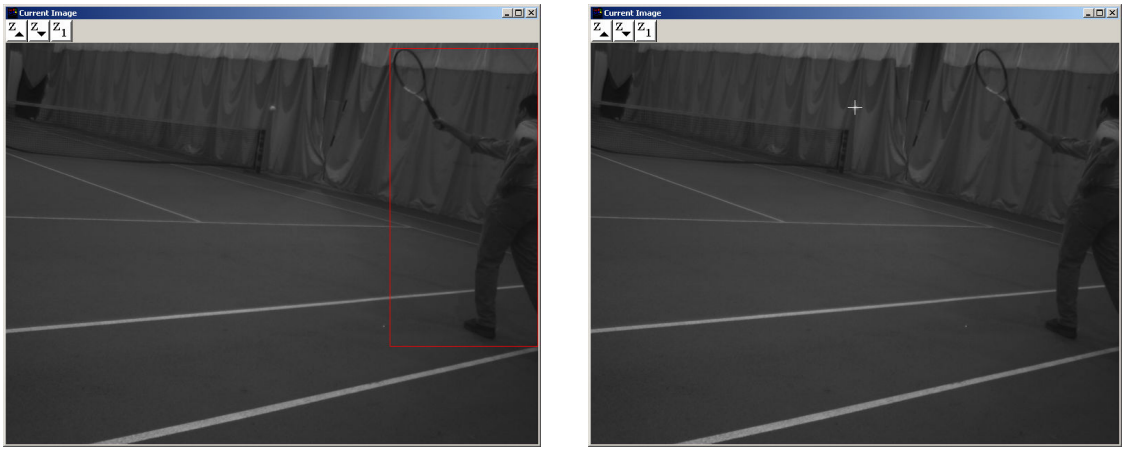
(c)

(d)



(e)

Figure 4.16: An example of ball candidate detection.



(a) (b)

Figure 4.17: Detect the player and the tennis ball.

After removing candidates residing within the region of the player, the rest of the candidates are further verified.

4.3.4 FURTHER VERIFICATION

The same verifications of shape and dynamic information, as discussed in our first technique, are performed to the remaining ball candidates. The final result is shown in Figure 4.17 (b).

Once the location of the tennis ball is detected, the background model is updated by taking into account the detected location. The average background image and standard deviation image are updated as follows:

$$B_t = (1-\rho) * B_{t-1} + \rho * C_t, \tag{4.1}$$

$$\sigma_t^2 = (1-\rho) * \sigma_{t-1}^2 + \rho * (C_t - B_t)^2, \tag{4.2}$$

where B is the average background image, C is the current image, t and t-1 are the time stamps, and ρ is the learning rate.

In this section, we presented the second technique of image differencing between the current and previous images: first, a background model is created which models the value of each background pixel as a single Gaussian distribution. For each queried image, ball candidates are detected by image differencing between the current and previous images. Candidates generated from the previous image are removed using background subtraction. Candidates inside the area of the player are also removed. In addition, the results are further verified with shape and dynamic information. The

remaining candidate is detected as the tennis ball. Finally, the background model is updated given the detected location.

In the next section, we are going to present our third technique which models the value of each background pixel as a mixture of Gaussian distributions.

4.4 ADAPTIVE BACKGROUND MODELING USING A MIXTURE OF GAUSSIANS

Our third approach is inspired by the technique introduced by Stauffer et al. [61]. The authors presented a background modeling technique using a mixture of Gaussian distributions. This type of background model is robust and dynamic compared with normal background modeling techniques, such as the average background image and single Gaussian per pixel techniques used in our first two methods. We applied this technique to our third method. The basic structure of our third technique is similar to our first technique: first, a background model is created which models the value of each background pixel as a mixture of Gaussian distributions. Ball candidates are detected by subtracting the background image from the current image. The area of the player is located and any candidates residing within the area of the player are removed. Remaining candidates are further inspected based on aspect ratio, compactness, roughness, and dynamics verification. The remaining candidates are detected as the ball. Finally, the background model is updated given the detected location.

The rest of this section is organized as follows: section 4.4.1 describes the process of background model creation. Section 4.4.2 illustrates the process of background model update.

4.4.1 BACKGROUND MODEL CREATION

Initially, there is no Gaussian distribution allocated for each background pixel. Once we start to process background images, every new pixel value is verified against its corresponding Gaussian distributions. It stops when a match is found or no distribution can be matched. A match is defined as a pixel value within 2.5 standard deviations of a distribution.

If none of the existing distributions matches the current pixel value, it updates the least probable distribution when the maximum number of Gaussians has already been created. In this case, the least probable distribution is replaced with a distribution with the current pixel value as its mean value, an initially high standard deviation, and low weight. It can also add a new Gaussian if the maximum number of Gaussians has not been reached. The newly created Gaussian is given the new pixel value as its mean, a relative high standard deviation, and low weight.

If a match is found, the weights of the existing distributions at time t are updated as follows:

$$w_{k,t} = (1-\alpha) * w_{k,t-1} + \alpha * M_{k,t}, \tag{4.3}$$

where α is the learning rate and $M_{k,t}$ is 1 for the distribution that matches and 0 otherwise. As a result of this assignment, the relative weight of the matched Gaussian is increased while others are decreased. Once the reassignment is done, the weights of the existing distributions are normalized.

The learning rate determines how fast this model responds to changes of the environment. Given a higher learning rate means changes to the environment will be adapted to the existing background model more quickly.

The parameters of the matched Gaussian distribution are updated as follows:

$$\mu_t = (1-\rho) * \mu_{t-1} + \rho * X_t, \tag{4.4}$$

$$\sigma^2 = (1-\rho) * \sigma_{t-1}^2 + \rho * (X_t - \mu_t)^2, \tag{4.5}$$

where $\rho = \alpha\eta(X_t \mid \mu_k, \sigma_k)$ and $\eta(X_t \mid \mu_k, \sigma_k)$ is calculated as:

$$\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(X-\sigma)^2}{2\sigma^2}}. \tag{4.6}$$

Once we update the parameters and the weights associated with each Gaussian, the existing Gaussians are ordered by the value of ϖ/σ. The distribution with higher weight ω and less standard deviation σ will be a more probable representation of the current background in that pixel location.

This process is repeated until the background model is gradually constructed and refined. When this background model is created, the gray value of each pixel is

represented as a mixture of weighted Gaussian distributions. The process of background model creation is illustrated in Figure 4.18.

4.4.2 BACKGROUND MODEL UPDATE

Once the background model is created, it can be easily applied to the process of object detection: every pixel in the current image is detected as the background if its value matches one of its Gaussian distributions. If no match is found, this pixel is detected as a foreground pixel. This background model is updated each time after a new image is processed. The parameters and weights associated with each distribution are updated in the same way as described above. The process of updating the existing background model is illustrated in Figure 4.19.

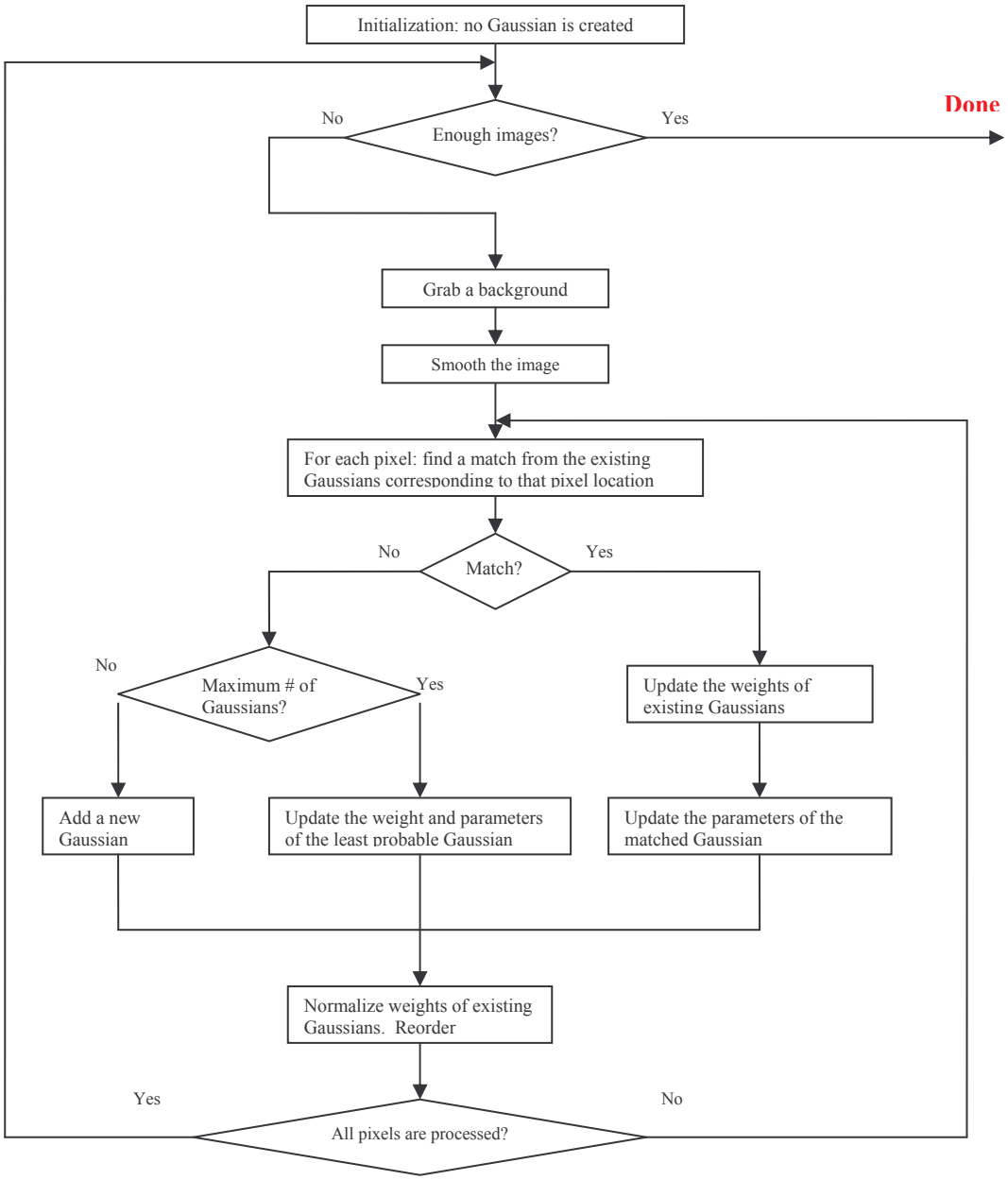


Figure 4.18: A visualization of background model creation.

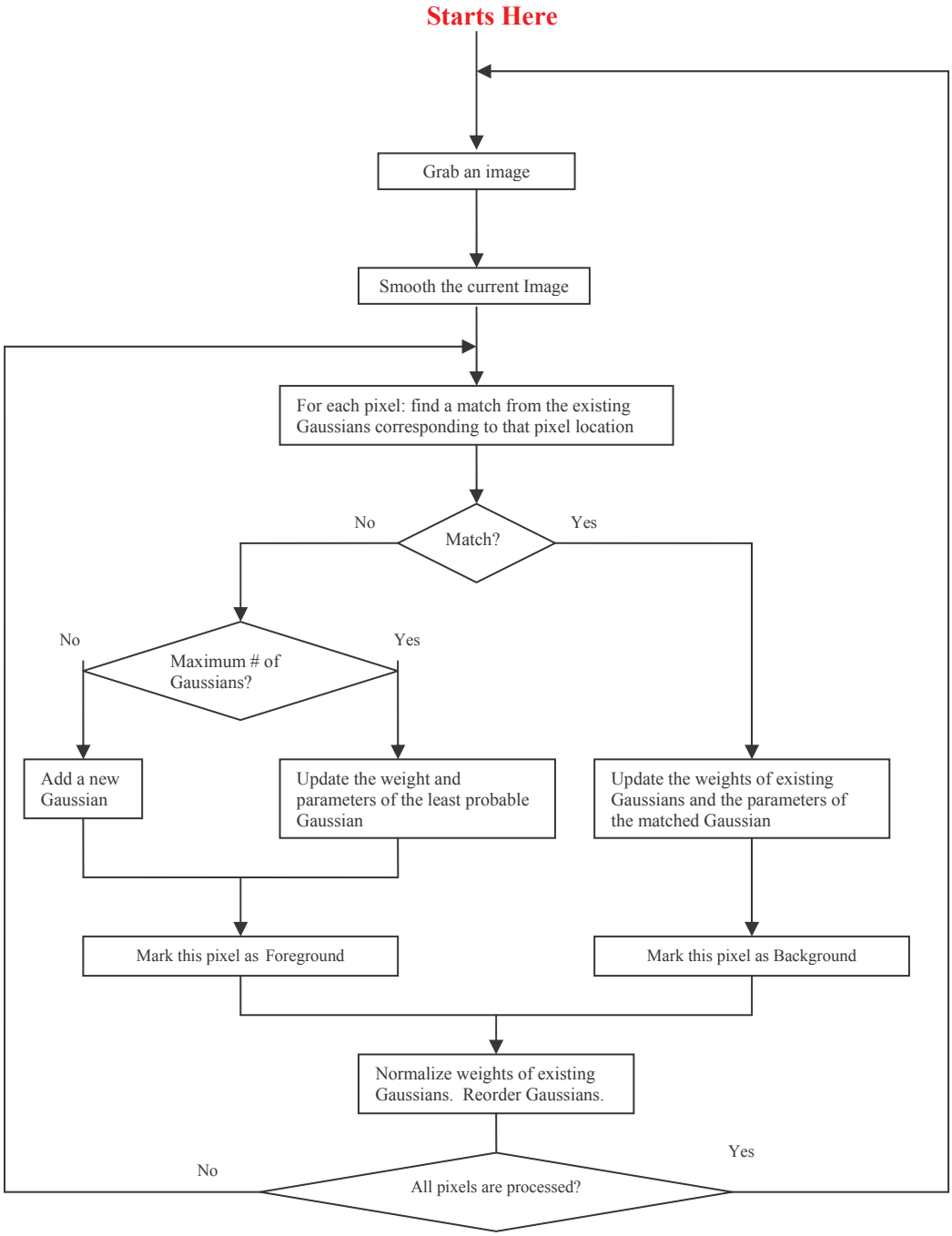


Figure 4.19: A visualization of object detection and background model update.

In this chapter, we presented three extended background subtraction techniques. They include: background subtraction with verification, image differencing between the current and previous images, and background modeling using a mixture of Gaussian distributions.

In the next chapter, the performance of our techniques will be systematically analyzed. The performance of each technique will be compared. In addition, the usability and limitations of our techniques will also be discussed.

CHAPTER 5

EVALUATION

The performance of our solutions is measured by three figures: the true positive rate, the false alarm rate described by Klette et al. [38], and the processing speed. The true positive rate, or the recognition rate in the case of image processing, is the rate of positive responses in the presence of feature instances, i.e., the ratio of the number of correctly recognized features compared to the total number of ground truths. The false alarm rate, or the false positive rate, is the rate of positive response in the absence of the features, i.e., the ratio of the number of incorrectly recognized features compared to the total number of ground false. It is desired that the true positive rate is maximized while the false alarm rate is minimized. In addition, the processing speed of our solutions is measured by the frame rate, i.e. frames processed per second.

