Master's Thesis

Comparison between gaze and moving objects in videos for smooth pursuit eye movement evaluation

Andrea Åkerström

Department of Electrical and Information Technology,
Faculty of Engineering, LTH, Lund University, June 2015.

Comparison between gaze and moving objects in videos for smooth pursuit eye movement evaluation

Andrea Åkerström

Department of Electrical and Information Technology
Lund University

Advisor:
Martin Stridh, Department of Biomedical Engineering

June 15, 2015

Abstract

When viewing moving objects in videos the movement of the eyes is called smooth pursuit. For evaluating the relationship of eye tracking data to the moving objects, the objects in the videos need to be detected and tracked. In the first part of this thesis, a method for detecting and tracking of moving objects in videos is developed. The method mainly consists of a modified version of the Gaussian mixture model, The Tracking feature point method, a modified version of the Mean shift algorithm, Matlabs function bwlabel and a set of new developed methods. The performance of the method is highest when the background is static and the objects differ in colour from the background. The false detection rate increases, when the video environment becomes more dynamic and complex. In the second part of this thesis the distance between the point of gaze and the moving objects centre point is calculated. The eyes may not always follow the centre position of an object, but rather some other part of the object. Therefore, the method gives more satisfactory result when the objects are small.

Contents

Chapter 1

Introduction

Eye tracking is today an active area of research. The application has become widely used in studies related to the visual system, in psychology and in the interactions between computer and humans.

An eye tracker is a system that tracks the movements of the eye so that an estimation of the gaze can be performed. Event detection has become an important area when trying to identify the different eye movements. An event detection algorithm is therefore needed to evaluate the eye tracking data. The main movements of the eye are fixations, saccades and smooth pursuit movements.

Until now, most eye tracking studies have been carried out using still images. Lately, however, motion pictures have become more interesting to study and the interest for detection and analysis of smooth pursuit movements has subsequently increased. When the relationship, of the eye tracking data and the motion pictures, is to be evaluated, one of the difficulties is that either the video data has to be manually measured, or an intelligent system has to be created for an automatic evaluation.

The analysis of videos is a widespread field of research and a large portion of the research is related to detection and tracking of moving objects. It can be applied in many areas like surveillance cameras, vehicle navigation, human computer interfaces and medical image analysis.

Moving object detection and tracking means segmentation and extraction of the moving objects and then tracking throughout the video frames. It is a complex problem due to all possible scenarios such as dynamic backgrounds, camera noise, camera movements, illuminations, occlusion, objects changing orientation and size.

The aim of this thesis is to attempt to develop an automatic evaluation of smooth pursuit detections, when viewing moving objects in videos. The thesis consists of the following two parts:

- Based on a investigation of different methods for detection and tracking of moving objects in video sequences, develop a method that gives the most adequate result for the present problem.
- Develop an automatic evaluation strategy for smooth pursuit detection methods, based on the detected and tracked moving objects in videos and eye tracking data that has already been analysed using an existing algorithm.

The thesis is outlined as follows:

- First a background to the problem is given in Chapter 2.
- The proposed method is described in Chapter 3.
- Chapter 4, the results are given.
- And finally, a discussion and a conclusion of the work are given in Chapters 5 and 6, respectively.

Chapter 2

Background

Detecting and tracking moving objects in a video sequence is a process that first identifies and extracts the objects from a scene and then the objects are tracked by finding the same objects in the subsequent frames and, in that way, be able to calculates the trajectories of the objects [27]. This is done based on the information in the video frames, such as visual features and motion information [1].

Detecting and tracking moving objects is a complex problem for several reasons such as dynamic backgrounds, camera noise, camera movements, illuminations, occlusion, objects changing orientation and size, etc. It should also be avoided to track non-stationary background objects such as swinging leaves, rain, snow and shadows cast by moving objects. Furthermore, in order to handle objects that stop or begin to move and objects that merge into the background and then become foreground constitute additional difficulties.

A common approach to this problem which consists of the two parts, detecting and tracking of moving objects, is to have different methods for the two parts. It is also common to use more than one method for each part.

The first part that detects the objects in a scene is typically a method that segments and extracts the moving objects from the video frames [14], [6], [4], [3], [22]. The main categories of methods for this part are optical flow, temporal differencing and background subtraction [20] also called background modeling [6].

The second part consists of the localization and tracking of the moving objects [14], [6], [4], [3]. There are several different methods for object tracking. The most common categories are region-based tracking, feature-based tracking, model-based tracking and active contour-based tracking [1].

Background modeling includes methods that use some kind of model of the background and then compares each frame in the video to this reference or background model. The regions that deviate from the background are identified as foreground and should consist of moving objects [6]. For dynamic backgrounds in videos the background reference or model must be continuously updated which makes the method computationally expensive [3]. Optical flow based methods calculates the flow vectors of moving objects between the frames in order to find the moving regions in a video. Temporal differing is detecting moving objects by using threshold values to find where pixels are differing from the pixels in the consecutive frames in a video [20].

The main idea of region-based tracking is to find the region of a detected object in the adjacent frame, using a similarity measure. A threshold value is used to decide if the region of the object has moved. In feature-based tracking the moving object is located with help from features that have been detected in one frame. The features, corresponding to the different moving objects, are then tracked in the following frames, so that the new location of the moving object can be found in every new frame. Model-based tracking algorithms define a model of the problem, usually based on knowledge of different possible scenarios in a video. It then detects the moving objects by matching the different parts of the frame in to the model and then sees if the parts belong to it [7]. Active contour-based tracking tries to track the contours of an object. Usually this is achieved by defining an energy function that is minimized [4].

2.1 Methods for objects detection

2.1.1 Frame differing

Frame differing is a method that uses the previous frame as its background model and differs this model to the current frame in order to find the moving objects. Frame differing is the easiest method of background modeling [6].

2.1.2 Median filter

Median filtering is commonly used and estimates the background from a number of previous frames. It calculates the median of the pixel values in the previous frames and compares it to the pixel values in the current frame. There is also a recursive version of the median filter. This method decreases or increases the median value for a pixel depending on if the value of the pixel from a new frame is larger or smaller [6].

2.1.3 Threshold segmentation method

The threshold segmentation method uses direct subtraction of the background after getting the background image. A threshold value is set and the value decides whether a pixel from an image is belonging to the background or not, based on how much it deviates from the background image [22].

2.1.4 Gaussian mixture model (GMM)

Gaussian mixture modeling (GMM) is a background modeling method that is used to extract the moving objects from dynamic background in a video. This is done by making a background model consist of a weighted sum of several Gaussian distributions. Every pixel then has a probability density function which consists of a number of Gaussian distributions that is decided from the history of the pixel.

The model can be adaptively updated by comparing each new pixel to a number of Gaussian distributions. If a pixel does not match any of the Gaussian distributions, the distribution with the lowest weight is replaced with a new distribution with an initially high variance, low weight and the new pixel value as its mean. The background model is given by the distributions that have weights larger than a chosen threshold.

A pixel belongs to the background if it matches any one of the distributions otherwise it belongs to a moving object. For a higher threshold the background model will consist of more Gaussian distributions and may therefore be able to model more dynamic backgrounds. If the background is static, the threshold may therefore be lower in order to avoid false motion detection. The GMM precision to extract moving object depends strongly on the threshold parameters [5].

The GMM is one of the most common methods for detecting moving objects in dynamic backgrounds [19]. The method may give false motion detection in complex scenes and it may have a slow learning of the background [5].

2.1.5 Optical flow

Optical flow uses the motion information in videos to detect moving objects in videos sequence. The method calculates characteristic flow vectors of moving objects over time [20]. This is done by, for each pixel in the current frame, trying to determine a two-dimensional velocity vector, which should indicate the position of the same pixel in the previous frame.

There are several algorithms for computing the optical flow that can be divided into the categories; differential, region-based matching and energy-based algorithms [16].

The Optical flow method is complex and sensitive to noise, [20], but shows a good performance and the computationally cost is low [23].

2.1.6 Salient Points

The salient points can be seen as landmarks in an image. They can for example be the corners of a building or the edges of an object. The methods based on salient points extract these points from every frame and then a post processing stage is needed to eliminate the salient points that do not belong to moving objects.

Two different algorithms for detecting salient points are the Scale-Saliency and Scale Invariant Feature Transform - SIFT. Both methods use a scale-space approach and estimates key point and salient regions.

The Scale-Saliency method chooses the salient regions by calculating the entropy for a area around a pixel at each scale and find where it peaks. The salient region is a circular area and it is invariant to rotations.

SIFT is a popular method that finds key points or regions and extracts features for them, which are invariant to scale, rotation and translation. In order to find the key points or region, a scale-space function is used that consist of a pyramid of differences of Gaussian frames constructed from the first frame. Key points are chosen by examining the maxima and minima of the function.

Compared to background subtraction, methods based on salient points represent a simple way to identify moving objects in video. Studies indicate that the Scale-Saliency algorithm is slightly superior to the STIF algorithm but computationally a little more expensive [3].

2.2 Methods for object tracking

2.2.1 Mean Shift Algorithm (MSA)

The Mean shift algorithm (MSA) is a method that constructs an object model and then use it to search for the object location in another frame.

The method needs a selected window that contains the object in the first frame. For the object model, a histogram distribution of the color is estimated with a nonparametric symmetric kernel. Then, the same kind of histogram is estimated for different candidate regions in the next frame.

The histogram distributions of the object and the different candidate regions are then compared and the best match gives the new location for the object. In order to measure the similarity between the two distributions the Bhattacharyya coefficient is usually used.

The MSA performs well when tracking moving objects in a simple environment. The method is sensitive to the influence of noise from the background [8] and its performance is low when the illumination condition changes fast, the object is being shadowed [13], or if the object changes direction suddenly [12]. It has become a successful method for searching of similarity regions, for its robustness [4], for highly tracking accuracy, and high algorithm speed [13].

2.2.2 CAMshift

Continuously adaptive mean shift: CAMshift is a method that is based on the MSA. The method was developed to overcome problems with the MSA, like that it is being unable to change the track window scale when an object changes size [18].

CAMshift has extended the MSA into a continuous frame sequence that uses the MSA in every frame. The initial search window in a frame is given from the result from the previous frame. [17] The method is adaptively updating the size of the search window and the color distribution of the object model.

CAMshift can track objects in a robust and efficient way. The method has good performance in simple situations, but in more complex situations it fails [18].

2.2.3 Particle filters

The Particle filter is a Monte Carlo integrated method that can track objects in videos with non-Gaussian or non-linear conditions.

The method randomly chooses a number of particles, in a region where an object is likely to be located and then weights are calculated for the particles. Based on the weighted particles a new region of interest can be chosen and the weighted particles can be updated. This procedure is repeated until the object is tracked.

Particle filter gives good tracking performance, but the method is also slow because of the very high number of calculations that has to be made [13].

2.2.4 Kalman Filters

The Kalman filter recursively predicts and corrects the states of a linear process that is disturbed with Gaussian process noise. This is done by minimizing the variance error that can be calculated with help from measurement of the process.

For tracking of moving objects, the Kalman filter includes a motion model and measurement equations. The states of the process are the different dynamical behaviors of the objects. The tracking of the objects is done by recursively estimating the states of the object.

Tracking objects using a Kalman filter is a robust method and it has given good results in many situations. However, in more complex situation such as inter-object occlusion, separation [9] and if the object changes direction suddenly [12] the Kalman filter may fail and gives uncertain result [9].

2.2.5 Active contour-based tracking

The active contour-based tracking method is used for tracking of the contours of an object. This is mainly achieved by defining an energy function that is minimized to segment an image [4]. The method needs an initial guess for the contour, which is then moved with help from the energy function to the boundaries of the object [10].

The methods snake and level set are the main methods for contour-based tracking [4]. The snake method has a parametric representation of the contour and the energy function will try to shorten the contour and make it smooth. This will make the snakes compact and robust to image noise and boundary gaps. The snake method is not good at handling objects that changes its shape.

The level set method has an implicit representation of the contour and is good at handling objects that change their shape. The method has some drawbacks including that it is not robust to boundary gaps [10].

Contour-based tracking is used to achieve high precision of the tracking, but the computational cost is often high [4].

2.2.6 Block Matching Techniques

In methods using block matching techniques, the video frames are divided into blocks. The blocks in a current frame are then compared to the blocks in an earlier reference frame in order to evaluate if the blocks are located in the same place in the frames or if they have moved and in that case to determine there corresponding motion vectors. Moving objects can be identified by combining the blocks with the same motion.

Block matching techniques can be divided into three parts: block determination, search method and matching criteria. The goal is to find a method that with as few searches as possible, in order to reduce the processing time, gives an adequate result. The first part determines properties of the blocks like their position, size, scale and the start location of the search for matching blocks. A simple block determination could be made, that divides the frames in disjoint blocks of fixed sizes and scales, or a hierarchical block determination can be made which creates different levels of the frames with different resolution. Each level has a simple block determination and the search starts at level with the lowest resolution and continues at each subsequent level where the best match is found from the previous level. Studies have shown that hierarchical block determination has a higher performance of finding the object motion vectors but a longer processing time, than the simple block determination.

The second part is to use a search method that decides where to search for the candidate blocks in the reference frame that would give the best match for the blocks in the current frame. The methods gives step sizes on how far away the next search for a candidate block would be.

The third part in the process is to decide the matching criteria so that it measures the dissimilarity of the blocks. The best match of all the candidate blocks in each search can then be determined. The method checks before every search if the blocks in the current and the reference frame are in the same place and in that case the search is terminated [2].