The evaluation was performed in three stages: we first recorded several videos from both an indoors and outdoors tennis court. Then each video was processed by different techniques. Finally, the results of the indoors and outdoors cases are analyzed.

The rest of this chapter is organized as follows: section 5.1 describes the process of video recording. Section 5.2 describes the process of video processing. Section 5.3 and 5.4 analyze the results of the indoors and outdoors cases. Section 5.5 discusses the deployment of our techniques to real-world applications. Finally, we give conclusions in section 5.6.

5.1 VIDEO RECORDING

To test the reliability and robustness of our solutions, sixty real tennis matches were recorded in both an indoors and outdoors tennis court. The length of each video is about 1 minute. The background, lighting, contrast, and shadow conditions of each tennis court were quite different. Videos were recorded by the SONY monochrome camera. This

camera can capture frames at 50 fps. However, this frame rate is achieved just by simply displaying the image sequences. More time is required to save images to hard disk and create videos. Therefore, we can only record videos at about 25 fps. This is still much faster than the webcam which can record only 8fps.

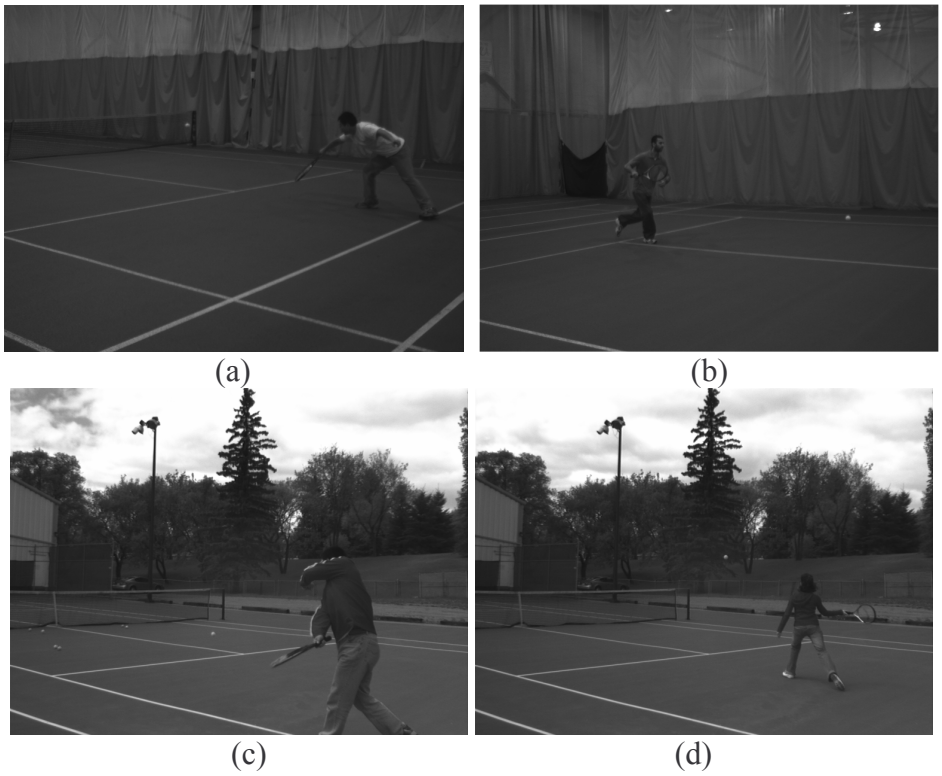


Figure 5.1: Sample images taken from an indoors and outdoors tennis court.

These videos were recorded at different positions beside the court so the entire scene is covered. Each video is about 1 minute and contains roughly 1500 images at a resolution of 782x582 pixels. Examples of indoors and outdoors images are shown in Figure 5.1.

5.2 VIDEO PROCESSING

Each video was processed by our techniques described in the previous chapter:

- ***BS*:** background subtraction with verification.
- ***ID*:** image differencing between the current and previous images.
- ***MG*:** adaptive background modeling using a mixture of Gaussian distributions.

To compare with our techniques, these videos were also processed by two of the techniques described in chapter 3. We are not able to test the color segmentation approach since the camera is monochrome. These compared techniques are:

- ***SH*:** background subtraction with shape recognition.
- ***BC*:** boosted classifiers trained with Haar-like features.

Volunteers then manually classified the results for each frame into the different categories as listed in Table 5.1.

The true positive rate and the false alarm rate can be generated based on the following equations:

$$true\ positive\ rate = \frac{\sum A}{\sum A + \sum B + \sum C} \tag{5.1}$$

$$false\ alarm\ rate = \frac{\sum D}{\sum D + \sum E} \tag{5.2}$$

Table 5.1: The classification of test results.

Detection / Result / Actual	A tennis ball is detected		No ball is detected
	Detected location is correct	Detected location is wrong	
A tennis ball is actually present	**A** (Correct)	**B** (Wrong location)	**C** (Miss)
A tennis ball is not actually present	**D** (False alarm)		**E** (Correct)

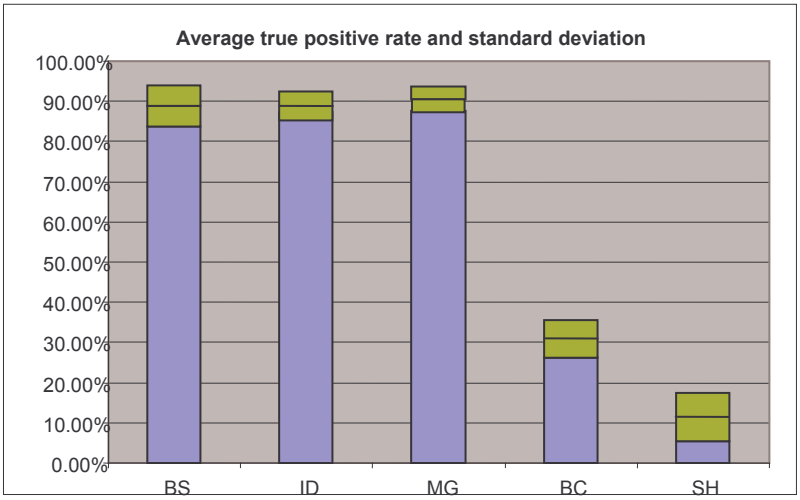
After all videos were processed, the indoors and outdoors results were presented and analyzed separately in the following sections. The performance of each technique was discussed and compared with each other. Several factors affecting the test results were identified and their influence on our techniques was also analyzed.

5.3 INDOORS RESULTS

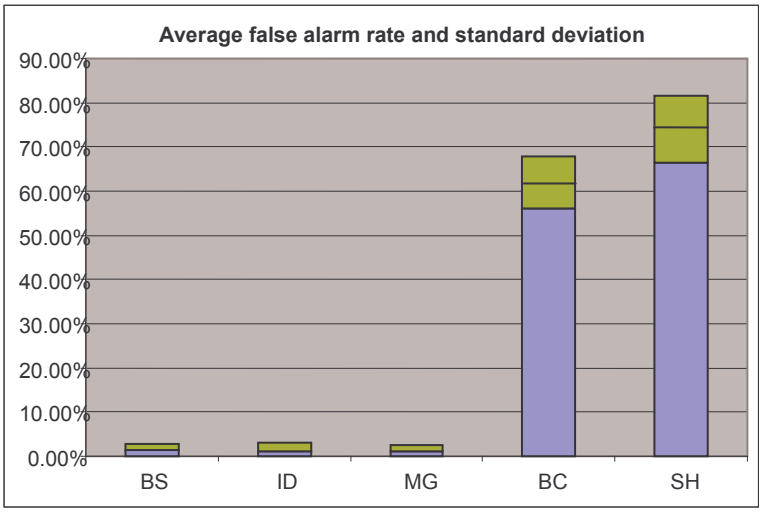
In this section, the results generated from the indoors environment are systematically analyzed: Section 5.3.1 discusses the general performance of our techniques compared with the BC and SH techniques. Section 5.3.2 discusses the elements influencing the test results of our techniques, which include the background models, and external factors. Section 5.3.3 analyzes the limitations of our techniques.

5.3.1 RESULTS ANALYSIS

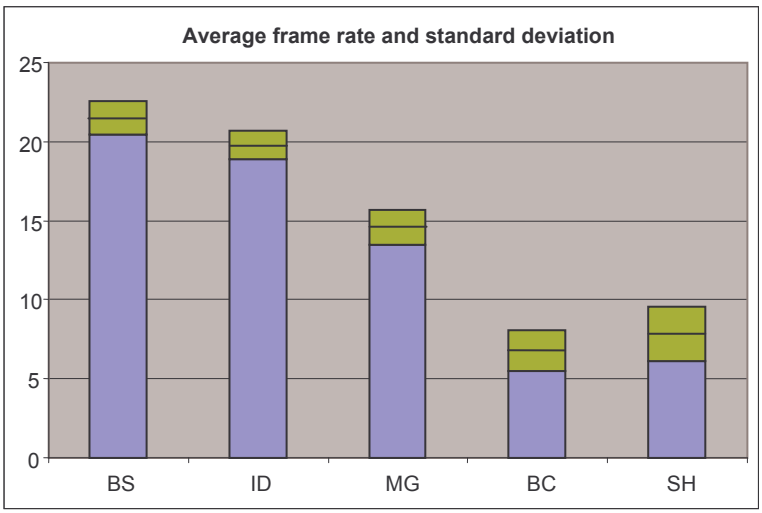
Overall our techniques demonstrate much higher detection accuracy and faster processing speed than the techniques of BC and SH. The average performance of each technique is illustrated in Figure 5.2. The vertical bars represent the average test results in terms of the true positive rate, false alarm rate and frame rate. The green area represents two times standard deviation and the mean value is indicated as the line in the middle of the green area. As shown in this figure, our techniques achieved approximately a 90% true positive rate which is more than three times higher than the BC and SH techniques; a less than 2% false alarm rate which is more than twenty times lower than the BC and SH techniques; and a frame rate of 20 fps which is about two times faster than the BC and SH techniques. In addition, the standard deviations of the true positive rate and false alarm rate generated from our techniques were 0.032 and 0.012 on average. These values are smaller than that generated from the BC and SH techniques (0.059 and 0.068 in their case). As a result, the performance of our techniques was more stable.



(a)



(b)

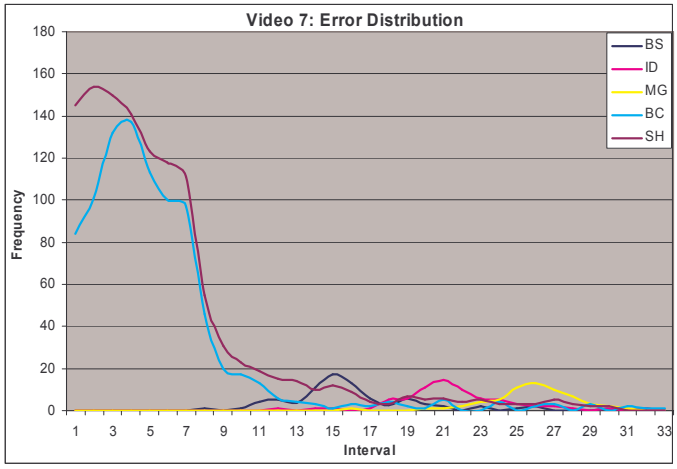


(c)

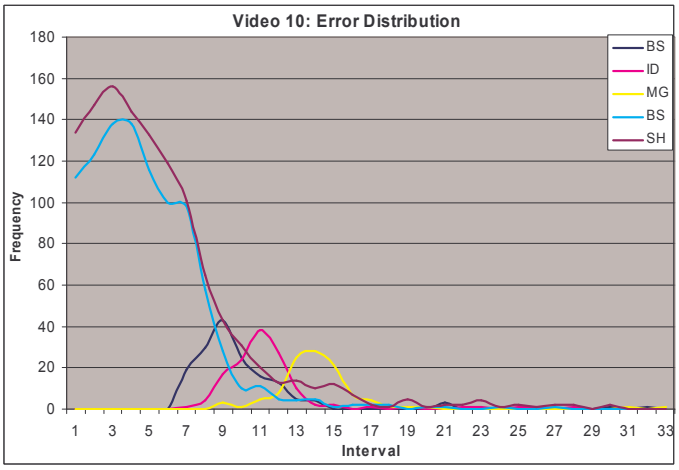
Figure 5.2: Average indoors performance.

To further evaluate the performance of our techniques, several videos were selected from which we created a histogram image. The histogram image was created as follows: first, we listed the frames where an error (e.g. false alert, missing the ball, or wrong location) was generated. The intervals between each adjacent error frames were calculated and the occurrence of different time intervals was then accumulated. From this result, we created a histogram image which illustrates how often an error happens: the x-coordinate of the histogram image represents different time intervals and the y-coordinate of the histogram image represents the frequency of each time interval. For example, suppose detection errors happened in frames 1, 4, 5, 8, and 13, the time intervals between each adjacent error frames are 3 (i.e. 4-1), 1 (i.e. 5-4), 3 (i.e. 8-5), and 5 (i.e. 13-8). Therefore, the frequencies of the time intervals 1, 3, and 5 are 1, 2, and 1 respectively. From this result, a histogram is created: the value of the x-axis ranges from 1 to 5; the values corresponding to x-coordinate 1, 3, and 5 are 1, 2, and 1 respectively. This histogram image illustrates how often an error happens and the distribution of errors. If errors happen frequently, the frequency of small time intervals is large. On the other hand, if errors happen rarely, the frequency of large time intervals is large. As you will see in section 5.3.2, one of the best results achieved by our techniques was from video 7, while one of the worst results was from video 10. The histograms generated from videos 7 and 10 are presented in Figure 5.3.

This figure shows that the BC and SH techniques generated more errors than our techniques. As a result, the true positive rate and the false alarm rate generated from the BC and SH techniques are worse. In addition, the BC and SH techniques generate errors more frequently than our techniques. As we can see, the frequency of short term intervals from the BC and SH techniques is quite big which means a large number of errors generated during a short period of time. The errors generated from the BC and SH techniques were concentrated in a few time periods during which the visibility of the ball was bad due to various factors such as distance and illumination changes. This discrepancy demonstrates that our techniques are more reliable and robust than the compared techniques.



(a)



(b)

Figure 5.3: Error distribution generated from videos 7 and 10.

We also noticed: there are no observed occurrences of consecutive errors as the smallest time interval is 6, which means the errors were sparsely distributed. We assume that errors would show up as outliers. Therefore, we would be able to use a smoothing/filtering operation to detect these errors. For example, if a detected location is too far way from the trajectory, an error is probably generated. For those identified errors, we can repair them by replacing their values with the result generated from interpolating the detected positions before and after the current errors; since there is going to be good data on both sides of the erroneous point. Based on this scheme, we would be able to develop a mechanism for detecting and repairing errors in the future.

5.3.2 ELEMENTS INFLUENCING THE TEST RESULTS

Based on the analysis, we found the results of our techniques are strongly affected by the accuracy of the background models. Techniques using a robust background model generate high detection accuracy but slow processing speed. On the other hand, techniques using a simple background model operate efficiently but the detection accuracy is reduced.

In addition, the test results are also affected by external factors such as the stability of background environment and the tennis ball's image size. Techniques performed in a stable environment generate higher detection accuracy and slightly faster processing. In addition, our extended solutions generate better result if the size of the tennis ball stays within the optimal range

In the following sections, we will analyze the influence of each element. Section 5.3.2.1 discusses the influence of background models. Section 5.3.2.2 discusses the influence of external factors.

5.3.2.1 Background Models

The performance of our techniques was strongly affected by the robustness and the accuracy of the background models. Given a more robust background model, foreground objects including the tennis ball can be detected with higher accuracy. As a result, the chance of missing or misclassifying the tennis ball is reduced, which consequently increases the true positive rate.

To analyze the causes of false alarms, we selected three videos with the highest false alarms (videos 7, 10, and 23) and recorded the sources generating false alarms. Based on the result, we found 82% of false alarms were generated by noise as shown in Figure 5.4.

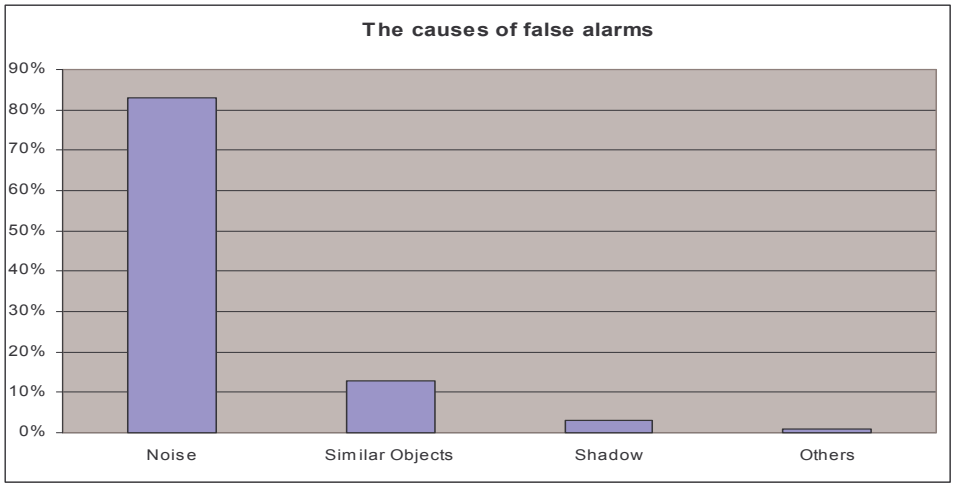


Figure 5.4: The main causes of false alarms. Noise is the main source of false alarms.

Figure 5.5: An example of false alarm generated from other unknown objects such as the one indicated as 2.

The accuracy of background models also influences the false alarm rate. A large amount of noise can be removed by background subtraction performed at the very beginning of our techniques. If a more robust and realistic background model is used, then more noise can be removed and consequently reduce the chances of false alarms. To prove this, we processed two randomly selected videos using our techniques and recorded the number of noise blobs per frame generated by background subtraction. The average result was then calculated and shown in Figure 5.6.

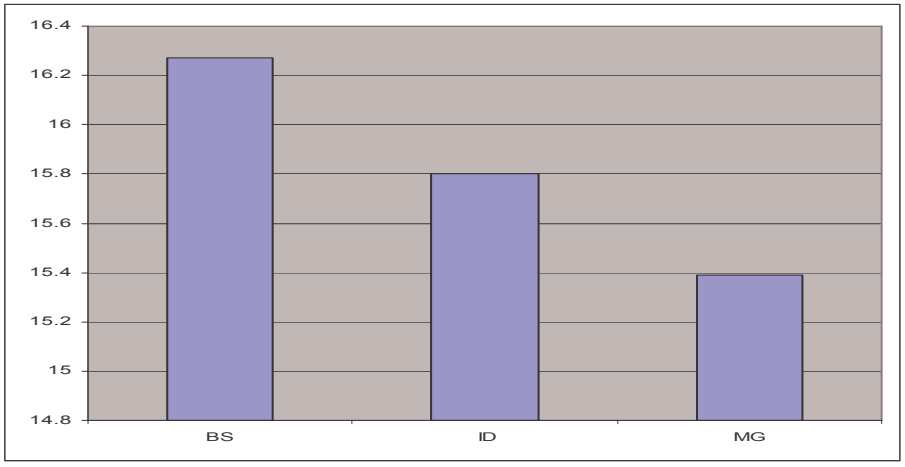


Figure 5.6: Average number of noise blobs generated from our techniques.

The processing speed depends on the number of image processing operations involved. The techniques that use simpler background models perform faster because they require fewer operations to detect the foreground objects and to update the current background model.

The background model used by the BS technique is a naive average of the image frames which is simple but lacks accuracy. As a result, the BS technique has the lowest positive rate (average 87.4%), the highest false alarm rate (average 1.35%), and the fastest processing speed (average 21 fps) among our techniques.

The ID technique models the background using a single Gaussian distribution pixel-wise. It is more accurate than the simple averaging but requires more time to develop the background model and detect foreground objects. Consequently, it has a higher positive rate (average 89.64%), a lower false alarm rate (average 1.07%), but a slower processing speed (average 19 fps).

The MG technique models the background using multiple Gaussian distributions and is the most accurate and robust among our techniques. It updates the current background model by adjusting the existing distributions in response to environmental changes. However, its computation cost is the highest of our techniques. This technique has the highest positive rate (average 90.73%), the lowest false alarm rate (average 1.02%), and the slowest processing speed (average 14 fps).

5.3.2.2 External Factors

In this section, two external factors affecting the test results are discussed, including the stability of the background and the tennis ball's image size. Based on these external factors, the performance of individual video was analyzed in terms of the true positive rate, the false alarm rate, and the processing speed. According to the analysis, we found the influence of these factors varies depending on the video's quality. Techniques applied to videos with better quality are less affected by these factors and consequently generate better result.

External Factors Analysis

Despite the accuracy of the background models, the results of our techniques were also affected by the stability of the background environment. A background environment is stable if it contains fewer distractions, such as illumination changes and other moving objects. Given a stable background environment, less noise or non-target candidates will be generated by our techniques. As a result, the chance of noise or non-target objects being detected as the tennis ball is reduced, which consequently improves the true positive rate and the false alarm rate. In addition, less effort is required to eliminate noise and non-target objects. Therefore, the processing speed is slightly faster in a stable environment.

Further, the image size of the tennis ball also influences the performance of our techniques. To demonstrate the influence that the image size has on our techniques, we processed three videos with the highest, the lowest and normal true positive rate (videos 5, 7, and 14) using our techniques and manually counted the detection results (either correct or wrong) based on the image size. The rates of correct detection based on each image size were then calculated and illustrated in Figure 5.7. This diagram shows that our techniques generate high true positive rate when the tennis ball size is between 15 and 35 pixels. The image of the tennis ball contains more accurate shape information when its size is in this range. The worst result is generated when the size is greater than 40 or less than 10 pixels. For a size less than 10 pixels, the tennis ball is most likely removed as a byproduct of noise elimination. For a size greater than 35 pixels, the performance declines because its size is past the predefined threshold. We can not solve

this problem by increasing the threshold because as we expand the threshold range, more noise will also be introduced, which reduces the detection accuracy.

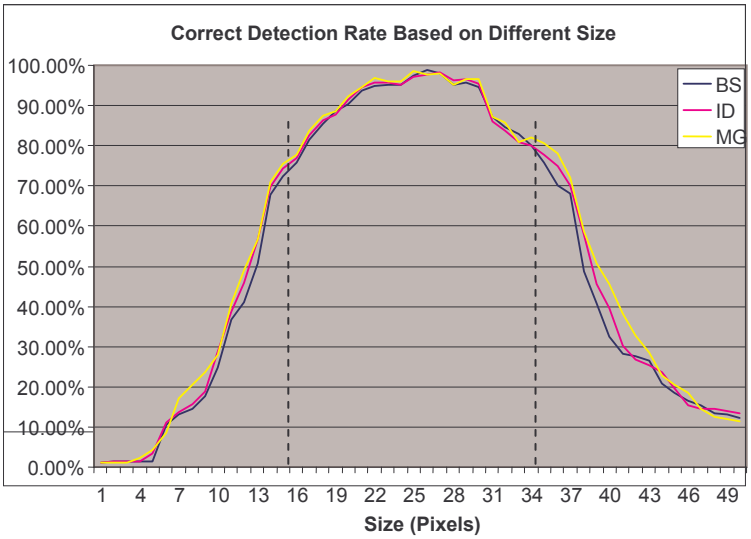


Figure 5.7: Correct detection rate generated from different image sizes of the tennis ball. The highest detection accuracy was generated when the size ranges from 15 to 35 pixels.

The Influence of the External Factors on Individual Result

The influence of the external factors on individual result is demonstrated in terms of the true positive rate, the false alarm rate, and the processing speed. Details are discussed as follows:

The true positive rate generated from each video by different techniques is illustrated in Figure 5.8. As shown in this figure, videos 5, 6, 13, and 20 always generate the highest true positive rate regardless of which technique was used. By analyzing these videos, we found the lighting condition in these videos is more stable than in the others. As shown in Figure 5.9 (a), the light sources did not directly expose to the camera. The flickering effect of the AC power frequency is less obvious than other videos. As a result, the background model created from these videos is more stable and accurate. To obtain the size information, we modified our techniques to record the distribution of the tennis ball's image size. The results show that in videos 5, 6, 13, and 20 the size of the tennis ball generally stays within the optimal range. For example, in video 13, there are 672 ball occurrences. Among them, 571 frames (85%) have a size between 20 to 30

pixels, 74 frames (11%) have a size between 10 to 20 pixels, in 20 frames (3%) the size is greater than 30 pixels, and 7 frames (1%) less than 10 pixels.

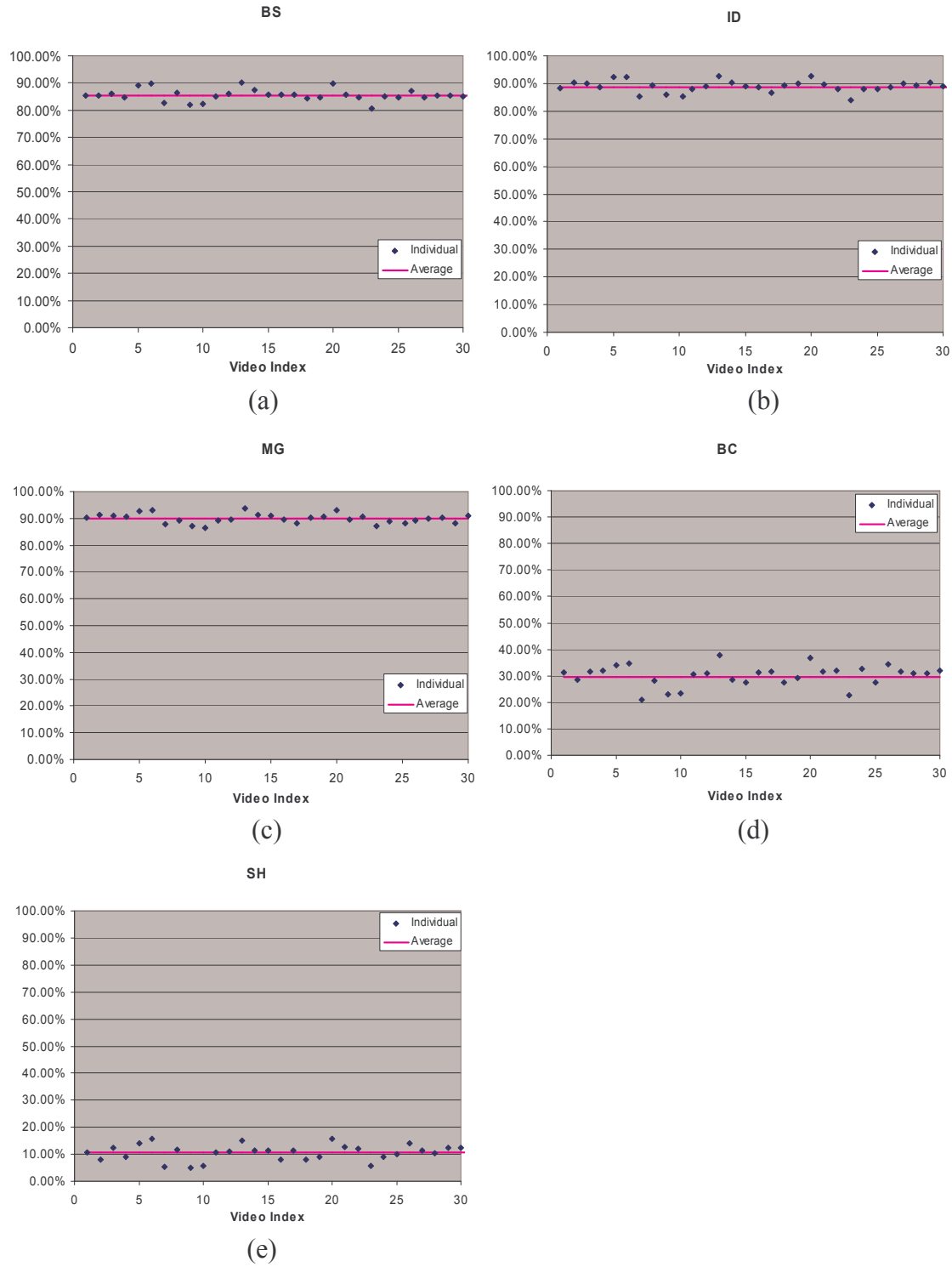
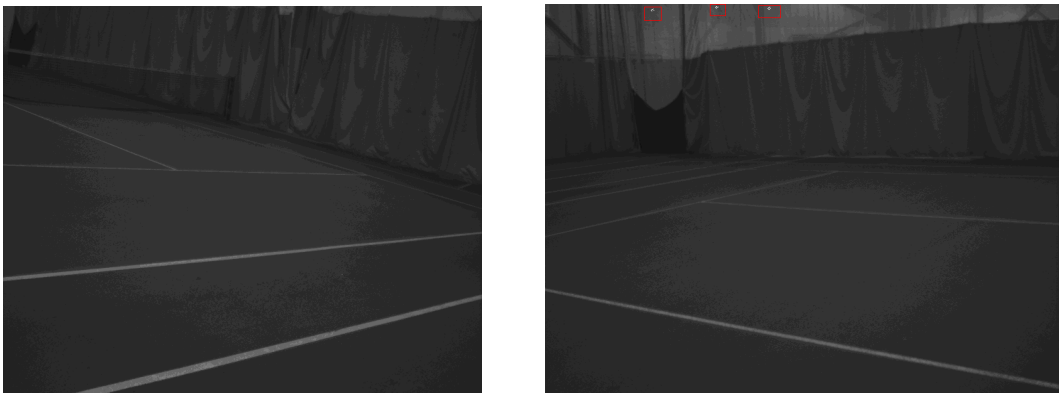


Figure 5.8: Indoors: true positive rate generated from each video by different techniques.