2.3 Other methods

2.3.1 Detecting objects

The method in [5] is an improved version of the GMM that has been used together with a support vector machine classifier. The GMM first determines which of the pixels that belong to the background or the foreground. After that, the support vector machine classifier determines if the foreground pixels are motion or non-motion pixels. The method was shown to give reliable and robust results in dynamic backgrounds. It converges faster at the initial stage than the traditionally GMM. Furthermore, this method showed a great improvement of decreasing the false motion detection and of the quality of the image segmentation [5].

In [22], moving objects are detected in static scenes. This is accomplished with an improved background extraction algorithm based on common region, which divide an image in to blocks and calculates the expectation and variance value. A modified threshold segmentation method is used to extract the moving objects. These improvements gave a greater robustness and higher quality of the methods [22].

A method for modeling of the motion of the background for detection of moving objects when the camera is moving is proposed in [23]. The optical flow between the frames is calculated and based on this, a pixel motion process is made. This is used togheter with the GMM for making a background motion model for a moving scene. The method was an effective approach and gave an accurate, adaptive and general background motion model [23].

In [24] a method for detecting moving objects with moving camera is proposed. The method is using flow vector distance classification and multiview geometric constraints. This method gave results that was robust and accurate. It could be used in unknown environment and the adaptability to light is good [24].

2.3.2 Tracking objects

[8] has proposed an improved MSA where the representation of the object is better. This is achieved by an improvement of the color histogram and using an exponent distribution that could be updated over time, instead of the kernel function. A further improvement for handling objects under occlusion was made by adding a sub-block occlusion detecting algorithm. The result showed that the improved MSA preformed better when tracking moving objects and handling occlusion than the traditional MSA [8].

Another improvement of the MSA is made by [11] where they have done an adaptive dynamic updating of the object model. The method showed improved performance when dealing with rigid and non-rigid objects, camera motion, background clutter and target partial occlusions, rotation, scale variations [11].

In [15], an improvement of the MSA has also been performed by making the bandwidth and kernel weights adaptive. The Kalman filter was then used for further improvement of the tracking accuracy. The method showed good result tracking objects and it had accelerated in convergence speed [15].

A fail-safe way for tracking of a general purpose object was proposed in [12]. They use a template-matching technique called normalized 2D cross correlation, to track objects in videos and then to update the template for the region of interest. If the template-matching does not find an object, both

Kalman-filter and fast MSA is used to compensate for this in an adequate way [12].

A Kernel-based method for finding an object region is used in [4], the MSA and the Kalman filter will then locate the initial tracking position. Then, a diffusion snake, based on a feature image that is generated from the objects color information, is used to improve the tracking precision. The first part of the method was shown to be able to locate the object effectively in complex condition such as camera motion, partial occlusion and clutter. By adding the second part of the method it gave high tracking precision and it was also able to handle severe deformation of object contour. But it had problems to track the object when the objects color was similar to the background, and the computational cost is high [4].

The MSA is an effective object tracking method, but it fails in more complex environments. Because of this, [13] has combined the MSA with the particle filter that has a higher tracking performance, but also has a higher computational cost. This is done using a feedback system based on the MSA and when it gives bad tracking results, it uses the particle filter to improve it. The method gave good result in situations such as for objects having shift, rotation and scaling and the tracking speed was adequate [13].

A method for multiple object tracking in complex situations such as inter-object, occlusion and separations was developed in [9]. In this method, correlation weights were applied to Histogram Intersection-HI, named Correlation-Weighted Histogram Intersection–CWHI. Then, Kalman filters was used to track each of the objects. The method preformed well under complex situations [9].

In [17] the CAMshift method and the Kalman filter were combined for tracking moving objects. Linear prediction was also used with the Kalman filter when objects were under occlusion to improve the robustness of the system. The tracking method was shown to be fast and robust. It was able to handle occlusion and camera movements, but because of that the method was using color histogram the object needed to differ in color from the background [17].

Since the CAMshift method has a low performance in more complex situations, [18] has made an improvement of the CAMshift method by combining it with a low-cost motion segmentation phase. The motion segmentation reduces the difficulties related to the background and object color interference. The method showed to be able to track moving objects in a better way than the traditionally CAMshift [18].

2.3.3 Detecting and tracking objects

A method for detecting and tracking multiple moving objects using a static camera was proposed in [21]. For extracting moving objects, a color background modeling was used together with a sensitivity parameter. Morphology was used to reduce noise and blob labeling was used to assemble moving objects. The tracking method consisted of prediction of the velocity and direction of the moving objects. The method showed to be computationally fast and it showed robustness against environmental nosie [21].

In [25], a codebook algorithm was constructed with a quantization/clustering technique to be used as a background model. Then, the foreground was segmented by a subtraction of the background model from the current frame. The particle filter was then combined with the MSA for tracking of the moving objects. The method could handle moving backgrounds, illumination and was robust to occlusion [25].

Chapter 3

Methods

The aim of this present thesis is to attempt to develop an automatic evaluation of smooth pursuit detections, when viewing moving objects in videos. The thesis was divided into two parts. Firstly, in section 3.1, based on a investigation of different methods for detection and tracking of moving objects in video sequences, develop a method that gives the most adequate result for the present problem. Secondly, in section 3.2, develop an automatic evaluation for smooth pursuit detection methods, based on the detected and tracked moving objects in videos and eye tracking data.

3.1 A method for detection and tracking of moving objects in videos

The main purpose of the proposed method is that it should detect and track multiple moving objects and compare the trajectories with detected smooth pursuit movements in the eye tracking data. The method should be able to handle as many complex situations in video sequences as possible.

In the development of the method, most decisions that have been made have been based on previous research and on the Matlab Computer Vision System toolbox. During the process, the results of the different parts of the method have been continuously evaluated and improvements that were shown to be necessary have been made.

The Matlab Computer Vision System toolbox has been useful in the way that it has been possible to test different methods and compare their performance without having to implement all methods my self. Based on their performanse, I have chosen to implement those that would be of use in my method. Furthermore, Matlab toolbox has worked as an easy way of checking if the developed method behaves in a similar way and by that verifying that they have been implemented correctly.

An important design aspect of the developed method is that it does not need to be able to work in real-time. This means that it is not constrained by the requirement that the processing has to be computationally fast.

In order for the method to serve its purpose, it has to include a method for detecting moving objects. This method has to be repeated for every video frame in case new moving objects appear. When a detection method is used repeatedly in this way and is combined with an additional method that determines which detections that belong to the same object, it can also work as an object tracking method. This approch has been chosen for the problem because it is easier and it saves time to use the same method for both the detection and tracking part. The detecting method can function as a key building block for the entire method. Other methods can then be added to improve the methods preformance and compensate for situations where the detection method fails.

As a detection method the Gaussian mixture model (GMM) was chosen, which is a background modelling method. This is a widely used method for modelling dynamic backgrounds. It can represent complex distribution of each pixel and is able to handle periodic noise, such as waving leaves on trees or moving sea waves [27]. Since the method suffers from false motion detection and can have a slow learning rate of the background [5], some modification and additional methods is needed to improve its performance.

Other background modelling methods has not been taken into consideration because they are more suited for videos with relatively static backgrounds [27]. Temporal differencing methods generally have a bad performance when extracting the exact shapes of some moving objects. Optical flow methods were under consideration in the process of choosing detection method, primarily because it can handle camera movements. On the other hand, the method are complex and sensitive to noise and cannot detect the exact contour of the moving objects [28]. It is preferable if the method is able to detect the contour of the objects. If optical flow had been used, an additional method for finding the exact contour had been needed. The snake GVF method, which is an Active contour-based tracking method, by Chenyang Xu and Jerry L. Prince was tested out, it did not give adequate results. The reason for this could be that the objects did not have strong enough edges. An additional reason for choosing the GMM over Optical flow methods is that there were more possibilities for the GMM when it comes of making different modifications and add other methods, which should improve its performance.

The GMM cannot handle camera shakes and movements. Therefore, the method needed some complementation for this purpose. From the Matlab Computer vision toolbox, Tracking feature point method (TFPM) which can estimate camera movements was chosen. A Block matching method for estimation of camera shakes and movements was also implemented and tested but it did not give any adequate results.

As an additional step in the method the Mean shift algorithm (MSA) was chosen. This tracking method is not supposed to work as a tracking method by itself, but can be used to support the GMM. It is preferably to have a method that is simple and easy to understand and by that also easy to work with. I chose this method over other tracking methods, to be able to easy make, for my purpose, relevant modifications. Furthermore, in previous studies it is well used together with other methods and it is the combination with other methods that is necessary to achieve my purpose. The MSA did not give a more accurate result of the exact position of an object in a frame than the GMM gave. Therefore, the MSA was not used as a traditionally tracking method that would help in the process of calculating the new position of an object in the next frame. Instead, a modified version to decide which detections in two subsequent frames made by the GMM that belongs to the same objects was used.

There were some other additional method that were needed during the course of the development. A necessary method was one that segmented all the pixels, which the GMM had detected, into different objects. Here, the Matlabs function bwlabel was used. Another method, that improved the end result significantly, was one that decreased false detections. For that I wrote an algorithm that removed object that were to small so they were considered to be noise and newly detected objects that seemed to be static.

The ultimate aim of this thesis is to use the detected and tracked moving objects for evaluation of smooth pursuit detection in corresponding eye-tracking data. It is then important that the method detects all the moving objects. Unfortunately, it was difficult to make this method good enough to give accurate result in all different kinds of videos. To still be able to use the method together with the eye tracking data, a program that could be used after the main method was made. This program is helping the user to manually modify the results by changing the settings. Modifications that can be done are to remove unwanted detections, merge a number of objects that belongs to the same object, and estimate the objects size and trajectory if the method is detecting the object wrongly in a number of frames. Because of this program, the aim when detecting and tracking moving objects is rather to detect to many object that could later be removed, than that some of the moving objects are missed.

3.1.1 Method overview

The block diagram in figure 3.1 shows the main steps in the proposed method for detection and tracking of moving objects in videos. The method starts by initializing the GMM in the step 'Init GMM'.

In the step 'Estimate camera movements', the camera movements are calculated using the TFPM. This is done based on the previous and current frames. Depending on the outcome from this estimation, there are two different paths for the algorithm to follow: The algorithm takes the first path if the movement is estimated to be zero and the camera has not moved. Then, it goes tdirectly into 'Update GMM'. The second path is taken if the amount of movements is not zero and the camera has moved. Then the algorithm goes into 'Adjust GMM', where the background model is moved to its new position. A initialization of the background model is also done on the new areas of the video scene, that appeared when the camera moved. The algorithm then continues in to 'Update GMM'.

In the step 'Update GMM', the GMM is updated and the pixels in the frame is divided into background and detected foreground pixels. Inputs to this box are the GMM that is going to be updated and the current frame that is used for the updating process. Another input is the previous frame, that is used to initialise a new Gaussian distribution in areas that have been falsely detected and static objects. These static objects and false detections are also an inputs. If a major part of the pixels, has been detected as forground pixels the algorithm goes back to the step 'Init GMM'. There, an initialization of the entire background model is done based on the current frame and all the moving objects disappears. This should happen in case of a scene change.

The detected foreground pixels are grouped together into different objects in the box 'Object segmentation'. Here, the centre points and size measurements are calculated for all the different objects that are detected by the GMM.

In step 'Mean Shift Algorithm', it is decided which objects in the previous frame that belongs to same objects in the current frame. The objects are here classified as moving object, which are the ones that find a matching object, or new objects, which are the ones that do not find a matching object. The moving objects in the previous frame, that are not detected in the current frame is marked as finished and are removed.

Both the new and the moving objects are passing the step 'Removal of

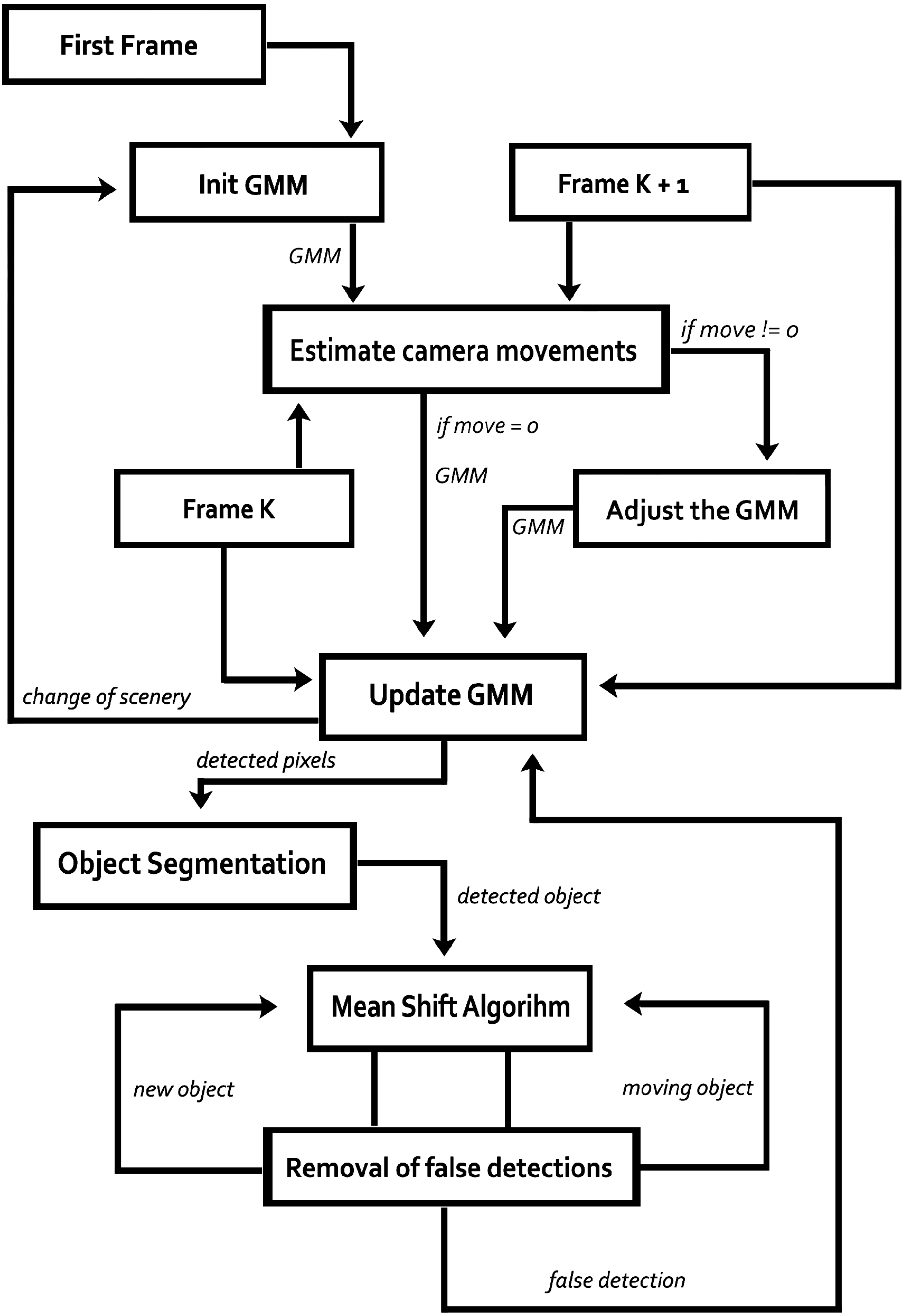


Figure 3.1: Block diagram over the main steps in the method of detecting and tracking moving objects in videos.

false detections'. Here, all the objects that are small enough to be considered as noise are classified as false detections. Moving objects that have only been occurring in a few frames and have not moved far enough are considered to be static objects and are also classified as false detections. All false detections goes back into 'Update GMM'.