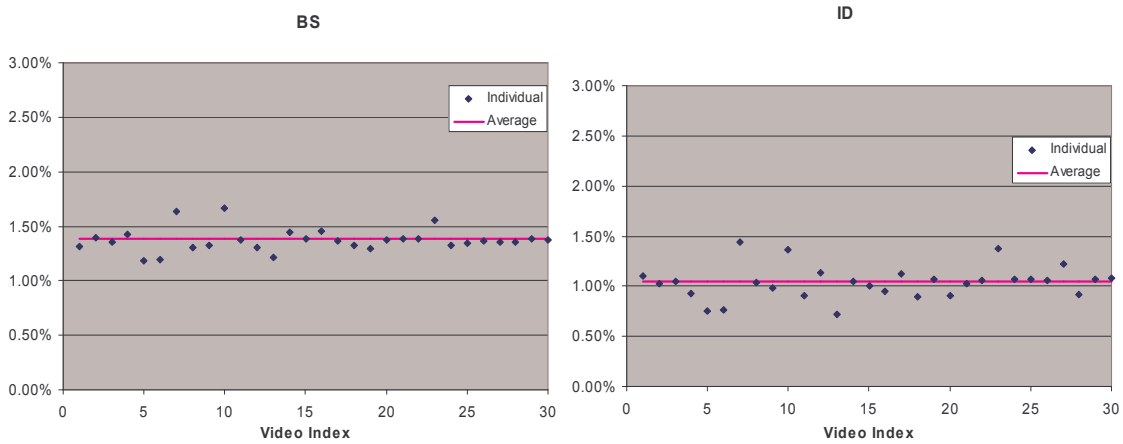
As opposed to these videos, our techniques generated the lowest true positive rate from videos 7, 9, 10, and 23. Having watched these videos, we found the backgrounds very unstable due to illumination changes. As shown in Figure 5.9 (b), three light sources indicated by the red boxes exposed directly to the camera which generates serious flickering effect. In addition, by going through each frame, we found the size of the ball appearing in these videos is quite small due to the long distance between the camera and the court. For example, video 7 was recorded by placing our camera at the end of the court and recording the event happened on the other side of the court. As a result, the distance between the tennis ball and the camera is quite large. In video 7, there are 709 occurrences of the ball. Among them, 227 frames (32%) have a size less than 10 pixels; 411 frames (58%) have a size between 10 to 20 pixels; 64 frames (9%) have a size between 20 to 30 pixels; 7 frames (1%) are greater than 30 pixels. Therefore, in a large number of frames, the ball size is out of the optimal detection range. Despite these videos generating the lowest true positive rate, the differences between them and others are not dramatic. Therefore, the influence of the external factors on the true positive rate is limited.



(a) (b)

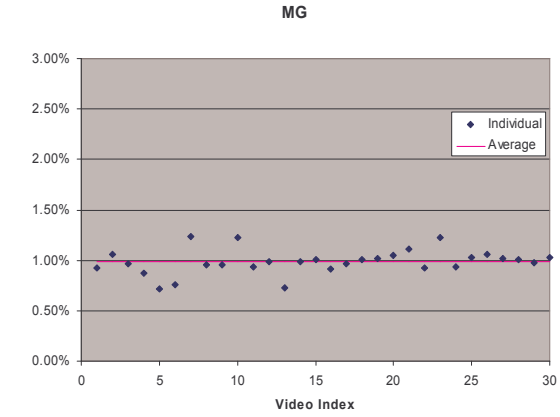
Figure 5.9: (a) A background image corresponding to video 13. (b) A background image corresponding to video 7. The lights are highlighted with the red boxes.

The False alarm rate generated from each video is illustrated in Figure 5.10. By analyzing Figure 5.10, we found our techniques all generated the lowest false alarm rate from videos 5, 6, and 13. Some videos, such as 7, 10, and 23, generated the highest false alarm rate regardless of the technique.

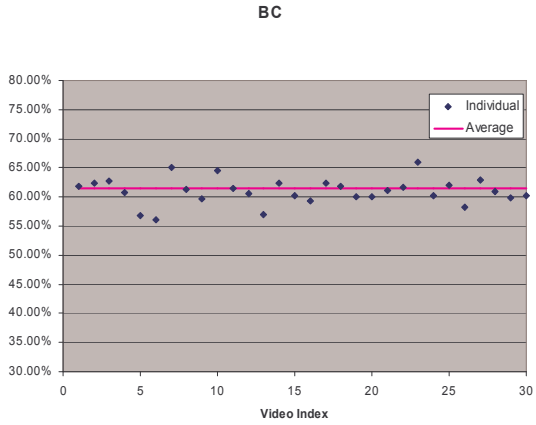


(a)

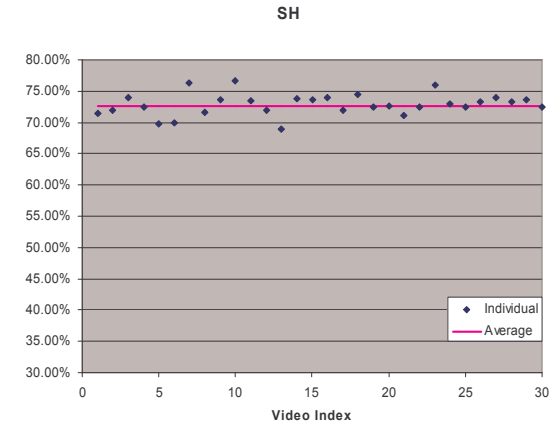
(b)



(c)



(d)

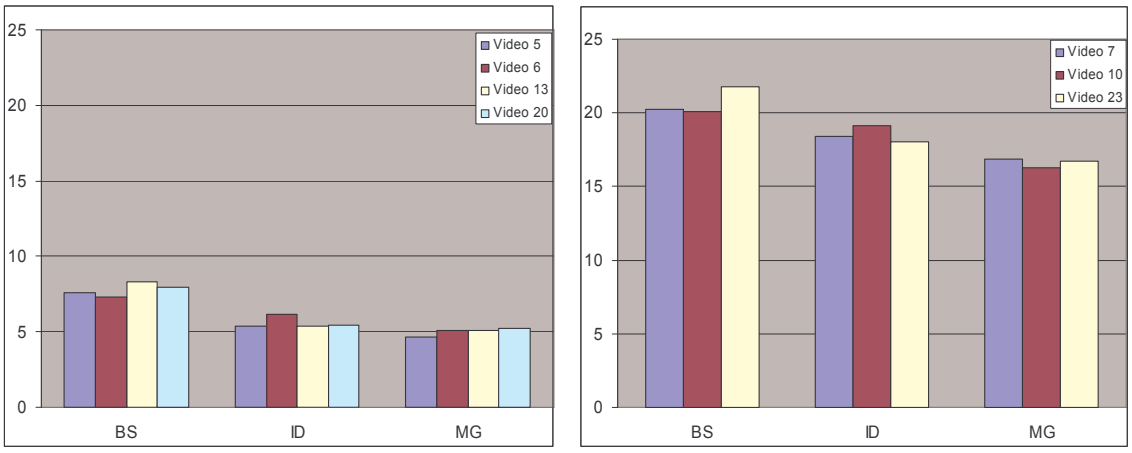


(e)

Figure 5.10: Indoors: false alarm rate generated from each video by different techniques.

As we described earlier, videos 5, 6, 13, and 20 were recorded in a stable lighting condition and less affected by light flickering. Therefore, the least amount of noise was contained in these videos. The possibility of false alarms for these videos was much

smaller than the others. As opposed to the videos listed above, the techniques produced the highest false alarm rate with videos 7, 10, and 23 because they are strongly affected by the flickering of light sources. This fact is illustrated in Figure 5.11.



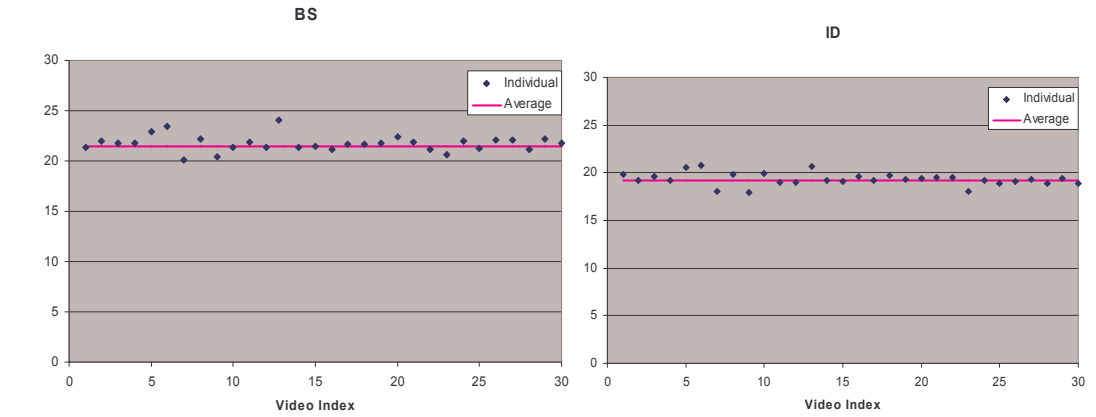
(a) (b)

Figure 5.11: The number of noise blobs/frame generated from different videos and techniques. (a) The least number of noise blobs were generated from videos 5, 6, 13, and 20. (b) The most number of noise blobs are generated from video 7, 10, and 23.

However, the difference between the lowest and highest false alarm rate is not significant. Therefore, the influence of the external factors on the false alarm rate is limited.

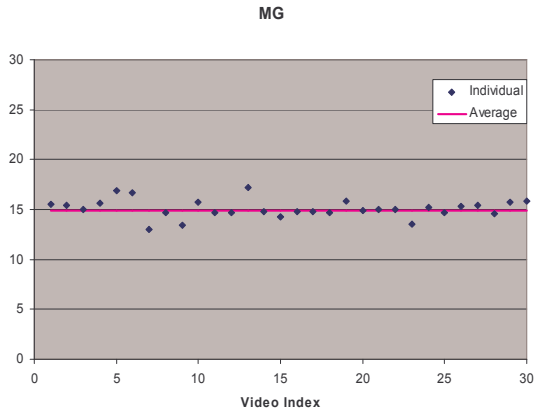
The processing speed depends on the number of image processing operations involved. If a video contains more moving objects or noise, these techniques will generate more tennis ball candidates after background subtraction. Consequently, more time is needed to classify these candidates as the player, the ball or noise.

As shown in Figure 5.12, videos 5, 6, and 13 have the fastest processing speed, while videos 7, 9, 23 have the slowest processing speed. As we described above, videos 5, 6, and 13 have the least amount of noise and therefore require the least amount of time to process video frames. On the other hand, videos 7, 9, 23 contain more noise than other videos. As a result, the processing speed slows down.

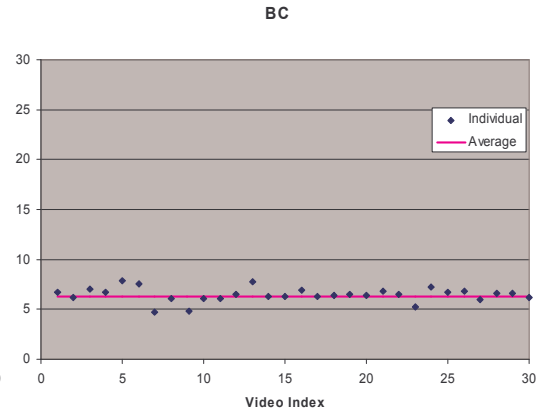


(a)

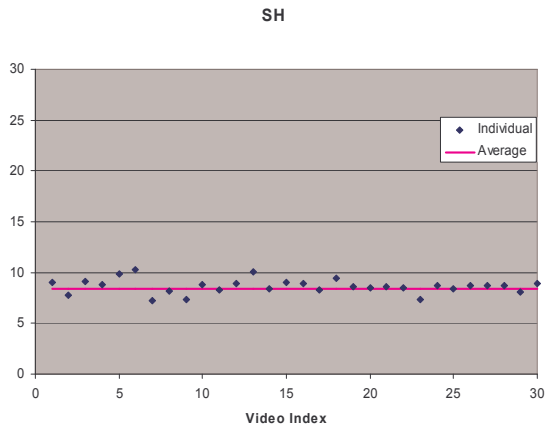
(b)



(c)



(d)



(e)

Figure 5.12: Indoors: frame rate generated from each video by different techniques.

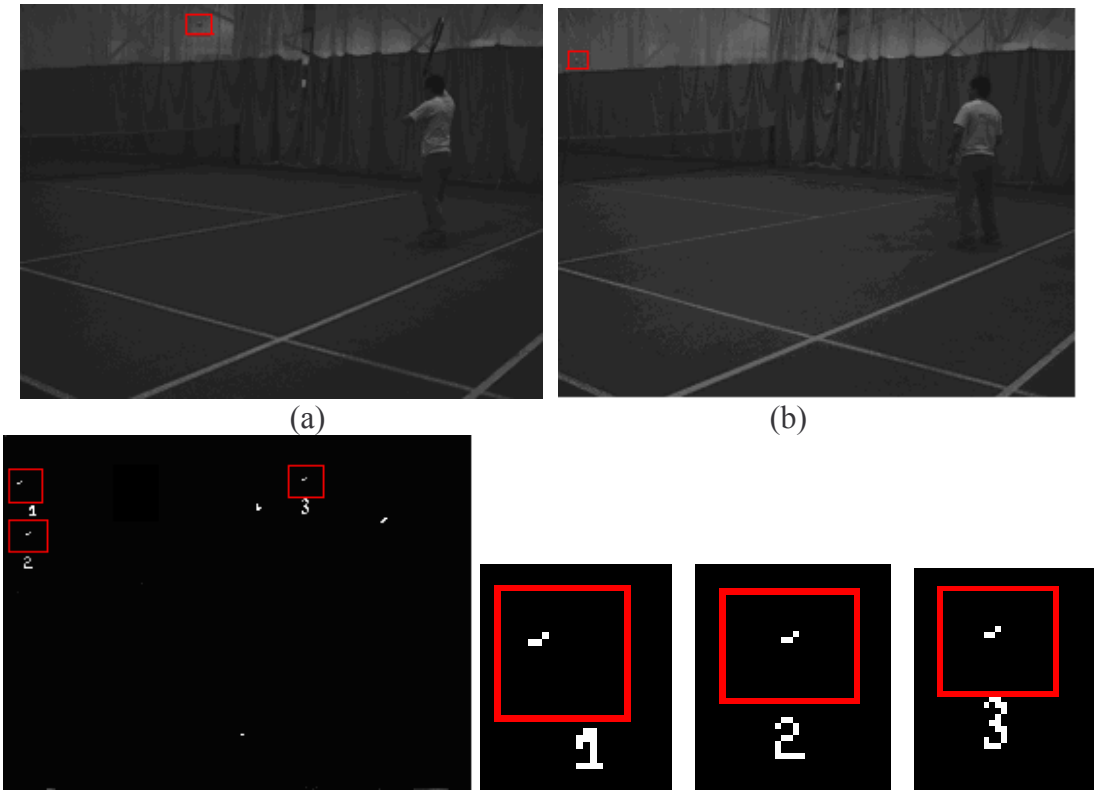
5.3.3 INDOORS RESULT DISCUSSION

For indoor environments, the overall performance of our solution is improved compared with the original approaches described in chapter 3. They achieve both higher

detection accuracy and faster processing speed. As discussed earlier, the performance of our techniques depend mainly on the robustness of the background model. As a result, the MG technique generates the highest detection accuracy and the slowest processing speed. The BS technique has the fastest processing speed and lowest detection accuracy. External factors such as lighting condition and image size of the tennis ball also affect the performance of our solutions. The influence of these factors on our solutions is also less than with our original approach.