Based on the new detections of the moving objects, the size, position and the amount of movement is calculated. The detected objects that have been classified as new objects or moving objects, takes the path back to the same box 'Mean Shift Algorithm' and is used in the next iteration.

3.1.2 Gaussian mixture model (GMM)

Traditional GMM

The Gaussian mixture model (GMM) is used for detecting pixels, in a video frame, that are part of moving objects. In this method, a mixture of Gaussian distributions is used to model the probability that a pixel I_t has the colour RGB vector values X_t at time t and this distribution is defined as

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * f(X_t|\mu_{i,t}, \Sigma_{i,t}) \tag{3.1}$$

For the i-th Gaussian distribution at time t the $\omega_{i,t}$ is the estimated weight, which represents the time proportions that those pixel values has been in a video scene. The parameter $\mu_{i,t}$ is the mean value, $\Sigma_{i,t}$ is the covariance matrix and K is the number of distributions which usually varies between 3 and 5. The function f is a Gaussian probability density function defined as

$$f(X_t|\mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{n/2}|\Sigma_{i,t}|^{1/2}} e^{-\frac{1}{2}(X_t-\mu_{i,t})^T \Sigma_{i,t}^{-1}(X_t-\mu_{i,t})} \tag{3.2}$$

where n is the length of the random vector X_t. Assuming the parts of X_t are independent and with the same variance $\sigma_{i,t}^2$, the covariance matrix can be computed as

$$\Sigma_{i,t} = \sigma_{i,t}^2 \mathbf{I} \tag{3.3}$$

The K Gaussian distributions are sorted after their values of $\omega_{i,t}/\sigma_{i,t}$, where the the first distribution has the largest value and the last one has the smallest value.

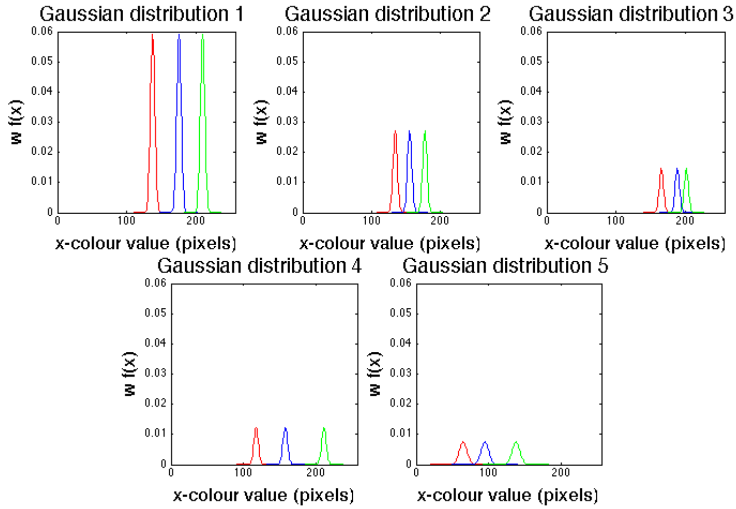


Figure 3.2: These figures show an example of a mixture of five Gaussian distributions that is used to model the probability that a pixel I_t has a certain colour value.

Figure 3.2 shows an example of a mixture of five Gaussian distributions that is used to model the probability that a pixel I_t has a certain colour value.

A new pixel I_t belongs to the i-th Gaussian distribution if the following condition is fulfilled.

$$|X_t - \mu_{i,t-1}| \leq D \cdot \sigma_{i,t-1} \tag{3.4}$$

where D is a threshold value. For the first match that is found, for a new pixel I_t, the parameters of the matching distribution are updated as follows:

$$\begin{aligned} \omega_{i,t} &= (1-\alpha)\omega_{i,t-1} + \alpha \\ \mu_{i,t} &= (1-\rho)\mu_{i,t-1} + \rho X_t \\ \sigma^2_{i,t} &= (1-\rho)\sigma^2_{i,t-1} + \rho(X_t - \mu_{i,t})^T(X_t - \mu_{i,t}) \end{aligned} \tag{3.5}$$

Here, α is the learning rate and $1/\alpha$ represents a time constant which determines the speed at which the parameters change. And, ρ is the second learning rate and is computed as

$$\rho = \alpha f(X_t|\mu_{i,t-1}, \Sigma_{i,t-1}) \tag{3.7}$$

For the unmatched distributions in the GMM, the parameters $\mu_{i,t}$ and $\sigma_{i,t}$ remain the same. The weight parameter is updated as follows:

$$\omega_{i,t} = (1-\alpha)\omega_{i,t-1} \tag{3.8}$$

All the updated weights are then normalized.

If a new pixel I_t does not belong to any one of the K distributions, the last distrubution is replaced by a new Gaussian distribution with the new pixel value as its mean value, a high variance value, and a low weight value.

The background model is chosen to consist of the first B Gaussian distributions that fulfills the following condition:

$$B = armin_b(\sum_{i=1}^{b} \omega_{i,t} > T) \tag{3.9}$$

where T is a threshold value for the minimum proportion of the data that should be included in the background model.

If a new pixel I_t matches anyone of the background model distributions, the pixel is classified as a background pixel, otherwise it is classified as a foreground pixel [29].

Modified GMM

In the method, five different Gaussian distributions have been used, because the GMM would then be able to describe a more diverse model of the background.

When working with the GMM a lower resolution in the video frames have been chosen. Regions in the frames of size $[5, 5]$ are then used instead of single pixels. The main reason for this is that even though the method does not have to work in real time and be computationally fast, the time for computing the GMM was so tedious that it was almost impossible to test and work with the program. To lower the resolution also lowered the computation time rapidly. Another advantage is that the number of false detections that the caused by noise are decreased when working with regions.

	Sign	*Value*
Number of Guassian distributions	K	5
Initial weight	$\omega_{i,0}$	0.2
Weight for a new Guassian distribution	$\omega_{5,t}$	0.1
Initial standard deviation	$\sigma^2_{i,0}$	10
Standard deviation for a new Guassian distribution	$\sigma^2_{5,t}$	30
Learning rate	α	0.005
Threshold for a pixel that belongs to the distribution	D	2.5
Threshold for determine the background model	T	0.15

Table 3.1: The initial parameter value and the value of the constants that is used in the method.

There are three different scenarios in which there is a need to make an initialization of the GMM. The first case is in the beginning of the process when the entire first frame needs initialization values. The second case is when there is a change of scenery in a video, so that the first frame that is representing a completely new scene needs initialization values. Finally, the third case is when the camera moves and the new regions of a scene appear and these new regions need initialization values.

Usually a number of training frames are used for the estimations of the initial values of the GMM. However, it is preferable if my method could start to give adequate result immediately after an initialization of the model and not need a number of frames for training of the background model. Due to that, I have chosen the initial mean values based on the current frame so $\mu_{1..5,t}(x,y) = [X_{1,t}(x,y), X_{2,t}(x-1,y-1), X_{3,t}(x+1,y-1), X_{4,t}(x-1,y+1), X_{5,t}(x+1,y+1)]$ where x and y are the position in the image matrix.

Table 3.1 shows the initial parameter values and the values of all constants used in the method. The selected values are based on which values that has been commonly used in other studies and by testing the method to see which values that give the best result for different video scenarios.

In this modified GMM, it is decided in the same way as in the traditional GMM if a new pixel I_t belongs to one of the Gaussian distributions. Likewise, if a match is found or not found, the values of the parameters are also updated in the same way as in the traditional method.

When the GMM is used together with the TFPM, which estimates the camera movements, an additional method is used for adjusting the models pixel positions in the video frame. This is needed because the models still need to represent the same pixels in the background after the camera has moved. First, the TFPM is calculating the pixels translational displacement and after that, all the new pixel positions, that the diffrent GMM now are representing, are determined.

When the Removal of false detections (RFD) is used, all the detected pixels that have been determined to be part of a non-moving object are called false motion pixels. A false motion pixel is then known to be part of the background scene. I have chosen to replace its last Gaussian distribution with a new distribution with initial values $\omega_{5,t} = 0.2$ and $\sigma^2_{5,t} = 10$, since it is then immediately included in the background model.

3.1.3 Tracking feature point method (TFPM)

The Tracking feature point method (TFPM) is used for estimate camera movements. In the TFPM, feature points are extracted from a frame and then a feature-tracking algorithm is estimating the point's translational displacement in the next frame.

In the TFPM, a block with the size $[31, 31]$, is chosen so that it covers a feature point in a frame. Each pixel in the block has the grey colour value $G(x, y)$, where x and y are the position in the image matrix. A candidate block, that covers a candidate feature point in the next frame, has the grey pixel value $F(x+h_x, y+h_y)$, with the displacement h. The aim is to find the displacement h where $G(x, y) = F(x + h_x, y + h_y)$ and this can be achieved by the following iteration.

$$h_{k+1} = h_k + H^{-1} \sum_{x,y} [\nabla F]^T [G(x, y) - F(x + h_{x,k}, y + h_{y,k})] \tag{3.10}$$

The initial value is set to $h_0 = 0$. The gradient $\nabla F = [\frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}]$ is calculated with the displacement h_k and $H = \sum_{x,y} [\nabla F]^T [\nabla F]$ [32].

The feature points are extracted from a frame by first dividing the entire frame into blocks, each block contains one potentially feature point. For finding blocks that contains strong feature the H is determined for every block and then the two eigenvalues λ_1 and λ_2 are calculated for each H. A block is decided to be a feature point block if

$$min(\lambda_1, \lambda_2) > \lambda \tag{3.11}$$

for a threshold value λ [33]. In this method, I have chosen to use the value

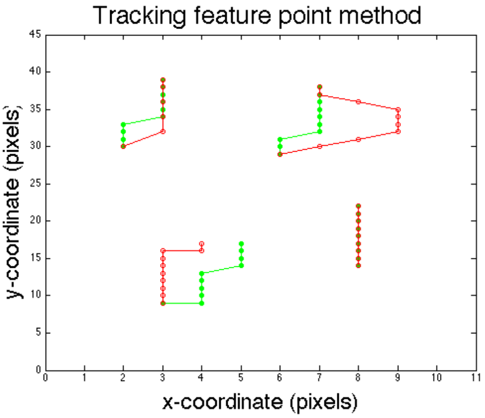


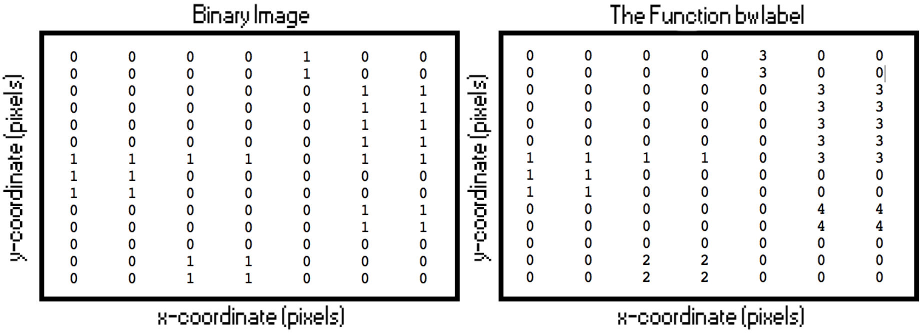
Figure 3.3: An example of four feature points that with the TFPM has estimated the points' translational displacement in the next frame step by step. The path that has the colour green shows the forward tracking and the colour red shows the backward tracking.

$\lambda = 50$ based on what gave the best result when testing the method. Furthermore, feature points that are positioned on moving objects are removed. This is because, feature points in the background are only wanted when its movements are consistent with the camera.

In addition to the first tracking point procedure, the feature-tracking algorithm is repeated from the tracked feature points' position in the next frame backwards to the previous frame. The distance error between the original feature point position and the resulting feature point position after tracking both forward and backward is calculated with the L_2 norm [31]. If the distance error exceeds a threshold of 1.4143, the feature point is considered to be unreliable and is eliminated.

An example of the TFPM can be seen in figure 3.3. The figure shows four feature points where the translational displacement has been estimated in the next frame step by step. The path that has the colour green shows the forward tracking and the colour red shows the backward tracking.

The iteration for calculating the displacement h is ended when the block error in one iteration is larger then the block error from the previous iteration. The number of iterations has a limit of 10 iterations. The camera movements are then calculated by taken the mean of the displacement values from all the remaining feature points.


Binary Image
The Function bwlabel
y-coordinate (pixels)
x-coordinate (pixels)

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

0	0	0	0	3	0	0
0	0	0	0	3	0	0
0	0	0	0	0	3	3
0	0	0	0	0	3	3
0	0	0	0	0	3	3
0	0	0	0	0	3	3
1	1	1	1	0	3	3
1	1	0	0	0	0	0
1	1	0	0	0	0	0
0	0	0	0	0	4	4
0	0	0	0	0	4	4
0	0	0	0	0	0	0
0	0	2	2	0	0	0
0	0	2	2	0	0	0

Figure 3.4: In the left panel an example of a binary image can be seen. Here, pixels that have been detected as part of a moving object in a frame, by the GMM, is marked with the ones and the zeros represent the background pixels. The panel to the right shows when the Matlabs function bwlabel has divided the binary image into different objects with diffrent object labels.

3.1.4 Object segmentation method (OSM)

This method is used for segemtation of all the detected pixels into objects. The detected pixels are the pixels that have been determined as being part of a moving object by the GMM.

In the Object segmentation method (OSM), one of Matlabs function bwlabel was used. The function uses a binary image, where the ones are representing detected pixels and zeros are representing background pixels. The function labels and divides a binary image into different components. This is done by calculate the detected pixels connected compnenets in the binary image. In this function, either 4-connected objects or 8-connected objects can be used, where the last-mentioned divides the image into more objects than the first-mentioned. Then, all the pixels that belong to the same objects are labelled with the same number. The background pixels are labelled with the number zero.

In this work the 8-connecting object method has been chosen, because it is preferable to have too many objects and then, if needed, merge some of them together, instead of having too few objects and then try to find a way to split them. The modified MSA is used to decide if different objects are actually part of the same object and therefore should be merged together. When merging objects together their different labels are just changed to the same number.

In panel 3.4 an example of the Matlab function bwlabel. In the panel to the left an example of a binary image can be seen. The panel to the right shows when the function bwlabel has been used.

3.1.5 Mean shift algorithm (MSA)

Traditional MSA

The Mean shift algorithm (MSA) is using object modelling to decide which object detections in two subsequent frames that belongs to the same objects. For the object model a feature histogram distribution based on hue, saturation and value, HSV, colour map is calculated and the u-th feature characteristic probability is defined as

$$\hat{q}_u = C \sum_{i=1}^{n} k(\| \frac{x_0 - x_i}{h} \|^2) \delta[b(x_i) - u] \tag{3.12}$$

where $\{x_i\}_{i=1...n}$ is the pixel locations in the object region, x_0 is the central pixel in the object region. The parameter h is the bandwidth and δ is the Kronecker delta function. The function $b(x)$ calculates which bin in a quantized feature space a pixel belongs to. A common number of bins to use are 16 x 16 x 16. The function $k(\|x\|^2)$ is a kernel function and a commonly used kernel is the Epanechnikov kernel

$$k(\|x\|^2) = \begin{cases} (1 - \|x\|^2) & if \|x\| \leq 1 \\ 0 & if \|x\| > 1 \end{cases}$$

The normalized factor C is defined as

$$C = \frac{1}{\sum_{i=1}^{n} k(\| \frac{x_0 - x_i}{h} \|^2)} \tag{3.13}$$

which assures that the condition $\sum_{u=1}^{m} \hat{q}_u = 1$ is fulfiled.

Figure 3.5 shows examples of hue, saturation and value histogram distribution that are used to model an object.

In the next frame, the object candidate region u-th feature characteristic probabilty is defined as

$$\hat{p}_u(y_i) = C \sum_{i=1}^{n} k(\| \frac{y_i - x_i}{h} \|^2) \delta[b(x_i) - u] \tag{3.14}$$

where y_i is the center of the region and C is a normalized factor. The

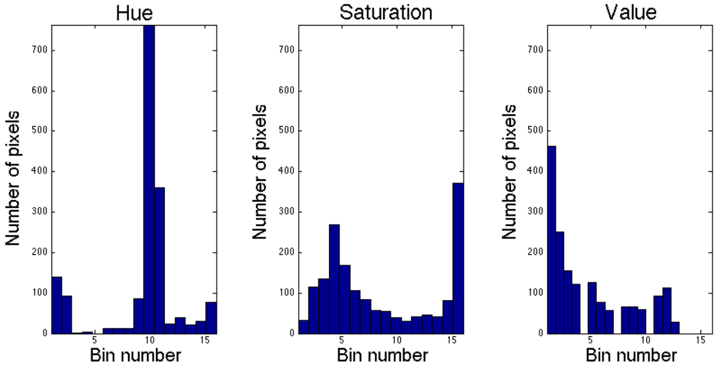


Figure 3.5: Three examples of hue, saturation and value histogram distributions that are used to model an object.

similarity between the initial object model and the candidate object model is measurered with the Bhattacharyya coefficient that is calculated as

$$\hat{\rho}(y) = \rho(\hat{p}(y), \hat{q}_u) = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y)\hat{q}_u} \tag{3.15}$$

The distance between the two distributions is calculated as

$$d(y) = \sqrt{1 - \hat{\rho}(y)} \tag{3.16}$$

For finding the best location for the candidate region the $\hat{\rho}(y)$ should be maximized. This is done by estimating the weights

$$w_i = \sum_{u=1}^{m} \sqrt{\frac{q_u}{p_u(y_i)}} \delta[b(x_i) - u] \tag{3.17}$$

Then the next location of the candidate region is derived according to

$$y_{i+1} = \frac{\sum_{i=1}^{n} x_i w_i g(\|\frac{y_i - x_i}{h}\|^2)}{\sum_{i=1}^{n} w_i g(\|\frac{y_i - x_i}{h}\|^2)} \tag{3.18}$$

where $g(\|x\|^2) = -k(\|x\|^2)$. This process is repeated until $\|y_{i+1} - y_i\| < \varepsilon$ or the number of iteration reaches a maximum value, [4] [17] [30].

Modified MSA

In this work a modified version of the MSA was used. The idea is to compare a detected objects' histogram and size in one frame with all possible candidate object histograms and sizes in the next frame around the same area and find the best match.

The u-th feature characteristic probability is changed to

$$\hat{q}_u = C \sum_{i=1}^{n} \delta[b(x_i) - u] \tag{3.19}$$

where only the pixels that have been detected by the GMM are included in the calculation. A kernel function is not used and consequently all pixels contribute with the same amount to the feature histogram. The normalized factor C becomes

$$C = \frac{1}{n} \tag{3.20}$$

The u-th feature characteristic probabilty of the candidate object is calculated in an analogue way

$$\hat{p}_u(y) = C \sum_{i=1}^{n} \delta[b(x_i) - u] \tag{3.21}$$

Here, $\hat{p}_u(y)$ is referring to the candidate object which has its centre in y.

The feature histogram is calculated for the current object in a frame and for all the candidate objects that has its centre point in a search area in the next frame. The search area is a circle centred in the same place as the current object had its centre. The circle radius R is calculated as followed

$$R = \begin{cases} max(w_0, h_0) + 10 & if \quad max(w_0, h_0) < 90 \\ 100 & if \quad max(w_0, h_0) \geq 90 \end{cases}$$

where the parameter w_0 is the width and h_0 is the height of the current object. The value 100 is decided to be the maximal movement an object can take between two frames. The decisions for calculating the radius are based on the movements the moving object had in the different videos that have been availble for testing.

The Bhattacharyya coefficient $\hat{\rho}(y)$ is used to measure the similarity between the two distributions. The similarity in size $A(y)$ of two objects are calculated as followed

$$A(y) = \frac{min(w_y/w_0, w_0/w_y) + min(h_y/h_0, h_0/h_y)}{2} \tag{3.22}$$

where the parameter w_0 is the width and h_0 is the height of the current object. The parameter w_y is width and h_y is the height of the candidate object.

A detected object in the next frame is only considered as a candidate object if $\hat{\rho}(y) > 0.5$, $min(w_y/w_0, w_0/w_y) > 0.3$ and $min(h_y/h_0, h_0/h_y) > 0.3$. The values of these thresholds are chosen so that they helped to decrease the false detection rate, without risking to remove detections of the moving objects. If none of the objects in the next frame fulfils these constraints, the current object is considered to have disappeared.

The similarity measurement $s(y)$ is determined as

$$s(y) = \frac{\hat{\rho}(y) + A(y)}{2} \tag{3.23}$$

The candidate object that gives the largest value of $s(y)$ is determined to represent the current object in the next frame.

In some cases, several smaller detected objects belong to the same larger object and they need to be merged together. Because of this, when the current object has found its best match among the candidate objects, a new calculation of the similarity measurement is done. The measurement is now calculated for the candidate object, which had the best match, merged together with each and everyone of all the other candidate objects in the search area. If adding some of the other candidate objects increases the similarity measurement, they are merged together with the candidate object that had the best match. When merging objects together, all the detected pixels for the different objects are the referring to the same object and a new centre point and size is calculated.

3.1.6 Removal of false detections (RFD)

In order to improve the method further, two different kinds of filter that should remove false detections of moving objects have been implemented. One filter that checks the sizes of the objects and one filter checks the movements of the objects.

If a detected objects' width or height exceeds a threshold of 5 and it is not in contact with another moving object it is considered to be noise and it is then classified as false detection.

A detected object has to move a certain length in the first frames to be classified as a moving object, otherwise it is considered to be static and is therefore classified as a false detection. I have chosen to look at the first six frames and the detected object has to have moved at least a length of ten pixels, calculated with the L_2 norm.

All false detections are removed by including them in the GMM that models the background.

3.1.7 Manual modification of the result (MMR)

The purpose of the implemented software is to facilitate the user to manually improve the resulting moving objects data, which is produced from the method for detect and tracking of moving objects in videos.

In the software, videos can be played with the data of the detected moving objects. All moving objects and frames are numbered in the videos. The user can then choose which detections of the moving objects that is correct and should be retained. In the software, the number of these objects can be filled in and all the other objects are then removed.

The user can remove inaccurate detections from a chosen number of frames, where a moving object occurs. This can be done both in the beginning and in the end of the moving objects trajectory.

The user can also specify if several objects are part of the same object and should be merged into one object. The new merged objects height, width and trajectory are estimated. This is done by first including the different objects areas so they belong to one object and then by calculating the new height, width and center position which is a point on the trajectory. Secondly, if there are frames in between the objects where the detection of the moving object is missing, the object information is estimated with linear interpolation using the objects' data from the frame it last occurred in and from the first frame it reappeared in.

3.1.8 Evaluation of the methods

The evaluation of the performance of the method is focused on the methods ability to detect and track moving objects.

In the evaluation process, there is a problem of not having real measurements of the contour and trajectories of the different moving objects in a video. Thus, the result from the method cannot be compared to the real measurements to validate their correctness. In this evaluation, the measurement for the moving objects are manually determined in each frame and used as validation data. Hence, the trajectory of the centre point of each moving object is manually calculated. Since this process is very time-consuming only the width and height of the objects are measured instead of the entire contour. Manually determined validation data contains a lot of errors, like inaccurate measurement when for example a moving object is blurry and do not have sharp edges.

As the video material, the proposed method was tested on four different videos. These videos contain different complex situations that the method is supposed to be able to handle.

In the first movie the background is static and contains one scene change. In both scens there is one object whih is moving fast in one main direction. The object changes its shape and sometimes some parts of the object has a similar colour as the background surrounding it.

In the second video, the background is mostly static but in some parts it is dynamic. The camera shakes and is sometimes making small and slow movements. There are several objects in the video and it also occurs size changes, occlusions and objects that have similar colour as the surrounding background.

The third video has a static background with camera shakes and several scene changes. There are small fast moving objects and one larger moving object that can be seen as noise instead of smooth pursuit movements. Objects are sometimes under occlusions and it happens that objects divide in to several smaller objects.

The fourth video has a static background with camera shakes. There is one small moving object that is under occlusion in parts of the video.

3.2 Comparison between the eye movements smooth pursuit and moving objects

The aim of the second part of this thesis is to develop an automatic evaluation strategy of an detection algorithm for eye tracking data. Specifically, the trajectories of moving objects are compared to the trajectories of smooth pursuit eye movements. The proposed evaluation method measures how well the eyes are following the moving objects.

3.2.1 The method

In the evaluation method, distance measure are defined between the gaze trajectory and each of the moving object trajectories. For gaze trajectories, preanalysed data has been used which were generaded as follows: The eye movements have been estimatet using an eye tracker. Then, an event detection algorithm has been used to evaluate the eye tracking data and determine if the eye movements are of the type of fixation, saccade or smooth pursuit [24].

In the evaluation method, the measurement on how well the eyes are following a moving object is done by calculate the distance between the point of gaze and the objects centre point. The distance is calculated with the L_2 norm and this is done in every frame.

3.2.2 Evaluation of the method

In order to investigate if the method gives a reasonable result, the automatic evaluation method has been tesded both on the estimated moving object data and on manually annotated object data.

The video material used for evaluation of the performance of the method are the first and the fourth videos that are described in the section 3.1.8. The data that contains the manually measured centre point of the moving objects for those two videos are also the same that are described in that section. The reason for only using the first and the fourth video was that they had already completed eye tracking data available.