Failures generated from videos with the worst accuracy (videos 7, 9, 10, 23) were collected and analyzed. We found the most common detection failures (54%) happened when the tennis ball appeared in regions with a similar gray value as the ball, i.e. the tennis ball adapts to the background. The visibility of the tennis ball is reduced due to the similarity between the tennis ball and the background (as shown in Figure 5.13 (a)) and is difficult to detect even with human eye. Another common cause of detection failures (38%), as illustrated in Figure 5.13 (b), was the size of the tennis ball changes rapidly when it appeared too far away from the camera. The small size of the tennis ball makes it indistinguishable from noise because their similar shape. For example, Figure 5.13 (c) shows the ball candidates generated from Figure 5.13 (b) and candidate 1 represents the tennis ball. However, candidates 2 and 3 also have the same shape as the tennis ball as illustrated in Figure 5.13 (d). In addition candidate 2 is close to the tennis ball resulting in the detection more difficult.

In addition, our techniques require a fair amount of time and various heuristics to set up proper parameter values. These parameters also need to be reset if the camera is relocated. The BS technique is based on seven parameters: the lower and upper thresholds of the tennis ball's size, the lower threshold of player's size, gray value differences between the background and the tennis ball, the aspect ratio, the compactness, and the roughness. Depending on the location of the camera and the background, these parameters are different for each video. For example, the size thresholds of the tennis ball are larger if the tennis is closer to the camera. The difference between the ball and the background is greater if the lighting condition is dim. The variation in parameter setting is quite large depending on the characteristics of the video. The sample parameter sets for two videos are illustrated in Table 5.2.



(a) (b) (c) (d)

Figure 5.13: Bad visibility due to the similarity and small size. (a) The ball is adapted to the background. (b) The size of the ball is small. (c) Ball candidates detected from (b). Candidate 1 represents the tennis ball. (d) The same shape between the tennis ball and candidates 2 and 3.

Table 5.2: The parameter list of the BS technique and associated values for different videos.

	Lower threshold of ball size	Upper threshold of ball size	Lower threshold of player size	Gray value difference between background & the tennis ball	Compact	Aspect ratio	Rough
Video 5	10	50	50	40	0.7	0.8	0.9
Video 26	15	60	60	45	0.7	0.8	0.8

If the location of our camera is fixed, these parameters can be pre-configured and used repeatedly without any modification. In addition, the value of these parameters can be the same for longer videos.

In addition, the ID technique requires another parameter: gray value differences between the current and previous frames. The value of this parameter varies depending on lighting conditions. For videos recorded in a good lighting condition, a larger value (more than 50) was selected because the tennis ball appears prominent. A smaller value (less than 20) was selected for videos recorded in a bad lighting condition. The average value of this parameter for all indoors videos was 38.

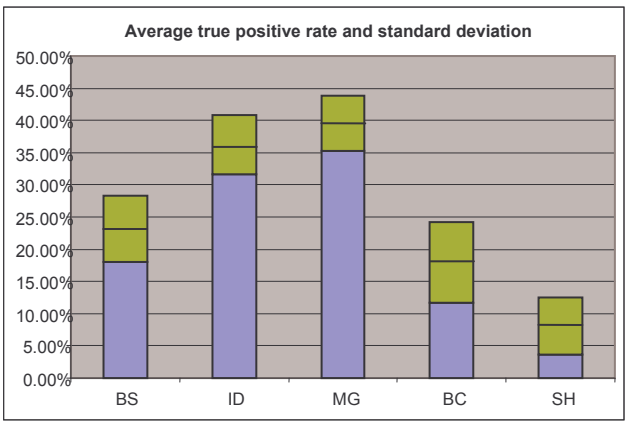
The parameters required by the MG technique include the learning factor with which the Gaussian distributions are updated and the number of Gaussian distributions used. Based on our experiments and the suggestion given by Stauffer et al. [61], the average learning factor is 0.3 and three Gaussian distributions are selected to balance the detection accuracy and efficiency. Selecting proper parameters requires lots of efforts and heuristics. In the future, we plan to develop a mechanism to automatically set up parameters according to different environments.

5.4 OUTDOORS RESULTS

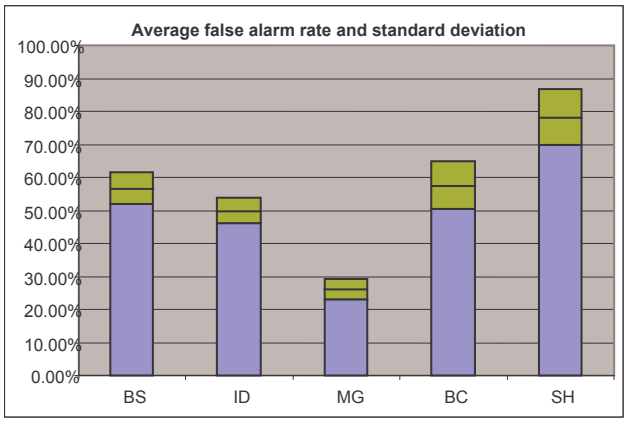
In this section, the results generated from the outdoors environment are systematically analyzed: Section 5.4.1 discusses the general performance of our techniques compared with the BC and SH techniques. Section 5.4.2 discusses the influence of the elements described in section 5.3.2. Finally, the limitations of our techniques are discussed in section 5.4.3.

5.4.1 RESULTS ANALYSIS

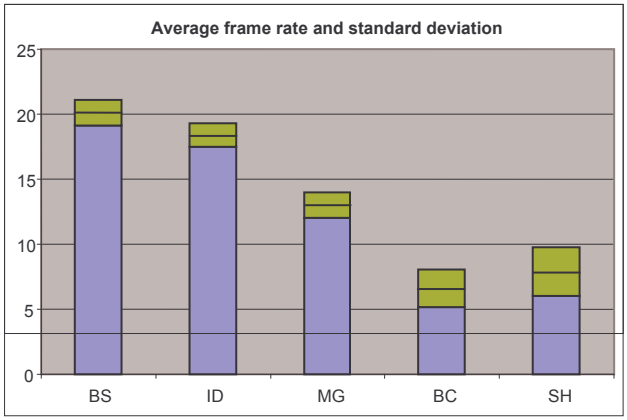
In the outdoors case, all techniques performed significantly worse than they did in the indoor environment. As shown in Figure 5.14, the average true positive rate of our techniques was less than 40%, about half that of the indoors case. The false alarm rate of our techniques was greater than 25%, more than ten times higher than the indoor case. The processing speed of our techniques was slightly less than the indoors case. In addition, our techniques achieved higher true positive rate and lower false alarm rate than the BC and SH techniques. However, the difference is not as impressive as the indoors case.



(a)



(b)



(c)

Figure 5.14: Average outdoors performance.

To further analyze the performance of our techniques, video 11 was selected from which we created a histogram image similar to the one described in section 5.3. Video 11 produced one of the best results for our techniques. As shown in Figure 5.15, the number of errors generated from our techniques was much bigger than the indoors case,

resulting less accuracy in terms of the true positive rate and the false alarm rate. In addition, our techniques generated errors more frequently than the indoors case. As we can see, the frequency of short term intervals from our techniques is quite big which means a large number of errors were generated during a short period of time. This demonstrates that our techniques lack robustness for such a complex environment.

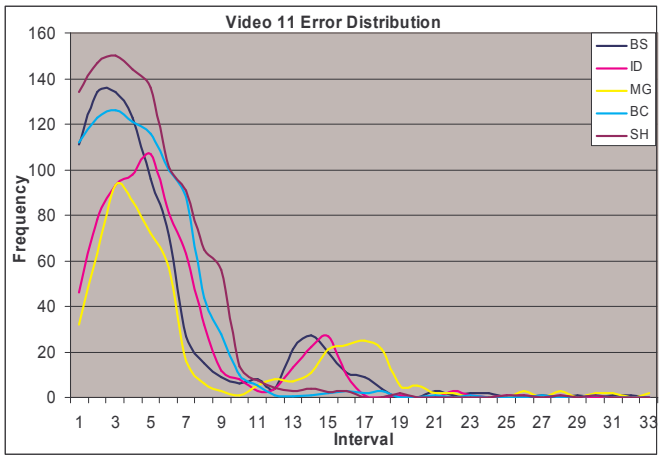


Figure 5.15: Error distribution for video 11.

5.4.2 ELEMENTS INFLUENCING THE TEST RESULTS

As discussed in section 5.3, the performance of our techniques strongly depends on the accuracy of the background models. The same pattern is demonstrated among our techniques, i.e. the MG technique generated the highest detection accuracy and the lowest processing speed. The BS technique generated the fastest processing speed with the lowest detection accuracy.

As illustrated in Figure 5.16 and Figure 5.17, the results generated from different outdoors videos are quite diverse compared with indoor videos due to the complexity and instability of the background environments. External factors, particularly illumination changes, have a significant impact on the performance of our techniques.

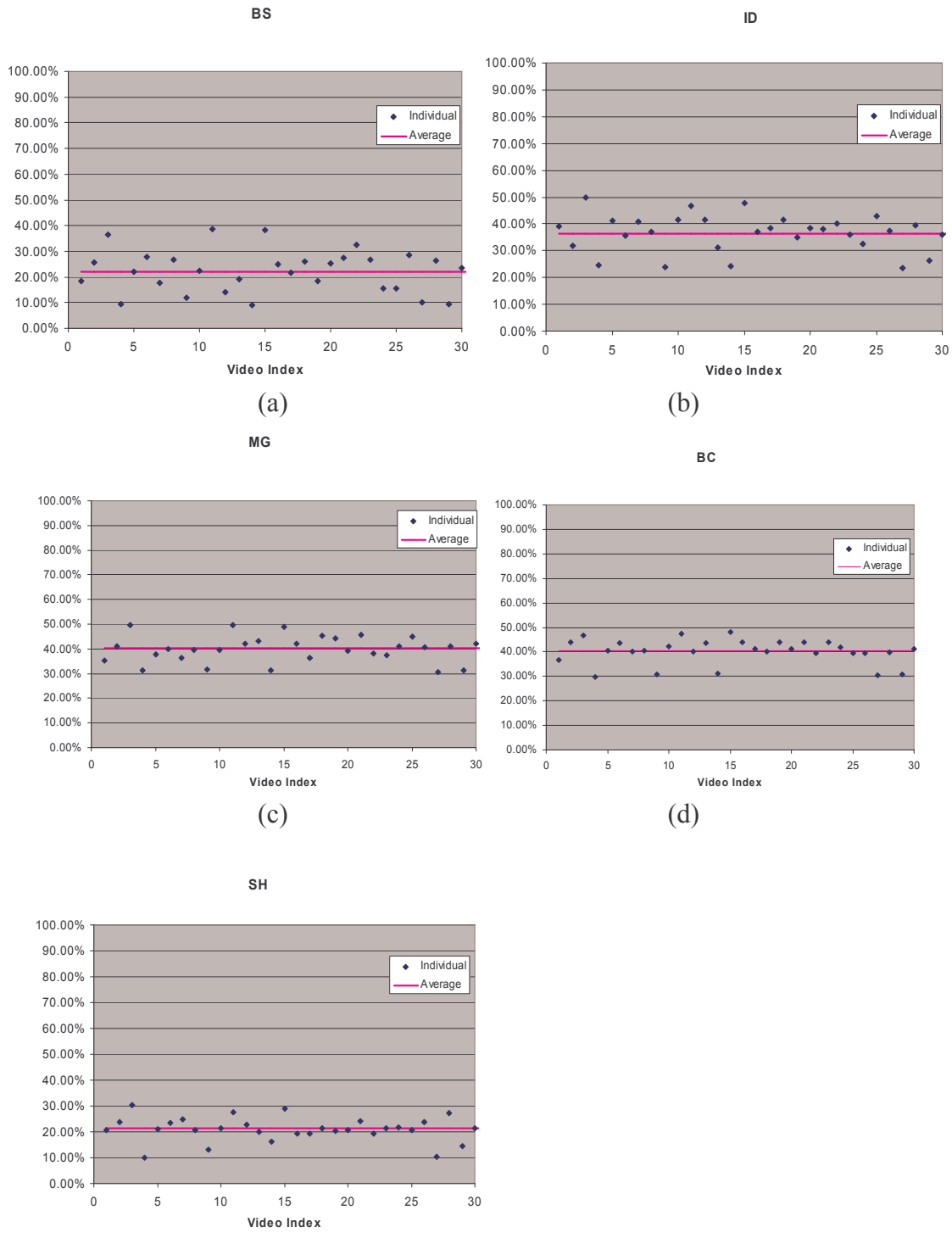


Figure 5.16: Outdoors: true positive rate generated from each video by different techniques.

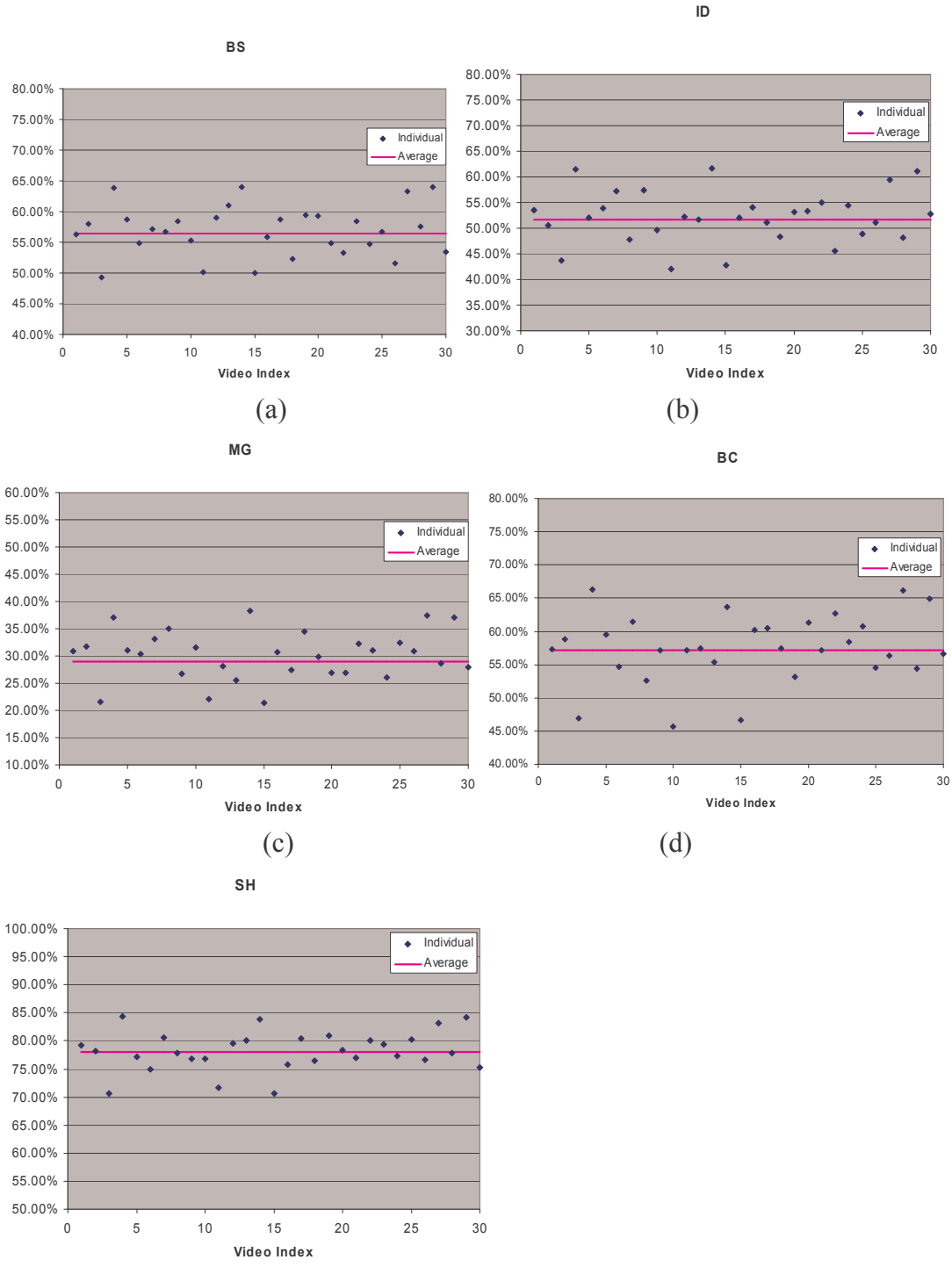


Figure 5.17: Outdoors: alarm rate generated from each video by different techniques

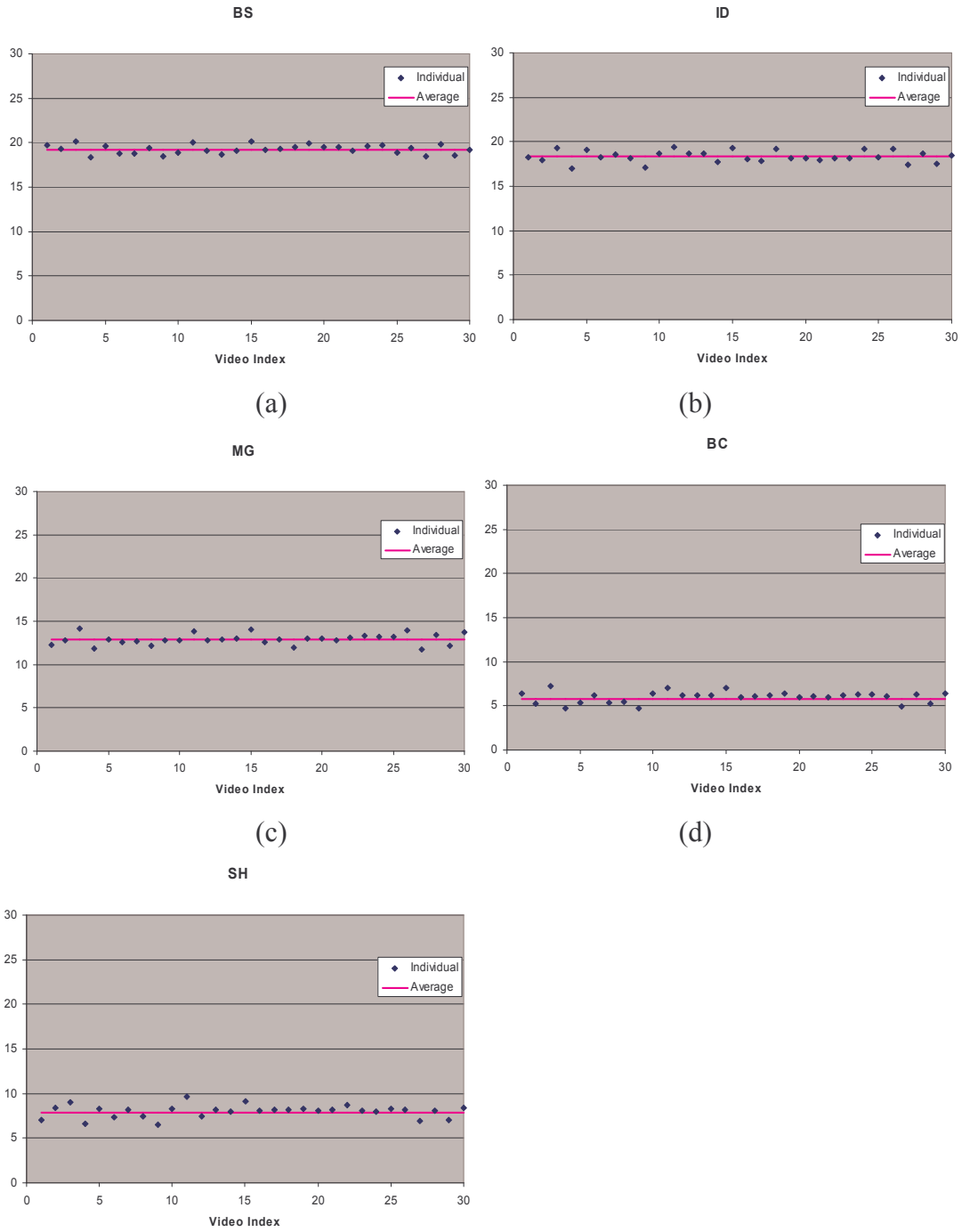


Figure 5.18: Outdoors: frame rate generated from each video by different techniques

5.4.3 OUTDOORS RESULT DISCUSSION

Our techniques are all based on the idea of background subtraction which depends strongly on the stability of the environment. If the environment changes constantly, there

is little connection between the background and the corresponding foreground images. As shown in Figure 5.19, the background in (b), particularly the clouds and the lighting, is quite different from the average background image in (a). Therefore, the subtraction between these two frames will generate an enormous amount of noise and non-target objects, making object segmentation difficult. The image processing techniques cannot solve this problem because there is too much noise similar to the tennis ball. The technique of adaptive background modeling using mixture of Gaussian distributions is capable of adapting to the environmental changes. But for this particular outdoor environment, the background changes so dramatically over a short period that this technique is not able to respond to the changes quickly enough. In the future, the performance of this technique needs to be assessed in various outdoors conditions.

(a) (b)

Figure 5.19: Complex outdoors environment.

As we found out: the environment of real outdoors matches, such as the Wimbledon as shown in Figure 5.20, is not as complicated as ours since their cameras mainly focus on the tennis ball. The problems of clouds, trees, and the wind which are the main sources of noise do not present in their videos. Therefore, their background condition is stable and close to our indoors case. Given this similarity, we believe our techniques can generate results similar to our indoors case if our tracking system operate in their environment.

Figure 5.20: Sample images taken from the Wimbledon tennis matches.

To further evaluate our techniques, we used a DVD quality video (about 30 minutes) recorded from a Wimbledon tennis match. Since there are no background images available for this video and the moving camera, our techniques can not be applied to this video. Instead, we iterated through each frame and manually detected the tennis ball. Among these frames (about 32400 frames), 92% of the tennis ball can be detected directly by the human eye. The other 8% (see Figure 5.21) are difficult to recognize due to blurring or other reasons. The similar result was achieved in our indoors experiments which generated about a 90% of detection accuracy.

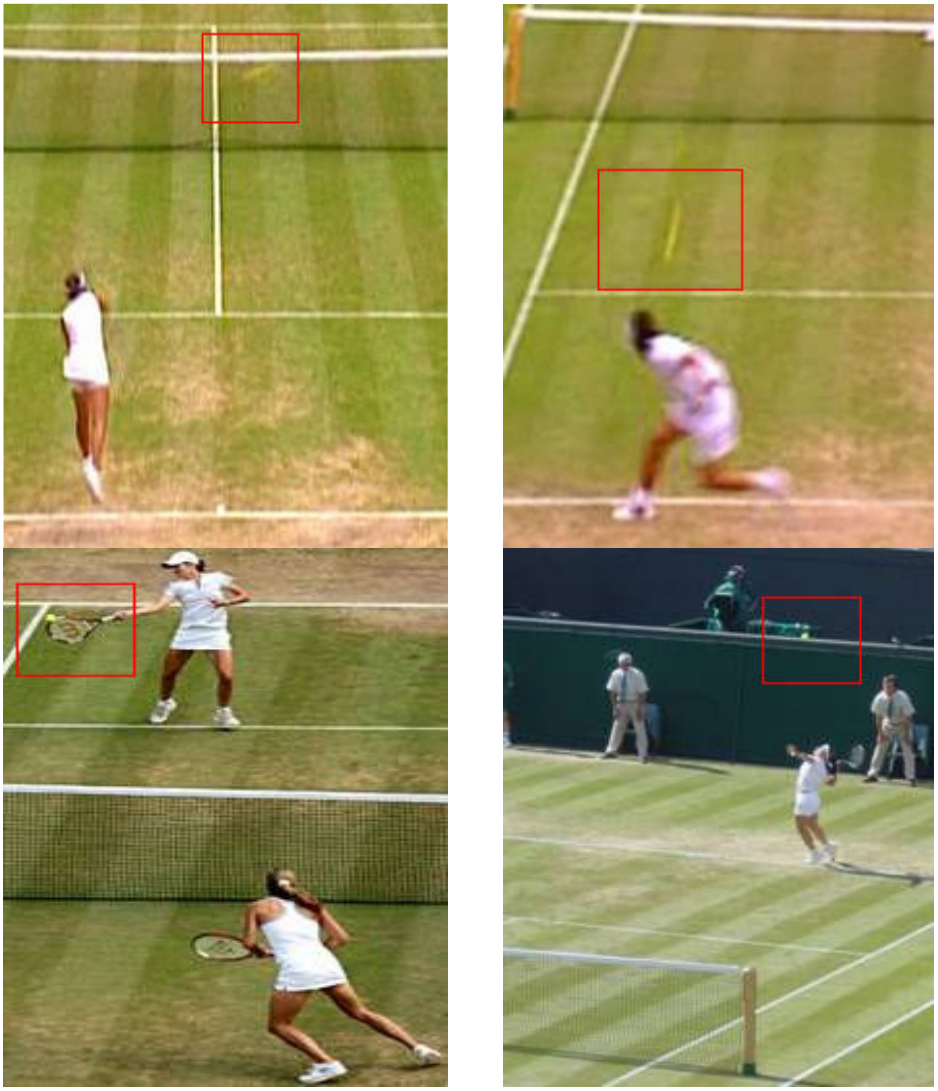


Figure 5.21: Bad visibility of the tennis ball. Images were taken from the Wimbledon tennis matches.

5.5 DEPLOYMENT

Many real-world applications such as a tennis training system can take advantage of our techniques. A tennis training system helps players to improve their performance by analyzing their gesture and the tennis ball's trajectory. The trajectory information can be generated from our techniques: first, several cameras, which monitor the tennis ball from different point of views, are placed in a training filed. The locations of the tennis ball appearing in each camera are detected separately by our techniques. These detected locations are then synchronized. Finally, the 3D trajectory is constructed from the synchronized locations. In addition, the body of the player can be constructed using the

technique described by Theobalt et al. [65] which resembles the human body from the color tags attached to the player. Once the model of the player and the 3D trajectory of the ball are created, they can be replayed in a virtual environment where the problems of the player's gestures are identified and possible suggestions can also be provided.

The field layout of this training system is illustrated in Figure 5.22. In order to achieve the best result, this system should operate in an indoors environment or an outdoors environment with a stable background. Four industrial cameras similar to the one we used are placed along the two sides of the court. Each pair of cameras watches a half of the court. To reduce the blurring effect, higher shutter speed should be selected. Extra light sources can be added to the field to compensate the dim illumination due to fast shutter speed. To reduce the flickering effect, lights should not be placed at positions exposing to the cameras directly.

The motion information of the tennis ball can be generated by our techniques. Four cameras produce four streams of image data. Each of them needs to be processed separately by one instance of the tracking technique. We will use the BS technique to track the tennis ball since it is the most efficient and has good detection accuracy. First, an average background image is created before starting the training session. Ball candidates are detected by background subtraction. The player is located so that candidates within the area of the player are removed. Remaining candidates are further verified based on shape and the dynamics information. The final result is detected as the tennis ball. To have the BS technique work properly, the parameter list of the BS technique needs to be tested and configured properly before the training system starts to operate.

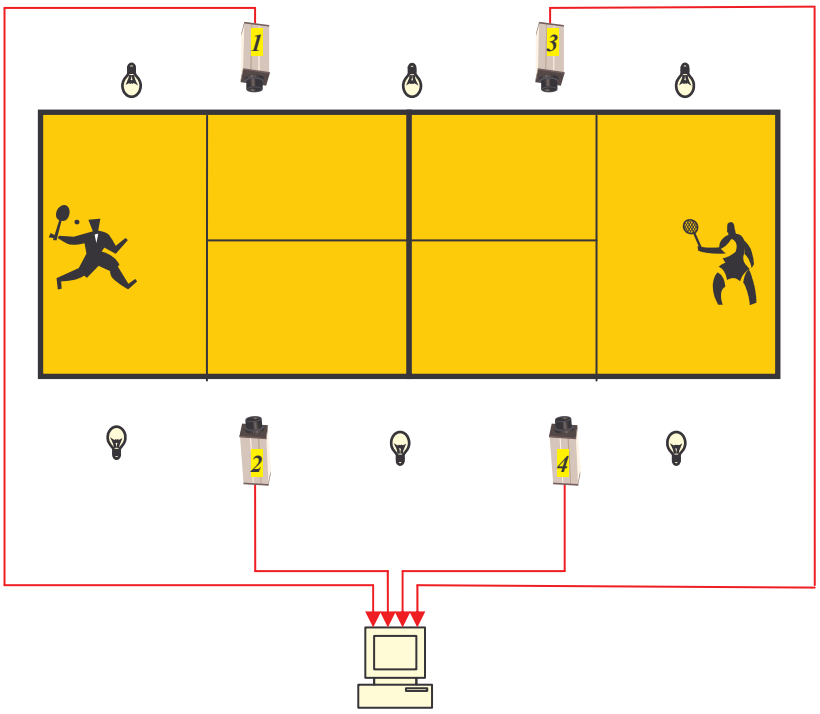


Figure 5.22: Field layout of a tennis training system.

However, having four instances of the BS technique running simultaneously is time-consuming. We could have each instance of the BS technique running on a different computer, but the system cost is increased. Given the fact that the tennis ball can only appear on the one side of the court, we optimize the tracking system by enabling only two instances of the BS which currently watch the tennis ball and suspend the others. Once the tracked information indicates that the tennis ball is going to move to another side of the court, the suspended instances of the BS are activated while the others are suspended. For instance in Figure 5.22, when the tennis ball is on the left side of the court, the two instances of the BS associated with video 1 and 2 are active while the others are suspended. Once the ball is about to move to the right side of the court, the instance of the BS associated with camera 3 and 4 resume and the others are suspended. As a result, the computational cost is reduced by 50%. To further increase the processing speed, we can apply the optimization strategy introduced in section 3.6 to each tracking instance: the object detection is performed in a region near the predicted location generated from the Kalman filter. Therefore, the amount of image data need to be processed is reduced and processing speed is increased.