Chapter 4

Results

4.1 Detection and tracking of moving objects in video sequences

In this section, the main results of the development of the detection and tracking method is presented. The results of the proposed method show its ability to detect and track objects in different video scenarios. Figures 4.1-4.12 show examples from the development process of the method. While figures 4.13-4.17 show the resulting performance of the finaliced method different video scenarios.

4.1.1 Test setup 1

The result of using only the GMM is presented in this section.

The GMM gives adequate results when the background is static and the object differ in colour from the background. An example of this scenario is shown in figure 4.1 a), which shows the resulting detections for one frame in the first video. Here, the background is static and the parts of the object that differs in colour from the background are detected correctly. However, the method fails to detect the object in the parts where the colour is almost the same as the surrounding background.

When the background is dynamic and when the camera is moving and shaking the method suffer from false detections, but the moving objects are still correctly detected as long as they differ in colour from the background. This can be seen in figure 4.2 a), which shows detections for a frame from the second video, where the two moving objects are detected but there are also a lot of false detections. The moving object in the right side in the frame is hard to distinguish from the false detections. The majority of the false detections are caused by small movements of the camera and some of it are caused by things like moving leafs.

A result for a frame in the third video is shown in figure 4.3 a). Here, the background is static and there are little camera shakes. The video contains a small fast moving object, which is detected correctly. There are also false detection caused by camera shakes. There are also detections of a person to the right in the frame, that only moves a little and therefore should be considered as a false detection.

An example frome the fourth video is shown in figure 4.4 a), where there is a static background that suffers a lot from camera shakes. The resulting video is covered with false detection from all the camera shakes. There is one small moving object that is detected, but it is difficult to distinguish from all the false detections.

4.1.2 Test setup 2

In this section the result of using both the GMM and TFPM is presented.

After adding the TFPM to the GMM the result for the videos with moving or shaking camera are in most cases improved. The best result is given when the camera makes small translational movements.

A result from the first video, which has a static environment, can be seen in figure 4.1 c). Here, when adding the TFPM causes extra false detections. The reason for this may be that some feature points are placed on the moving object, which then contributes to the camera movement estimation.

In figure 4.2 c) the improvement can be seen in the second video where the camera makes small and slow movements. After adding the TFPM, the false detections have been reduced significantly and the two moving objects are clearly showing in the frame.

Figure 4.3 c) shows a reduction of the false detections that are made from camera shakes in the third video. The false detections of a person moving to the right are not improved, since those false detections are not a result of the camera movements.

Figure 4.4 c) shows that the false detections in the fourth video have been reduced only slightly. This is because the camera has a lot of camera shakes. Additionally, it seems to have both rotational and translational movements and the TFPM is only supposed to handle translational movements. Furthermore, the result shows that the moving object is more visible after adding the TFPM.

4.1.3 Test setup 3

The result, presented in this section, is of using GMM, OSM and MSA together.

Dividing the detected pixels, from the GMM, into different objects with the OSM method and using the MSA to merge together different detected parts that should belong to the same moving objects, gives poor result when there are a lot of false detections. Using the MSA for tracking of each moving object though the different frames works most of the time, but the result depends on the amount of false detections.

In the first video, there are very little false detections and based on the detected pixels the moving object is segmented accuratly, which can be seen in figure 4.1 b). The missing part of the moving object is the result of the GMM inadequate performance. Using the MSA to merge together different detected parts that should belong to the same moving objects works some time, but not when the different detected parts are too far away from each other.

Figure 4.2 b) shows the result for one frame of the second video. There are a lot of false detections in the frame and of the two moving objects in the frame are segmented correctly. Both sizes of the detected moving objects are too large because it includes falsely detected pixels.

In figure 4.3 b), the results of the third video can be seen. Since the small moving object in the frame is not surrounded by falsely detected pixels, it is segmented correctly. The person to the right in the frame is only moving a little and should therefore not be classified as a moving object and it is also not segmented correctly.

In the fourth video, the segmentation of objects fails completely, which can be seen in figure 4.4 b), due to that the frame is covered in false detections. The majority all of the detected pixels, that are included in the segmentation of the small moving object, are falsely detected pixels.

4.1.4 Test setup 4

In this section the result of using GMM, OSM, MSA and TFPM is presented.

When using the TFPM together with the GMM, OSM and MSA, the moving objects are detected more correctly, due to the reduction of the falsely detected pixels. Merging together different detected parts that should belong to the same moving objects, using the MSA, works some times. It is rarely that the MSA merge together detected parts to the moving objects that do not belong to it. The processes of using the MSA for tracking of each moving object though the different frames are also improved when the TFPM is added.

Figure 4.1 d) shows a frame for the first video. Here, the segmentation of the moving object is basically the same as before when the TFPM was not used.

When the TFPM has decreased the level of falsely detected pixels in the second video, one of the moving object segmentations is accurate and the other is almost correct but some false detected pixels are included in the detected object. This can be seen in figure 4.2 d).

Figure 4.3 d) shows the result for one frame in the third video. The small moving object is still segmented correctly.

The result for the fourth video is still containing a lot of falsely detected pixels. However, when adding the TFPM the small moving object seems to be segmented in a correct way. Figure 4.4 d) shows a frame for the fourth video.

4.1.5 Test setup 5

The result of using the GMM GMM, OSM, MSA, TFPM and RFD is presented in this section.

When the RFD is added to the GMM, OSM, MSA and TFPM, the number of false detections decreased significantly. The filter that removes too small objects, in the RFD, is the major cause of the reduction of the number of false detections. Adding the filter that removes nonmoving objects, in the RFD, adds further to this reduction.

Figure 4.5-4.8 a) and b) show the results when using the filter that removes the small objects only and c) and d) show the results when using both filters.

Figure 4.5 shows the results from the first video. The false detections are decreased considerbly when small objects are removed and a few more are removed when nonmoving object are removed.

The filter that removes small objects is heavily reducing the number of false detections in the second video, which can be seen in figure 4.6. Before using the filter, only one of the moving objects were segmented correctly but now both segmentations are correct. Adding the filter that removes nonmoving objects are reducing false detections a little bit more.

In the third video, the removal of small objects are reducing the false detections and adding the removal of non moving objects are reducing false detection some more. This can be seen in figure 4.7.

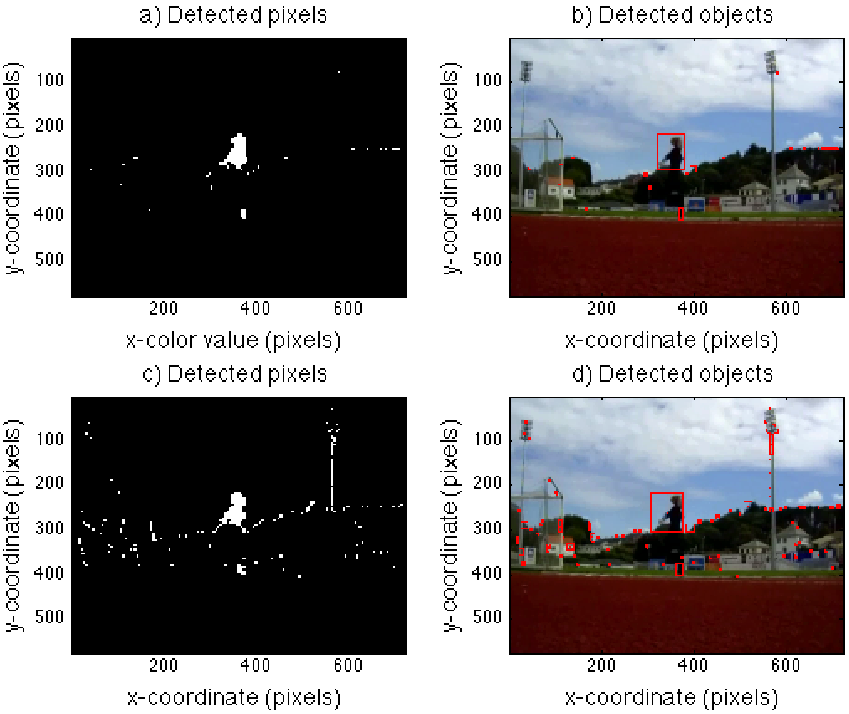


Figure 4.1: The result for frame 20 in video 1, where there is one moving object that has its centre point at [340, 313]. a) The detected pixels using GMM. b) The detected objects using GMM, OSM and MSA. c) The detected pixels using GMM and TFPM. d) The detected objects using GMM, TFPM, OSM and MSA.

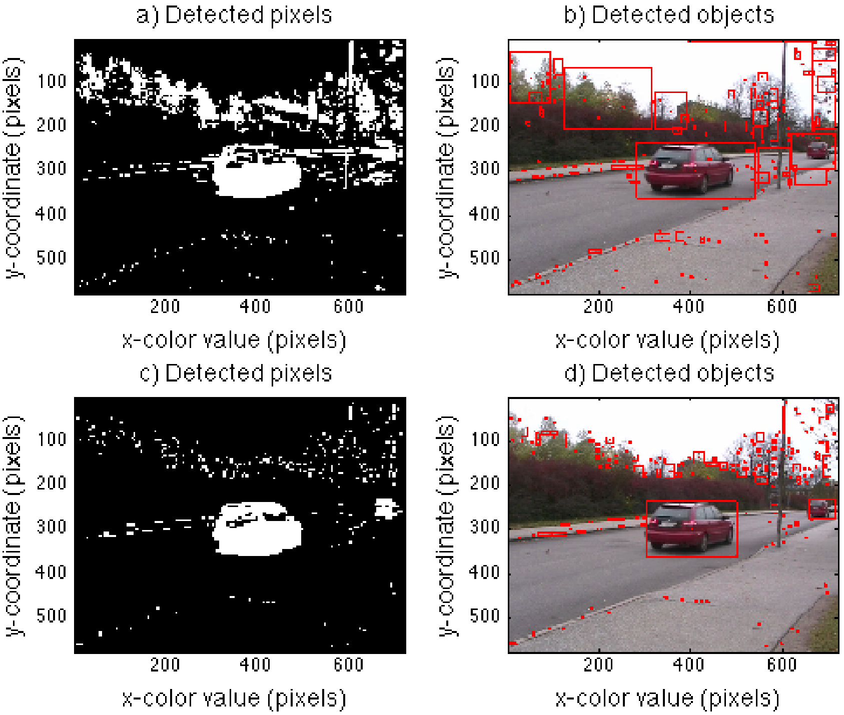


Figure 4.2: The result for frame 100 in video 2, where there are two moving objects that have their centre points at [401, 298] and [679, 252]. a) The detected pixels using GMM. b) The detected objects using GMM, OSM and MSA. c) The detected pixels using GMM and TFPM. d) The detected objects using GMM, TFPM, OSM and MSA.

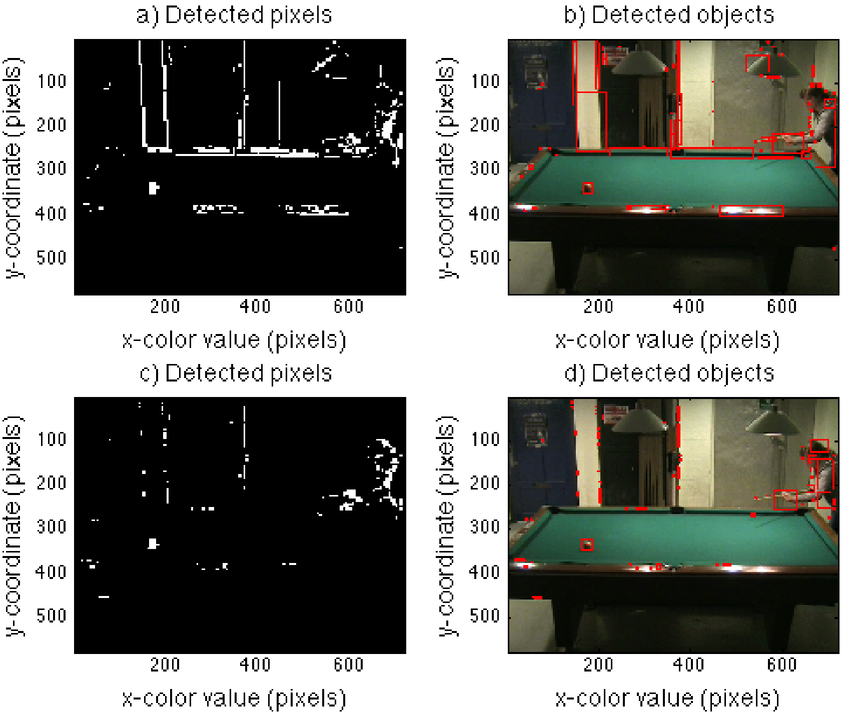


Figure 4.3: The result for frame 50 in video 3, where there are two moving objects that have their centre points at [175, 336]. a) The detected pixels using GMM. b) The detected objects using GMM, OSM and MSA. c) The detected pixels using GMM and TFPM. d) The detected objects using GMM, TFPM, OSM and MSA.

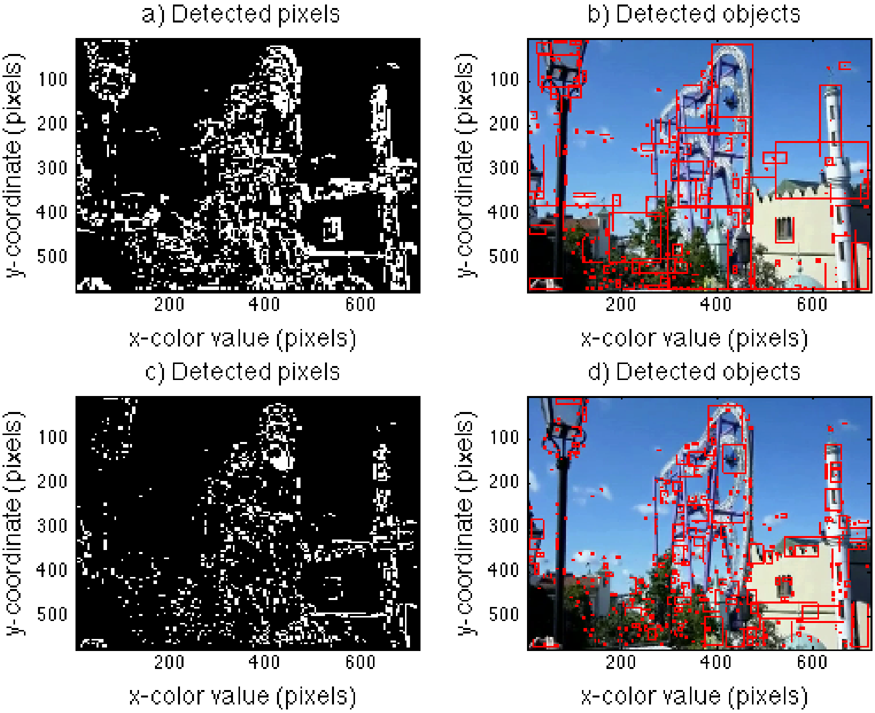


Figure 4.4: The result for frame 150 in video 4, where there are two moving objects that have their centre points at [424, 146]. a) The detected pixels using GMM. b) The detected objects using GMM, OSM and MSA. c) The detected pixels using GMM and TFPM. d) The detected objects using GMM, TFPM, OSM and MSA.

In figure 4.8, result for the fourth video is shown. Here, the filter that removes small objects decreases the false detections a lot and when additionally using the second filter, that removes nonmoving objects, the number of false detections decreases a lot as well.

4.1.6 Summary over video sequences of the proposed method

The proposed methods are in most cases able to detected and track the entire or parts of the moving objects in the video sequences. False detections appear in video sequences where the background is dynamic or the camera is shaking or moving.

In figure 4.9, all detected and tracked moving objects in the first video can be seen. The background is static and therefore there appear only a couple false detections. The two moving objects are partly or completely detected and tracked though the frames. The MSA has failed to merge parts of the same moving object together. This is because those parts are too far away from eachother when the parts in between are lost when the moving objects and the background have a similar colour. The video has one change of scenery between the two moving objects, which the methods are able to handle.

The moving objects in the second video is shown in figure 4.10. The two moving objects are almost correctly detected and tracked in the video sequence. One of the objects is divided into two different detections during a few frames, when the object is under occlusion. There are a lot of false detections mostly caused by the dynamic background occurring in a certain part of the frames. Often the false detections are small and disappear after a few frames.

In the sequence for the third video, the fast and small moving object are accuratly detected and tracked, which can be seen in figure 4.11. All of the false detections shown in the figure are from the person in the video that is moving a little bit and is therefore considered to be noise and not a moving object.

Figure 4.12 shows the detected and tracked moving objects in the third video. It takes some frames before the methods manage to detect the small moving object in the video and it is lost a couple of times. The cause of this is that the moving object is under occlusion at those times. There are plenty of false detections because of the camera shakes, which seems to be both translational and rotational. The TFPM method is only estimating the translational displacement of the camera.

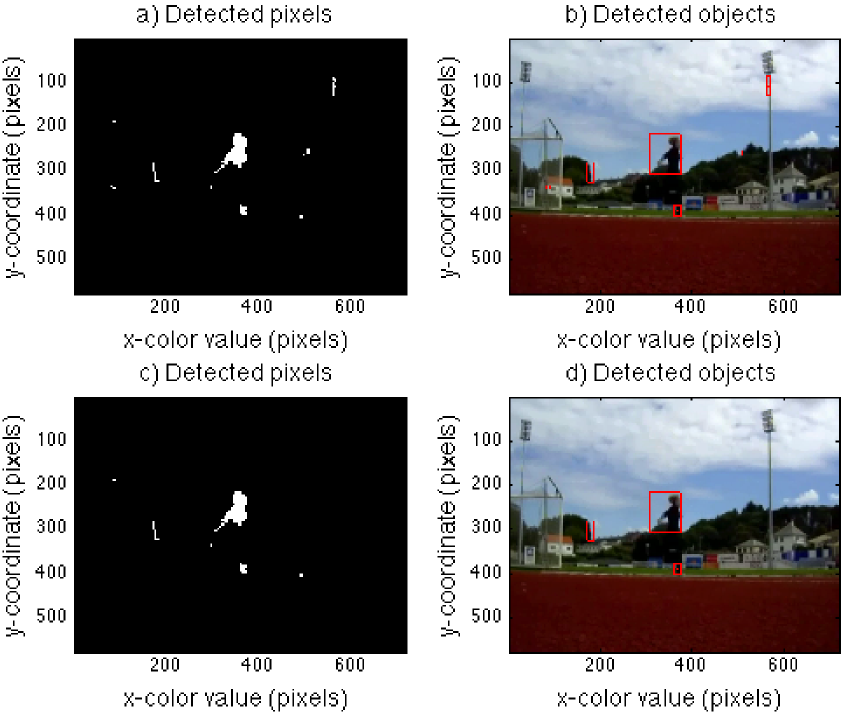


Figure 4.5: The result for frame 20 in video 1, where there is one moving object that has its centre point at [340, 313]. In all of these result, GMM, TFPM, OSM and MSA, are included in the method. a) The detected pixels and b) The detected objects, here the methods have been used together with the filter that removes too small objects in RFD. c) The detected pixels and d) The detected objects, here the methods have been used together with a filter that removes to small objects and a filter that removes non moving objects in RFD.

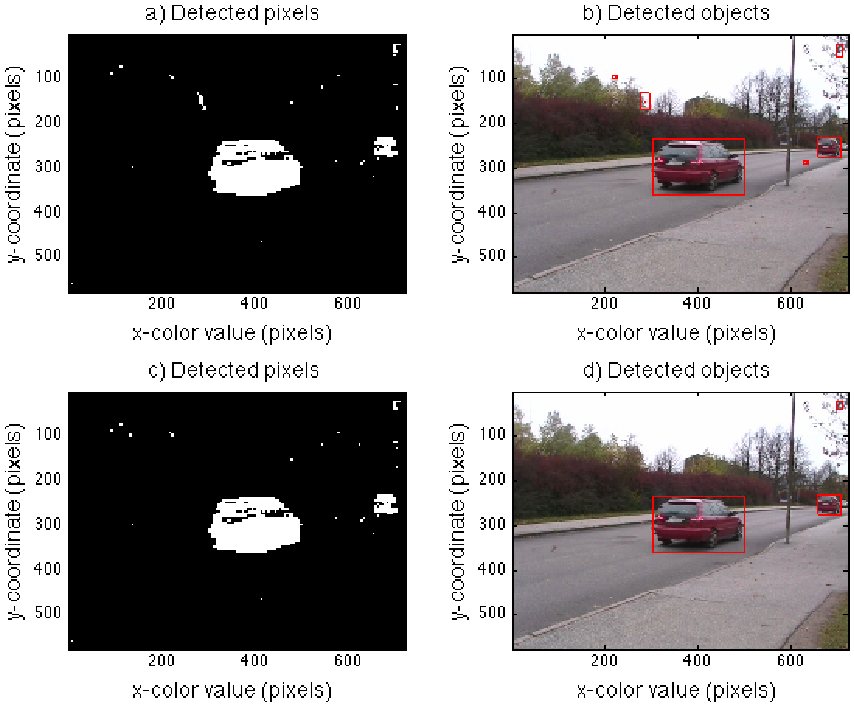


Figure 4.6: The result for frame 100 in video 2, where there are two moving objects that have their centre points at [401, 298] and [679, 252]. In all of these result, GMM, TFPM, OSM and MSA, are included in the method. a) The detected pixels and b) The detected objects, here the methods have been used together with the filter that removes too small objects in RFD. c) The detected pixels and d) The detected objects, here the methods have been used together with a filter that removes too small objects and a filter that removes non moving objects in RFD.

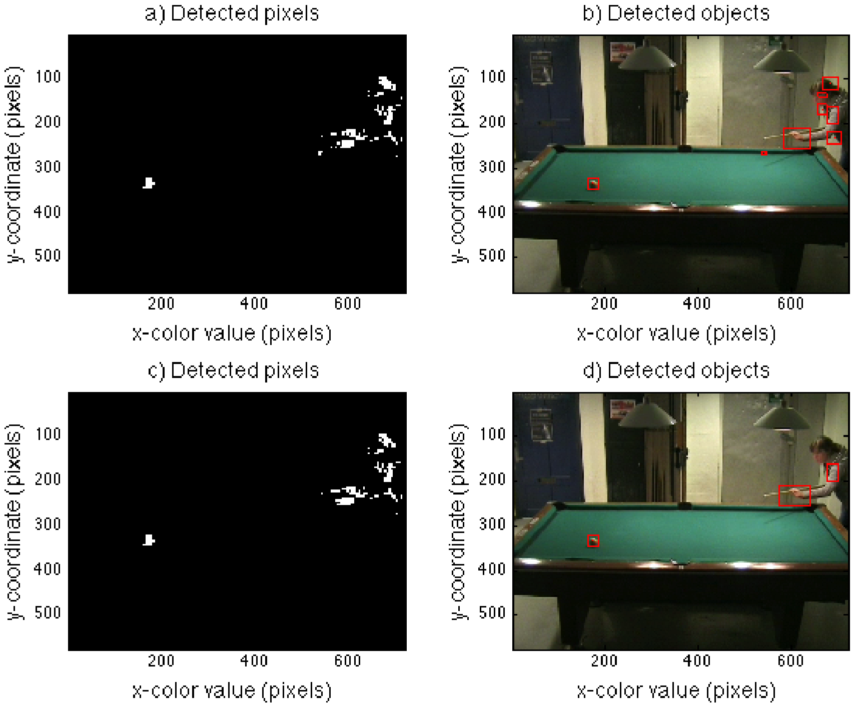


Figure 4.7: The result for frame 50 in video 3, where there is one moving objects that has its centre points at [175, 336]. In all of these result, GMM, TFPM, OSM and MSA, are included in the method. a) The detected pixels and b) The detected objects, here the methods have been used together with the filter that removes too small objects in RFD. c) The detected pixels and d) The detected objects, here the methods have been used together with a filter that removes to small objects and a filter that removes non moving objects in RFD.

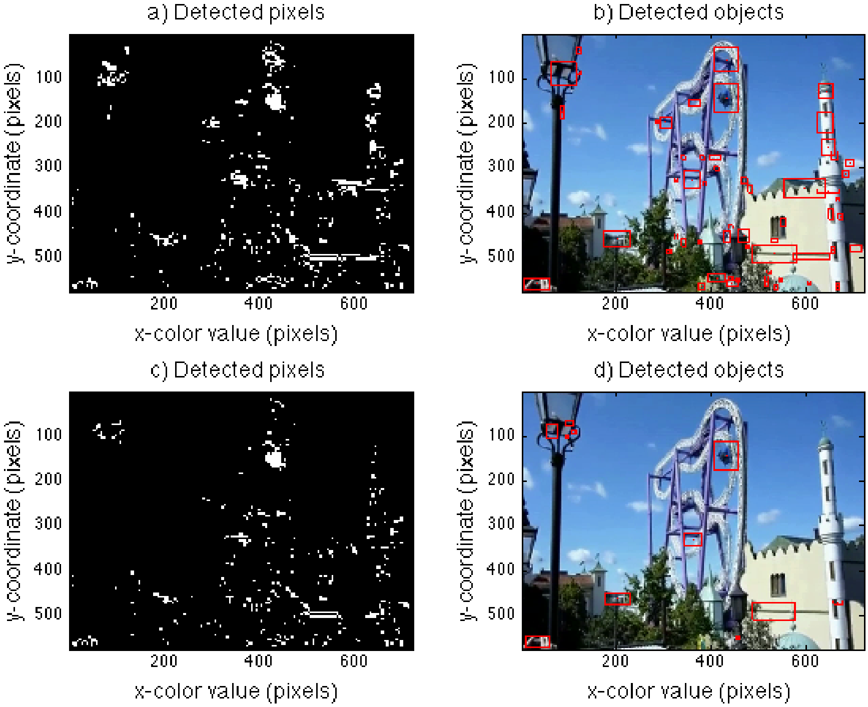


Figure 4.8: The result for frame 150 in video 4, where there is one moving objects that has their centre points at [424, 146]. In all of these result, GMM, TFPM, OSM and MSA, are included in the method. a) The detected pixels and b) The detected objects, here the methods have been used together with the filter that removes to small objects in RFDd. c) The detected pixels and d) The detected objects, here the methods have been used together with a filter that removes too small objects and a filter that removes non moving objects in RFD.

4.1.7 The proposed method with MMR

When the resulting proposed method is used togheter with MMR, the desired object detections are extracted and by that, all false detections are removed. In figures 4.13-4.17 the remaning moving objects are compared to the real measurements of the moving objects, which has manually been determined. In each figure, a) a frame with the detected object is shown. In b) trajectory showing the moving objects centre positions in each frame both from the real and estimated measurement. In c) shows the distance calculated as the L_2 norm, between the real and estimated measurement of the centre position in the different frames. In d) the ratio of the real and estimated measured area of the moving objects in the different frames is shown.

Figures 4.13 and 4.14 show the result for two different video clips in the first video. In the first video clip, using MMR, three different parts of the first moving objects have been merged together and in the second video clip four different parts of the second moving object has been merged together, where the end of one of the parts trajectory has been removed. In figure 4.9, the different parts that have been detected can be seen. Because of the similar colour that the moving object and the background have, parts of the moving object fail to be detected. This causes the centre point to deviate from its real trajectory in a lot of the frames, which can be seen in figure 4.13. The colour of the moving object and the background differs a little bit more in the second video clip and therefore gives a better result, which can be seen in 4.14. The ratio of the estimated and real error is closer to one and the estimated trajectory follows the real trajectory much better.