5.6 CONCLUSION

In this chapter, we evaluated the performance of our techniques by applying them to a large number of videos recorded in a real tennis environment. The solutions were also compared to two other techniques. According to test results, the overall performance of our techniques is superior to the BC and SH in terms of the accuracy and efficiency. In an indoors environment our techniques were able to track the tennis ball with much higher accuracy than the compared techniques. Based on the analysis, we found the performance of our techniques is strongly affected by the robustness of the background model. As a result, the MG technique achieves the highest detection accuracy, while the BS technique achieves the lowest detection accuracy. In addition, the test results are also affected by external factors such as the stability of the background environment and the image size of the tennis ball. As a result, the outdoor results were worse than indoors because of the instability of the outdoors environment due to various reasons such as clouds, wind, and trees.

CHAPTER 6

CONCLUSION AND FUTURE WORK

Tracking a tennis ball during a real tennis match is challenging due to its high speed, small image size of the ball, and the requirement of fast processing speed. Most of the techniques currently available are not suitable for solving this problem. These techniques normally track slow moving objects, such as the human body. The size of the objects they track is usually much bigger than the tennis ball. For small objects such as a tennis ball, the feature space provided in such a small region is often too limited to describe the object. In addition, the processing speed is strongly affected when higher resolution and frame rate cameras are used.

In this thesis, we first presented the solution of background subtraction with color/shape segmentation. We compared it with the technique of Haar classifiers. Despite the existing limitations such as the strict requirement of proper lighting and slow motion, the result showed that background subtraction is a feasible solution for object tracking. Given this promising result, we then extended this approach and developed three more robust techniques which are also based on background subtraction. They are:

Background subtraction with verification: First a background model is created. Ball candidates are detected by background subtraction. The area of the player is located and any candidates residing within the area of the player are removed. Remaining candidates are further inspected based on shape and the dynamics information. Finally the remaining candidates are detected as the ball.

Image differencing between the current and previous images: First a background model is created which represents the value of each pixel as a single Gaussian distribution. Ball candidates are detected by image differencing between the current and previous frames. Candidates from the previous frame are removed by background subtraction. Once ball candidates in the current image are found, the tennis ball is detected in the same way as our first solution.

Adaptive background modeling using a mixture of Gaussians: This technique improves the first two techniques using a more sophisticated and accurate background model. Instead of using a single Gaussian, this technique uses a mixture of Gaussians to model the value of each background pixel. In addition, changes to the environment can be gradually adopted into the existing background model which makes this technique more dynamic and flexible.

To test the robustness and reliability of our techniques, several videos were recorded using the high speed and high resolution camera in both an indoors and outdoors tennis court. Overall the performance of our techniques is superior to the BC and SH techniques. In the case of the indoors court, our techniques were able to achieve about a 90% true positive rate which is about three times higher than the compared techniques, a less than 2% false alarm rate which is about twenty times lower than the compared techniques, and a frame rate of 20 fps which is about two times faster than the compared techniques. In addition, the performance of our techniques is more stable since the true positive rate and false positive rate were less fluctuated than that of the compared techniques. In the case of the outdoors court, the performance of our techniques was dramatically reduced. The average true positive rate was less than 40% and the false alarm rate was greater than 25%. The processing speed was slightly slower than the indoors case.

Based on the analysis, we found that the results of our techniques are affected by the accuracy of the background model. Given a more robust and accurate background model, our techniques can achieve better results. On the other hand, techniques using a more robust background model operate slower since more image process operations are involved. The technique of BS generates the lowest detection accuracy and the fastest processing speed. The background model used by the BS technique is a naive average of the image frames which is simple but lacks accuracy. As a result, the BS technique has the lowest positive rate (average 87.4%), the highest false alarm rate (average 1.35%), and the fastest processing speed (average 21 fps) among our techniques. The ID technique models the background using a single Gaussian distribution pixel-wise. It is more accurate than the simple averaging but requires more time to develop the background model and detect foreground objects. Consequently, it has a higher positive

rate (average 89.64%), a lower false alarm rate (average 1.07%), but a slower processing speed (average 19 fps). The MG technique models the background using multiple Gaussian distributions and is the most accurate and robust among our techniques. It updates the current background model by adjusting the existing distributions in response to environmental changes. However, its computation cost is the highest of our techniques. This technique has the highest positive rate (average 90.73%), the lowest false alarm rate (average 1.02%), and the slowest processing speed (average 14 fps).

In addition, the performance of our techniques is also affected by external factors such as the stability of the background environment and the tennis ball's image size. The impact these external factors have on the indoors results is very limited. However, the impact they have on the outdoors results is much more significant due to the complexity and instability of outdoors environments. Therefore, the results achieved from the outdoors court were not as sound as the indoors case. However, by analyzing real outdoors tennis matches such as the Wimbledon, we found their background is not as complicated as ours since their cameras focus on the tennis ball. We believe our techniques can achieve much better result in their environments. We do not have the background images of their tennis court. Therefore, we are not able to test our solutions in their environment.

Despite the existing limitations, our techniques are able to track a tennis ball with very high accuracy and fast speed which can not be achieved by most tracking techniques currently available. We are confident that the motion information generated from our techniques is reliable and accurate. Giving this promising result, we believe some real-world applications, such as tennis training and tactics systems, can benefit from our techniques.

In this thesis we analyzed various object detection and tracking techniques. Their strengths and limitations were also discussed. We then implemented some techniques and analyzed their suitability for tennis tracking problems. We found that background subtraction is a feasible solution for real-time object tracking due to its simplicity and good performance. We also discovered several tennis ball detection techniques, such as edge images, object color, dynamic information, and shape information. With this information, we developed solutions which are based on background subtraction and

various tennis ball detection techniques. Based on our experience, we found that for large objects such as a human body, most object-based techniques are able to detect and track the target objects as they can be recognized easily. For small and fast moving object such as the tennis ball, the shape information is often too limited and inaccurate. Therefore, object-based techniques alone are not able to solve the entire problem. A more reliable approach should take advantage of both the object-based and non-object based techniques. If the speed is crucial, techniques should be avoided which involve a lot of computations such as the Hough transform and Haar classifiers. In addition, an optimization strategy can also be developed to further increase the processing speed.

There are some issues remaining for future work. Currently, parameters associated with these techniques need to be set up manually. They need to be reset when the camera is moved to a different location. In the future, a mechanism which can automatically configure these parameters will be constructed. To compromise with the complex and unstable environments, a more robust technique needs to be developed which is able to track the ball in both indoors and outdoors environments. The Kalman filter does not predict system states very accurately. In the future, we hope to make improvements in the state estimation in order to reduce the chances of needing to search the entire image. Another avenue for future work involves discriminating between ball and non-ball objects based on object trajectories.

LIST OF REFERENCES

[1] Amini, A.A., Tehrani, S., and Weymouth, T.E. (1988). Using Dynamic Programming for Minimizing the Energy of Active Contours in the Presence of Hard Constraints. In Proceedings Second International Conference Computer Vision, pp. 95-99.

[2] Assfalg, J., Bertini, M., Colombo, C., and Del, B. (2002). Semantic Annotation of Sports Video. In Proceedings of IEEE Multi-Media, vol. 9, no. 2, pp. 52-60.

[3] Babaguchi, N., Kawai, Y., and Kitahashi, T. (2002). Event Based Video Indexing by Inter-Modal Collaboration. In Proceedings of IEEE Transactions on Multimedia, vol. 4, no. 1, pp. 68–75.

[4] Bajcsy, R., and Kovacic, S. (1989). Multiresolution elastic matching. In Computer Vision, Graphics and Image Processing, vol. 46, pp. 1-21.

[5] Barron, J., Fleet, D., and Beauchemin, S. (1994). Performance of Optical Flow Techniques. In Proceedings of International Journal of Computer Vision, vol. 12, no. 1, pp. 42 – 77.

[6] Bennet, N., Burridge, R., and Saito, N. (1999). A Method to Detect and Characterize Ellipses Using the Hough Transform. In IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 7, pp. 652 – 657.

[7] Bouguet, J. (1999). Pyramidal implementation of the Lucas Kanade Feature Tracker Description of the Algorithms. OpenCV Documentation, Micro-Processor Research Labs, Intel Corporation.

[8] Boyle, R.D., and Thomas, R.C. (1998). Computer Vision: A First Course. Blackwell Scientific Publications, pp 32 - 34.

[9] Bradski, G. (1998). Computer Vision Face Tracking as a Component of a Perceptual User Interface. In Proceedings of Workshop Applications Computer Vision, pp. 214--219.

[10] Cai, Q., and Aggarwal, J. K. (1996). Tracking Human Motion Using Mutiple Camers. In Proceedings of International Conference on Pattern Recognition, pp 68 – 72.

[11] Canny, J. (1986). A Computational Approach to Edge Detection. In Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, pp. 679-698.

[12] Choi, S., Seo, Y., Kim, H., and Hong, K. S. (1997). Where are the Ball and the Players?. In Proceedings of International Conference of Image Analysis and Processing, pp 196–203.

[13] Comaniciu, D., and Meer, P. (2002). Mean shift: A Robust Approach toward Feature Space Analysis. In Proceedings of PAMI, vol. 24, no. 5, pp 603-619.

[14] Cutler, R., and Davis, L. (1998). View-Based Detection and Analysis of Periodic Motion. In Proceedings of International Conference on Pattern Recognition, pp. 495 - 500.

[15] Das, M., and Anand, J. (1995). Robust Edge Detection in Noisy Images Using an Adaptive Stochastic Gradient Technique. In Proceedings of the 1995 International Conference on Image Processing (ICIP'95).
[16] D'Orazio, T., Ancona, N., Cicirelli, G., and Nitti, M. (2002). A Ball Detection Algorithm for Real Soccer Image Sequences. In Proceedings of the 16th International Conference on Pattern Recognition (Icpr'02), Vol.ume 1 - Vol.ume 1. ICPR. IEEE Computer Society.
[17] Duda, R.O., and Hart, P.E. (1972). Use of Hough Transform to Detect Lines and Curves in Pictures. In Proceedings of Communication of the ACM, vol. 15, pp. 11–15.
[18] Eaton, R., and Scassellati, B. (2000). ViSIT: Visual Surveillance and Interaction Tracking. Social Robotics Laboratory Yale University New Haven, CT 06511.
[19] Faugeras, O. D. (1993). Three-Dimensional Computer Vision: A Geometric Viewpoint. In MIT Press.
[20] Friedland, N., and Adam, D. (1989). Automatic Ventricular Cavity Boundary Detection from Sequential Ultrasound Images using Simulated Annealing. In Proceedings of IEEE Transactions on Medical Imaging, vol. 8, pp. 344-353.
[21] Friedman, N., and Russell, S. (1997). Image Segmentation in Video Sequences: A Probabilistic Approach. In Proceedings of Uncertainty of Artificial Intelligence.
[22] Gee, J.C., Reivich, M., and Bajcsy, R. (1993). Elastically Deforming 3D Atlas to Match Anatomical Brain Images. In Proccedings of Journal of Computer Assisted Tomography, vol. 17, pp. 225-236.
[23] Gerald, B, F. (1995). Introduction to partial differential equations. Princeton University Press.
[24] Grimson, W.E.L., Stauffer, C., Romano, R., and Lee, L. (1998). Using Adaptive Tracking to Classify and Monitor Activities in a Site. In Proceedings of Computer Vision and Pattern Recognition, pp. 1-8.
[25] Gueziec, A. (2002). Tracking Pitches for Broadcast Television. In Proceedings of IEEE Computer, vol. 35, no. 3, pp. 38-43.
[26] Han, M., Hua, W., Xu, W., and Gong, YH. (2002). An Integrated Baseball Digest System Using Maximum Entropy Method. In Proceedings of ACMM 2002, pp 347-350.
[27] Haritaoglu, I.D., Harwood, D., and Davis, L. (1998). W4 – a Real Time System for Detecting and Tracking People and Their Parts. In Proceedings of the European Conference on Computer Vision.
[28] Haritaoglu, I.D, Harwood, D., and Davis, L. (2000). W4: Real-Time Surveillance of People and Their Activities. In IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, issue 8, pp 809 – 830.
[29] Hawk-Eye. (2003). Available from http://news.bbc.co.uk/sport1/hi/tennis/2977068.stm.
[30] Illingworth, J., and Kittler, J. (1988). A Survey of the Hough Transform. In Computer, Vision, Graphics, Image Processing, vol. 44, pp. 87 – 116.
[31] INTEL. (2002). Open Source Computer Vision Library. Available from http://www.sourceforge.net/projects/opencvlibrary.
[32] Intille, S.S., Davis, J.W., and Bobick, A.F. (1997). Real-Time Closed-World Tracking. In Proceedings of the IEEE Computer Society Conf. on CVPR, pp. 697-703.

[33] Ishii, I., and Ishikawa, M. (1999). Self Windowing for High-Speed Vision. In Proceedings of Systems and Computers in Japan, vol. 32, no. 10.
[34] Ishikawa, M. (1998). Super-Parallel Super-High-Speed Visual Information System: General-Purpose Vision Chip and Grayscale Photoelectron Vision System. In Proceedings of Applied Physics, vol. 67, pp. 33-38.
[35] Kalman, R.E. (1960). A New Approach to Linear Filtering and Prediction Problems. In Transactions of the ASME - Journal of Basic Engineering, vol. 82: pp. 35-45.
[36] Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. In Proceedings of International Journal of Computer Vision, vol. 1, pp. 321-331.
[37] KawTraKulPong, P., and Bowden, R. (2001). An Improved Adaptive Background Mixture Model for Real-Time Racking with Shadow Detection. In Proceedings of Second European Workshop on Advanced Video-Based Surveillance Systems.
[38] Klette, R.H., Stiehl, S., Viergever, M.A., and Vincken, K.L. (2000). Performance Characterization in Computer Vision. London: Kluwer.
[39] Lam, M., Chan, M., Leung, J., Wong, R., Hang C., and Jin, J.S. (2003). Computer-Assisted off-side Detection in Soccer Matches, In Proceedings of Technical Report, School of Information Technologies, University of Sydney.
[40] Lienhart, R., and Maydt, J. (2002). An Extended Set of Haar-like Features for Rapid Object Detection. In Proceedings of ICIP, pp. 900-903.
[41] Liyuan, L., Huang, W., Gu, T., Qi, T. (2002). Foreground Object Detection in Changing Background Based on Color Co-Occurrence Statistics. In Proceedings of Sixth IEEE Workshop on Applications of Computer Vision, pp 269 – 274.
[42] Lucas, B.D., and Kanade, T. (1981). An iterative Image Registration Technique with an Application to Stereo Vision, Proceedings 7th Joint Conference on Artificial Intelligence, pp. 674-679.
[43] Maxwell, B., and Shafer, S. (1994). A Framework for Segmentation Using Physical Models of Image Formation, In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition.
[44] McIvor, A. (2000). Background Subtraction Techniques. In Proceedings of Image and Vision Computing. Available from http://www.mcs.csuhayward.edu/~tebo/Classes/6825/ivcnz00.pdf.
[45] Miller, M.I., Christensen, G.E., Amit, Y., and Grenander, U. (1993). Mathematical Textbook of Deformable Neuroanatomies. In Proceedings of the National Academy of Sciences, USA, vol. 90, pp. 11944-11948.
[46] Müller, B., and Anido, R. D. (2004). Distributed Real-Time Soccer Tracking. In Proceedings of the ACM 2nd International Workshop on Video Surveillance &Amp; Sensor Networks, VSSN '04. ACM Press, pp 97-103. Available from http://doi.acm.org/10.1145/1026799.1026816.
[47] Nalwa, V. (1993). A Guided Tour of Computer Vision. Reading. MA: Addison Wesley
[48] Ohta, Y. (1980). A Region-Oriented Image-Analysis System by Computer. Doctoral Dissertation, (Kyoto University, Japan).
[49] Papageorgiou, C., Oren, M., and Poggio, T. (1998). A General Framework for Object Detection. In International Conference on Computer Vision.