The result for the first moving object in the second video can be seen in figure 4.15. The detection of the moving object is only divided in two parts during a few frames. The two detected parts, which can be seen in figure 4.10, are merged together with the MMR. When the moving objects first appear in the frame, the method do not manage to detect the entire object area. Therefore, the estimated centre position and area is inaccurate in the beginning. After that, the performances of the methods are improved and their estimation of the trajectories follows the true trajectories better.

In the third, video the MMR has been used to extract one moving object from all the false detections. All the detections can be seen in figure 4.11. The result for the one extracted moving object is shown in figure 4.16. The moving object is small and fast and the background surrounding it is of one colour which is different from the object. Because of this, the method

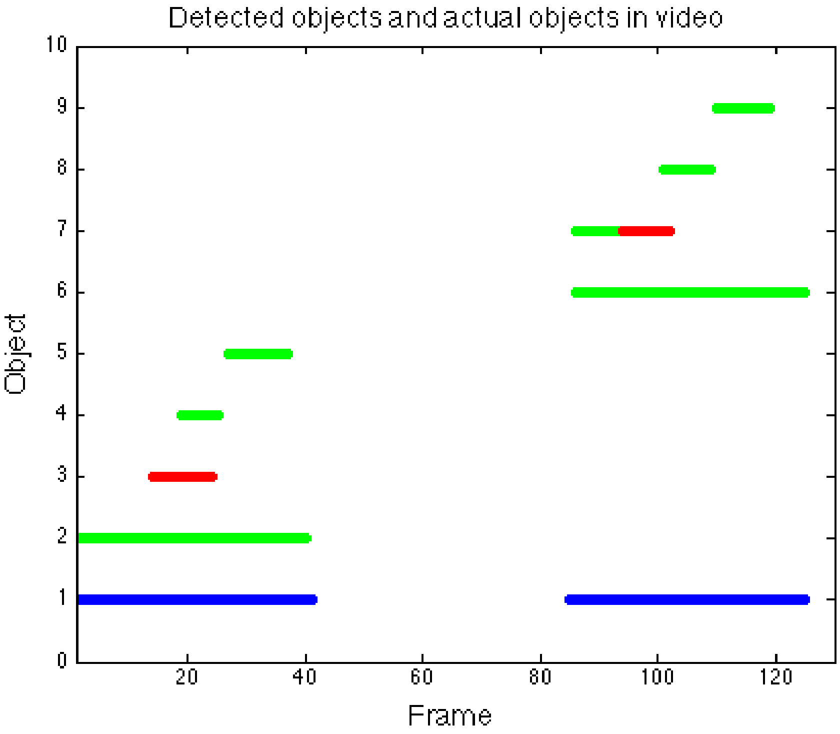


Figure 4.9: The detected and tracked moving objects in the first 130 frames of video 1, using the GMM, OSM, MSA, TFPM and RFD. The figure shows all the detections and the actual objects in the video. The actual moving object is shown with the colour blue, the detected and tracked moving objects that are part of an actual moving object is shown in green colour and false detections are shown in red colour.

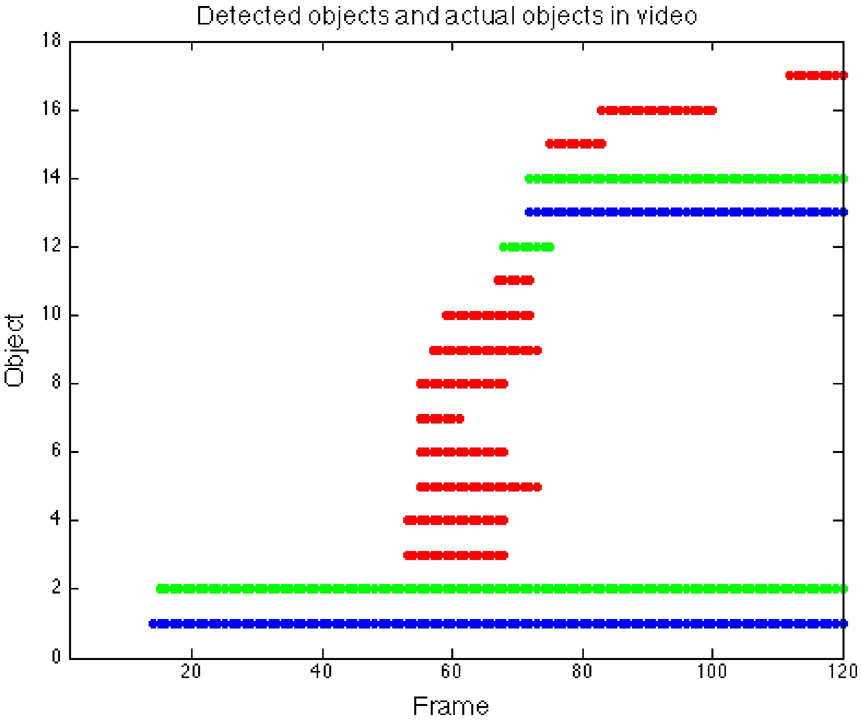


Figure 4.10: The result from detecting and tracking moving objects in the first 120 frames in video 2, using the GMM, OSM, MSA, TFPM and RFD. The figure shows all the detections and the actual objects in the video. The actual moving object is shown with the colour blue, the detected and tracked moving objects that are part of an actual moving object is shown with green colour and false detection is shown with red colour.

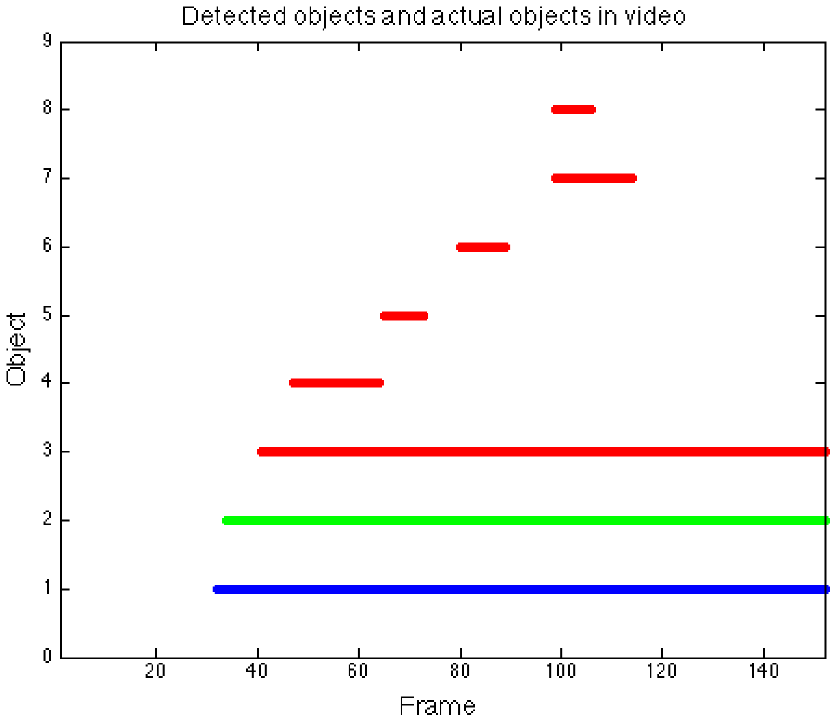


Figure 4.11: The detected and tracked moving objects in the first 150 frames of video 3, using the GMM, OSM, MSA, TFPM and RFD. The figure shows all the detections and the actual objects in the video. The actual moving object is shown with the colour blue, the detected and tracked moving objects that are part of an actual moving object is shown in green colour and false detections are shown in red colour.

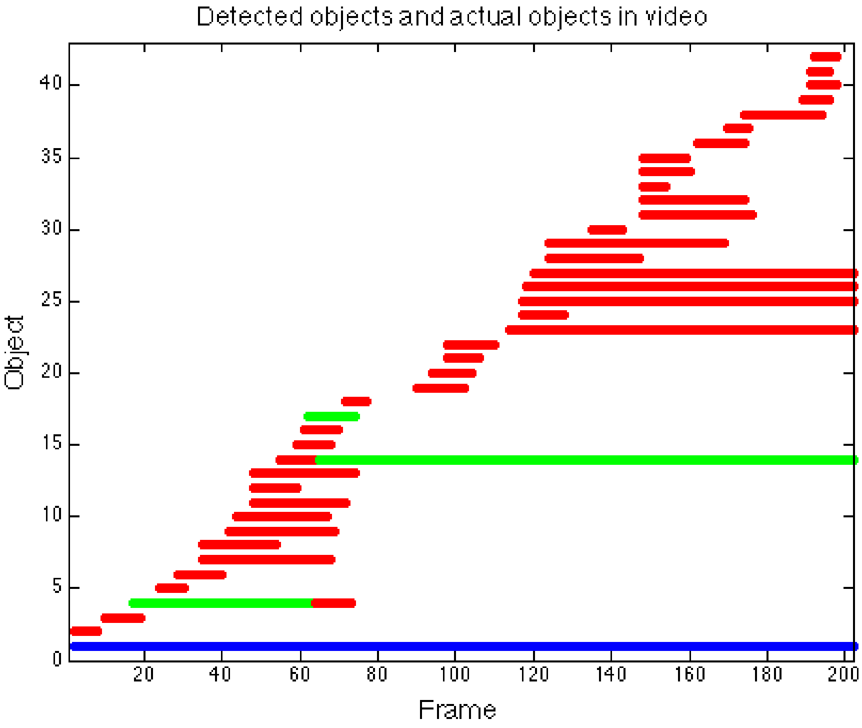


Figure 4.12: The detected and tracked moving objects in the first 202 frames of video 4, using the GMM, OSM, MSA, TFPM and RFD. The figure shows all the detections and the actual objects in the video. The actual moving object is shown with the colour blue, the detected and tracked moving objects that are part of an actual moving object is shown in green colour and false detections are shown in red colour.

produces an almost accurate estimation of the real trajectory.

Finally, in figure 4.17, the result for the fourth video is presented. Here, MMR has been used to merge together three detections, remove the begining of the trajectory of one of them and remove the end of the trajectory of another. All detections are shown in figure 4.12. The video contains a lot of camera shakes and the moving object is under occlusion a few times. Therefore, the methods only manage to estimate the real trajectory roughly.

4.2 Automatic evaluation of smooth pursuit detections

In this section, the results of the automatic evaluation of smooth pursuit detections can be seen in figures 4.18-4.20. The result shows the eyes ability to follow the moving objects in video sequences with the eye movement smooth pursuit. In figures 4.18-4.20, the colour purple shows where the eyes are preforming smooth pursuit movements. Panel a) shows the distance between the point of gaze and the centre point of the moving object estimated with the proposed method. Here, the mean distance is also calculated from the first to the last moment that smooth pursuit movements occurs. In panel b) the distance between the point of gaze and the real centre point of the moving object that has been manually determined is shown. Finally in c), the difference between the result when using the proposed method and manually determined measurement is shown.

The result for the first moving object in the first video can be viewed in figure 4.18. In a), the mean distance was calculated to 179 pixels. a) and b) seem to have a similar results. The amount of error in c) varies, and by comparison with figures 4.13 c) and d), the amount of error is larger in roughly the same places as the area error is large and by that the centre position is more incorrect.

Figure 4.19 is showing the result for the second moving object in first video. In a), the mean distance was calculated to 193 pixels. Also here, for the second moving object, panel a) and b) shows a similar result, as the first moving object. The amount of error in c) seems to be a little bit lower, compared to the result for the first object. The error level is larger approximately where the objects' area error is greater meaning that the centre position is more displaced, which can be seen in figures 4.14 c) and d).

The result for the fourth video can be seen in figure 4.20. In a) the mean

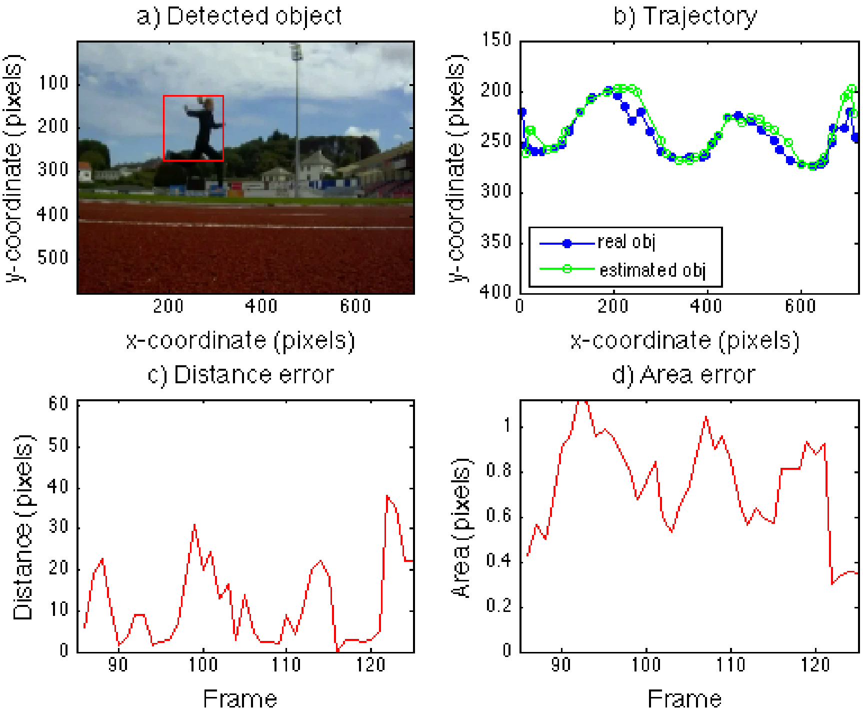


Figure 4.13: The result for the first detected and tracked moving object in video 1, when MMR has been used. The estimated measurements of the moving object are compared to the real measurements of the moving objects. In a), frame 20 of the detected object. In b), the moving objects centre position in each frame for both the real and estimated measurements. In c) the difference, calculated with the $L_2 norm$, between the real and estimated measurement of the centre position in the different frames. Finally in d), the ratio between the estimated and the real areas of the moving objects in the different frames.