[50] Pers, J., Vuckovic, G., Kovacic, S., and Dezman, B. (2001). A Low Cost Real-Time Tracker of Live Sport Events. In Proceedings of 2nd International Symposium on Image and Signal Processing and Analysis, pp. 362 – 365.
[51] Pingall, G., Jean, Y., and Carlbom, I. (1998). Real-Time Tracking for Enhanced Tennis Broadcasts. In Proceedings of CVRP, pp. 260–265.
[52] Pingali, G., Opalach, A., and Jean, Y. (2000). Ball Tracking and Virtual Replays for Innovative Tennis Broadcasts. In Proceedings of the International Conference on Pattern Recognition (Icpr'00)-Vol.ume 4 - Vol.ume 4 (September 03 - 08, 2000). ICPR. IEEE Computer Society.
[53] QUESTEC. 2003. Available from http://www.questec.com/q2001/news/1999/030599.htm.
[54] Ren, J.C., Orwell, J., Graeme, A., and Xu, M. (2004). A General Framework for 3d Soccer Ball Estimation and Tracking. In Proceedings of ICIP, pp. III: 1935-1938.
[55] Ridder, C., Munkelt, O., and Kirchner, H. (1995). Adaptive Background Estimation and Foreground Detection using Kalman-Filtering. In Proceedings of International Conference on recent Advances in Mechatronics, ICRAM'95, UNESCO Chair on Mechatronics, pp 193–199.
[56] Ristic, B., Arulampalm, S., and Gordon, N. (2004). Beyond the Kalman filter: Particle Filters for Tracking Applications. Artech Houses.
[57] Roberts, L, G. (1965). Machine Perception of Three-Dimensional Solids. In Proceedings of Optical and Electro-Optical Information Processing. MIT Press, pp 159-197.
[58] Schachter, B., Davis, L., and Rosenfeld, A. (1979). Some Experiments in Image Segmentation by Clustering of Local Feature Value. In Proceedings of Pattern Recognition, vol. 11, no. 1, pp 19-28.
[59] Sonka, M., Hlavac, V., and Boyle, R. (1999). Image Processing, Analysis, and Machine Vision. Chap. 5. pp. 163-164.
[60] Staib, L.H., and Duncan, J. S. (1992). Boundary Finding with Parametrically Deformable Models. In Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, pp. 1061-1075.
[61] Stauffer, C., and Grimson, W. E. L. (1999). Adaptive Background Mixture Models for Real-Time Tracking. In Proceedings of Computer Vision Pattern Recognition, pp. 246-252, Ft. Collins, CO.
[62] Storvik, G. (1992). A Bayesian Approach to Dynamic Contours. In Proceedings of Technical Report Report No. 860, Norwegian Computing Center, Oslo. Thesis for the Dr.Scient degree at the University of Oslo.
[63] Storvik, G. (1994). A Bayesian Approach to Dynamic Contours Through Stochastic Sampling and Simulated Annealing. In Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, pp. 976-986.
[64] Sudhir, G., Lee, J.C.M., Jain, A.K. (1998). Automatic Classification of Tennis Video for High-Level Content-Based Retrieval. In Proceedings of International. Workshop on Content-Based Access of Image and Video Databases CAIVD'98, pp. 81-90.
[65] Theobalt, C., Albrecht, I., Haber, J., Magnor, M., and Seidel, H. (2004). Pitching a Baseball: Tracking High-Speed Motion with Multi-Exposure Images. In

Proceedings of ACM Trans. Graph. 23, 3, pp. 540-547. Available from http://doi.acm.org/10.1145/1015706.1015758.

[66] Toyama, K., Krumm, J., Brumitt, B., and Meyers, B. (1999). Wallflower: Principles and Practice of Background Maintenance. In Proceedings of IEEE International Conference on Computer Vision, pp. 255 – 261.

[67] Treptow, A., Masselli, A., and Zell, A. (2003). Real-Time Object Tracking for Soccer-Robots without Color Information. In Proceedings of the European Conference on Mobile Robots (ECMR).

[68] Viola, P., and Jones, M.J. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. In Proceedings of CVPR, pp. 511-518.

[69] Viola, P., and Jones, M.J. (2001). Robust Real-time Object Detection. In Proceedings of the Second International Workshop on Statistical and Computational Theories of Vision – Modeling, Learning, Computing and Sampling.

[70] Wang, J.R., and Parameswaran, N. (1999). Survey of Sports Video Analysis: Research Issues and Applications. Available from http://crpit.com/confpapers/CRPITV36Wang.pdf.

[71] Welchand, G., and Bishop, G. (1988). An Introduction to the Kalman filter. In Siggraph. Available from http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf.

[72] Williams, D. J., and Shah, M. (1992). A Fast Algorithm for Active Contours and Curvature Estimation. In Proceedings of CVGIP: Image Understanding, vol. 55, No. 1, pp. 14-26. Available from http://www.cs.ucf.edu/~vision/papers/shah/92/WIS92A.pdf.

[73] Wren, C. R., Azarbayejani, A., Darrell, T,. and Pentland, A. (1999). Real-time Tracking of the Human Body. In Proceedings of IEEE Transactions 6 Frame 177 Frame 213 Frame 229 Frame 242.

[74] Xie, X and Mirmehdi, M. (2003). Geodesic Colour Active Contour Resistent to Weak Edges and Noise. In Proceedings of the 14th British Machine Vision Conference, pp. 399-408.

[75] Yu, X., Xu, C., Leong, H.W., Tian, Q., Tang, Q., and Wan, K.W. (2003). Trajectory-based Ball Detection and Tracking with Applications to Semantic Analysis of Broadcast Soccer Video. In Proceedings of the Eleventh ACM international Conference on Multimedia. MULTIMEDIA '03. ACM Press, New York, NY, pp. 11-20. Available from http://doi.acm.org/10.1145/957013.957018.

APPENDIX A

The feature selection algorithm introduced by Viola et al. [68]

- Given example images $(x_1,y_1),\ldots(x_n,y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,l} = 1/2m, 1/2l$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positive respectively.
- For t = 1,…T:
    - Normalize the weights,

      $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$

      so that w_t is a probability distribution.
    - For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to:
      $w_t, e_j = \sum_i w_i \mid h_j(x_i) - y_i$.
    - Choose the classifier, h_t, with the lowest error e_t.
    - Update the weights:
      $w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$,
      where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and
      $\beta_t = \frac{e_t}{1-e_t}$.
    - The final strong classifier is:

      $$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} a_t h_t(x) \geq 1/2 \sum_{t=1}^{T} a_t \\ 0 & otherwise \end{cases},$$

      where $a_t = \log\frac{1}{\beta_t}$.

APPENDIX B

Intel CS430 features

Image Sensor Type	CCD
Interface Type	USB
Still Image Capture Resolution	640 x 480
Video Capture Resolution	640 x 480 • 352 x 288 • 320 x 240 • 176 x 144 • 160 x 120
Digital Video Capture Speed	30 frames per second
Special Features	Snap-shot Button • Auto Exposure • Automatic White Balance
Color Depth	24 Bit

Sony XC-HR58 industrial camera

Image Device	1/2 type Progressive Scan CCD
Effective Picture Elements	782 (H) x 582 (V)
Image Size	SVGA 767 (H) x 580 (V)
Lens Mount	C Mount
Scanning System	50 Hz Non-Interlaced
External Sync Signal	HD/VD (2 to 5 Vp-p)
Video Output	1 Vp-p, Sync negative, 75 ohms unbalanced
Horizontal Resolution	600 TV lines
Minimum Illumination	1 lux (F1.4, Manual Gain MAX +18dB GAIN)
S/N Ratio	58 dB
Gain	Manual (0 to 18 dB)/ Fixed (0 dB) selectable

Shutter	Normal Shutter: OFF to 1/20,000 sec with switches Restart/Reset Async External Trigger Shutter (Non-Reset, Reset) Switchers or with Trigger Pulse Width
Gamma	1 (Fixed)
Normal Shutter Speed	1/100 to 1/20,000 sec
External Trigger Shutter Speed	1/4 to 1/100,000 sec
High Rate Scanning - R/R Mode	Binning Off: Max. 200 frames/sec (111 lines) Binning On: 300 frames/sec (57 lines)
Vibration Resistance	10 G (20Hz to 200 Hz)
Shock Resistance	70 G
Operating Temperature	-5° C to 45° C (23° F to 113 ° F)
Storage Temperature	-30° C to 60° C (-22° F to 140° F)
Operating Humidity	20 to 80% (no. condensation)
Storage Humidity	20 to 95% (no. condensation)
Power Requirements	DC 12V (+10.5V to 15V)
Power Consumption	2.0 W
Weight	50 g (1.8 oz)
Dimensions (W x H x D)	29 x 29 x 30 mm (1-3/16 x 1-3/16 x 1-3/16 inches)
Regulations	UL 6500 listed, FCC Class A Digital Device, CE (EN61326/97+A1/98), AS4251.1+A4252.1

APPENDIX C

Average test results of the indoors case

Techniques \ Test Results		Average True Positive Rate	Average False Alarm Rate	Average Processing Speed (Frames/sec)
Simple Background subtraction with verification	Optimization No	87.4%	1.35%	21.45
	Optimization YES	86.1%	1.21%	22.45
Image differencing between the current & previous frames	Optimization No	89.64%	1.07%	19.05
	Optimization YES	87.39%	1.22%	21.47
Background modeling using mixture of Gaussians	Optimization No	90.73%	1.02%	14.71
	Optimization YES	87.46%	1.97%	15.67
The cascade of boosted classifiers trained with the Haar-like features	Optimization No	30.12%	62.16%	6.16
	Optimization YES	31.38%	57.54%	8.12
Hough Transform for circle detection	Optimization No	11.8%	73.35%	8.25
	Optimization YES	11.3%	75.79%	10.10

Average test results of the outdoors case

Techniques \ Test Results		Average True Positive Rate	Average False Alarm Rate	Average Processing Speed (Frames/sec)
Simple Background subtraction with verification of various criteria	Optimization No.	23.46%	56.42%	20.05
	Optimization YES	22.89%	61.65%	22.48
Image difference between current & previous frames	Optimization No.	35.62%	49.82%	19.53
	Optimization YES	33.94%	51.73%	19.05
Adaptive background modeling using mixture of Gaussian distributions	Optimization No.	39.46%	26.46%	13.77
	Optimization YES	38.13%	25.38%	14.07
The boosted classifiers trained with the Haar-like features	Optimization No.	27.23%	52.63%	6.26
	Optimization YES	30.87%	54.49%	8.25
Hough Transform for circle detection	Optimization No.	3.66%	77.54%	8.12
	Optimization YES	3.43%	76.42%	10.91

APPENDIX D

The impact the tennis ball's image size has on the correct detection rate

Background subtraction with verification

Size of the ball (pixels)	Correct detection rate	Size of the ball (pixels)	Correct detection rate
1	1.20%	26	98.83%
2	1.31%	27	98.06%
3	1.34%	28	95.17%
4	1.35%	29	95.72%
5	1.41%	30	94.48%
6	10.51%	31	87.18%
7	13.17%	32	84.73%
8	14.65%	33	82.85%
9	17.76%	34	80.09%
10	24.87%	35	75.66%
11	36.73%	36	70.00%
12	40.98%	37	68.01%
13	50.75%	38	48.75%
14	67.91%	39	40.67%
15	72.36%	40	32.46%
16	75.88%	41	28.10%
17	81.62%	42	27.75%
18	85.39%	43	26.38%
19	88.68%	44	20.73%
20	90.37%	45	18.62%
21	93.68%	46	16.39%
22	94.83%	47	15.44%
23	95.02%	48	13.49%
24	95.11%	49	13.02%
25	97.46%	50	12.32%

Image differencing between the current and previous frames

Size of the ball (pixels)	Correct detection rate	Size of the ball (pixels)	Correct detection rate
1	1.22%	26	98.33%
2	1.38%	27	98.76%
3	1.44%	28	96.81%
4	1.55%	29	95.33%
5	1.76%	30	95.63%
6	12.11%	31	88.42%
7	13.76%	32	85.51%
8	15.08%	33	83.67%
9	18.16%	34	80.79%
10	25.76%	35	76.36%
11	37.63%	36	71.83%
12	41.48%	37	68.16%
13	51.52%	38	49.47%
14	69.16%	39	43.69%
15	73.41%	40	33.09%
16	78.74%	41	29.61%
17	82.47%	42	28.09%
18	87.82%	43	26.78%
19	89.07%	44	20.23%
20	90.76%	45	19.78%
21	94.18%	46	17.21%
22	95.03%	47	16.04%
23	95.82%	48	13.09%
24	95.21%	49	13.35%
25	96.36%	50	12.66%

Background modeling using a mixture of Gaussians

Size of the ball (pixels)	Correct detection rate	Size of the ball (pixels)	Correct detection rate
1	1.32%	26	98.09%
2	1.22%	27	98.76%
3	1.45%	28	96.07%
4	1.47%	29	94.38%
5	3.53%	30	94.49%
6	12.11%	31	88.03%
7	13.75%	32	85.65%
8	14.25%	33	81.75%
9	17.06%	34	80.28%
10	25.32%	35	75.98%
11	38.64%	36	68.45%
12	44.03%	37	67.61%
13	53.5%	38	50.31%
14	70.31%	39	43.45%
15	74.06%	40	34.76%
16	76.18%	41	30.46%
17	82.87%	42	26.44%
18	85.09%	43	24.36%
19	87.43%	44	23.65%
20	92.18%	45	20.32%
21	94.75%	46	17.97%
22	96.12%	47	16.96%
23	96.92%	48	12.69%
24	96.01%	49	12.13%
25	97.87%	50	11.66%

APPENDIX E

Average number of noise blobs per frame generated from different videos
Note: the least number of the noise are generated from video 5, 6, 13, and video 20. The most number of the noise are generated from video 7, 23, and 25.

The Technique		Average Number of Noise Blobs Per Frame
Simple Background subtraction with verification	Video 5	7.62
	Video 6	7.34
	Video 13	8.32
	Video 20	7.95
	Video 7	20.24
	Video 23	20.08
	Video 25	21.79
Image difference between current & previous frames	Video 5	5.39
	Video 6	6.19
	Video 13	5.37
	Video 20	5.47
	Video 7	18.43
	Video 23	19.13
	Video 25	18.06
Adaptive background modeling using mixture of Gaussian distributions	Video 5	4.66
	Video 6	5.07
	Video 13	5.06
	Video 20	5.22
	Video 7	16.85
	Video 23	16.25
	Video 25	16.72