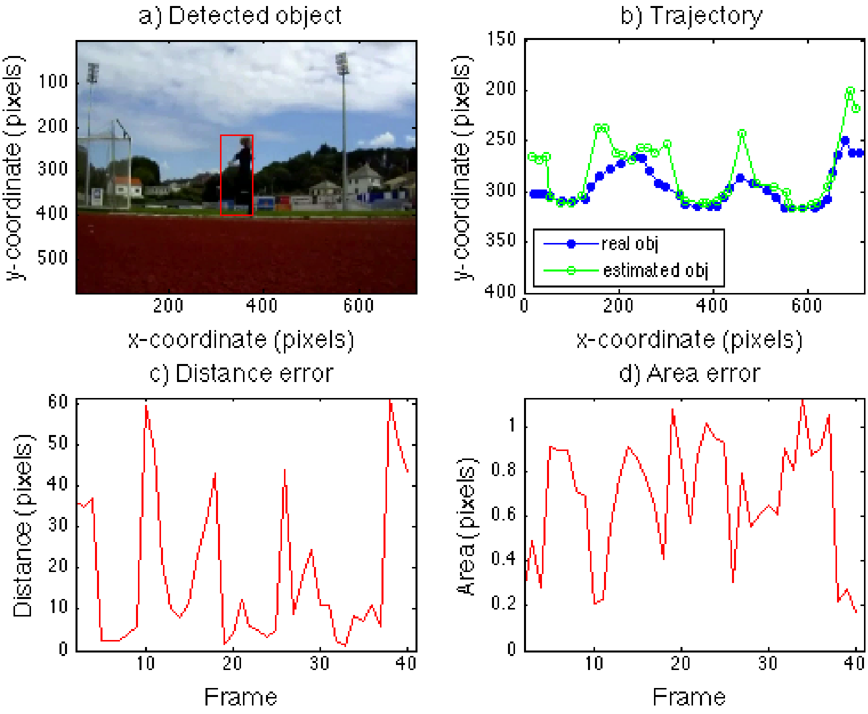


Figure 4.14: The result for the second detected and tracked moving object in video 1, when MMR has been used. The estimated measurements of the moving object are compared to the real measurements of the moving objects. In a), frame 100 of the detected object. In b), the moving objects centre position in each frame for both the real and estimated measurements. In c) the difference, calculated with the L_2 norm, between the real and estimated measurement of the centre position in the different frames. Finally, in d), the ratio between the estimated and the real areas of the moving objects in the different frames.

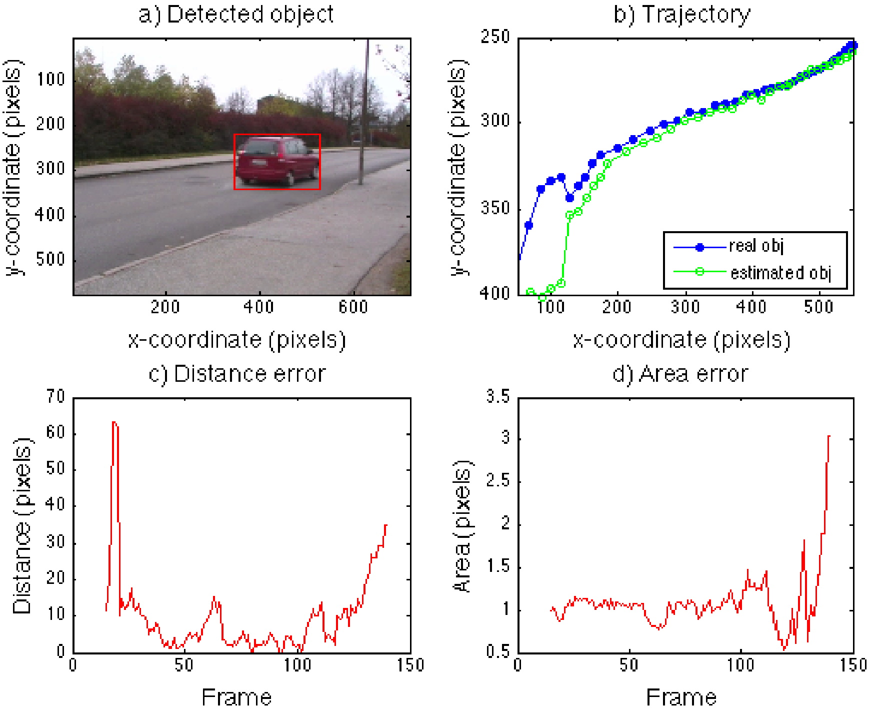


Figure 4.15: The result for the first detected and tracked moving object in video 2, when MMR has been used. The estimated measurements of the moving object are compared to the real measurements of the moving objects. In a), frame 40 of the detected object. In b), the moving objects centre position in each frame for both the real and estimated measurements. In c) the difference, calculated with the L_2 norm, between the real and estimated measurement of the centre position in the different frames. Finally, in d), the ratio between the estimated and the real areas of the moving objects in the different frames.

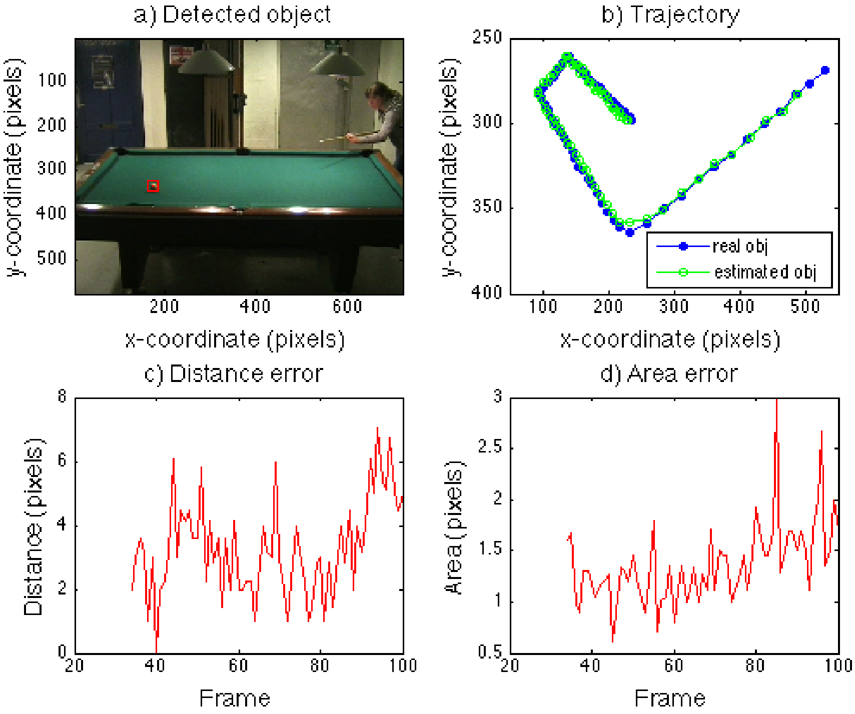


Figure 4.16: The result for the first detected and tracked moving object in video 3, when MMR has been used. The estimated measurements of the moving object are compared to the real measurements of the moving objects. In a), frame 50 of the detected object. In b), the moving objects centre position in each frame for both the real and estimated measurements. In c) the difference, calculated with the L_2 norm, between the real and estimated measurement of the centre position in the different frames. Finally, in d), the ratio between the estimated and the real areas of the moving objects in the different frames.

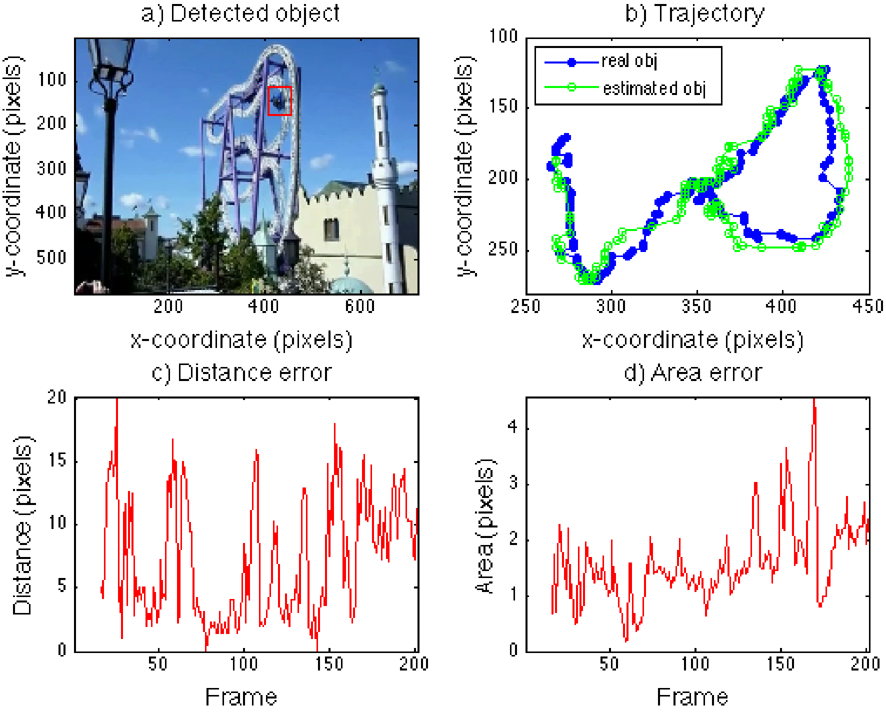


Figure 4.17: The result for the first detected and tracked moving object in video 4, when manual sorting has been used. The estimated measurements of the moving object are compared to the real measurements of the moving objects. In a), frame 50 with the detected object. In b), the moving objects centre position in each frame for both the real and estimated measurements. In c) the difference, calculated with the L_2 norm, between the real and estimated measurement of the centre position in the different frames. Finally, in d), the ratio between the estimated and the real areas of the moving objects in the different frames.

distance in was calculated to 176 pixels. In a) and b) the results vary a little bit at many times, but it still contains the same main traits. In c), the error level is fluctuating a lot, which it also does in figure 4.17 c) and d) that shows the centre point position error and the area error of the moving object.

Based on the result in this section, the mean distance between the point of gaze and the centre point of the moving object, estimated with the proposed method, is 183 pixels.

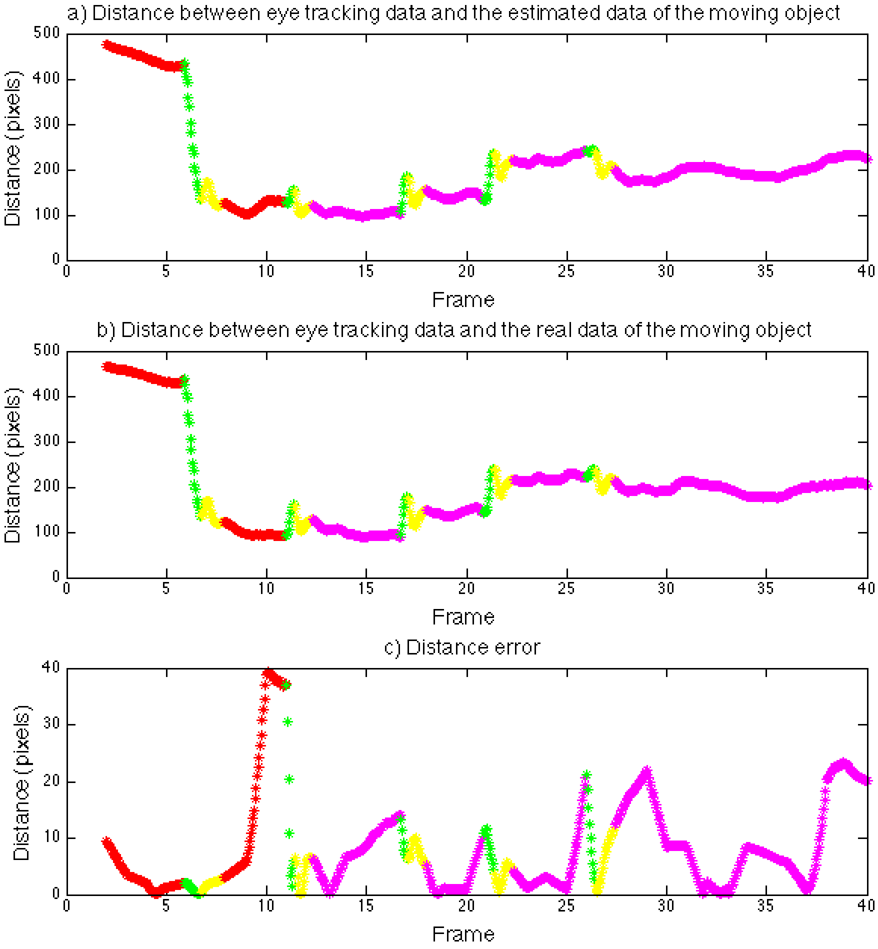


Figure 4.18: Result for the automatic evaluation of smooth pursuit detections when it is applied to the first object in the video 1. The smooth pursuit movements of the objects are marked with the colour purple. In a) the distance between eye tracking data and the estimated data of the moving object. In b) the distance between the eye tracking data and the manually measured data of the moving object. In c) the distance error between the values in figures a) and b).

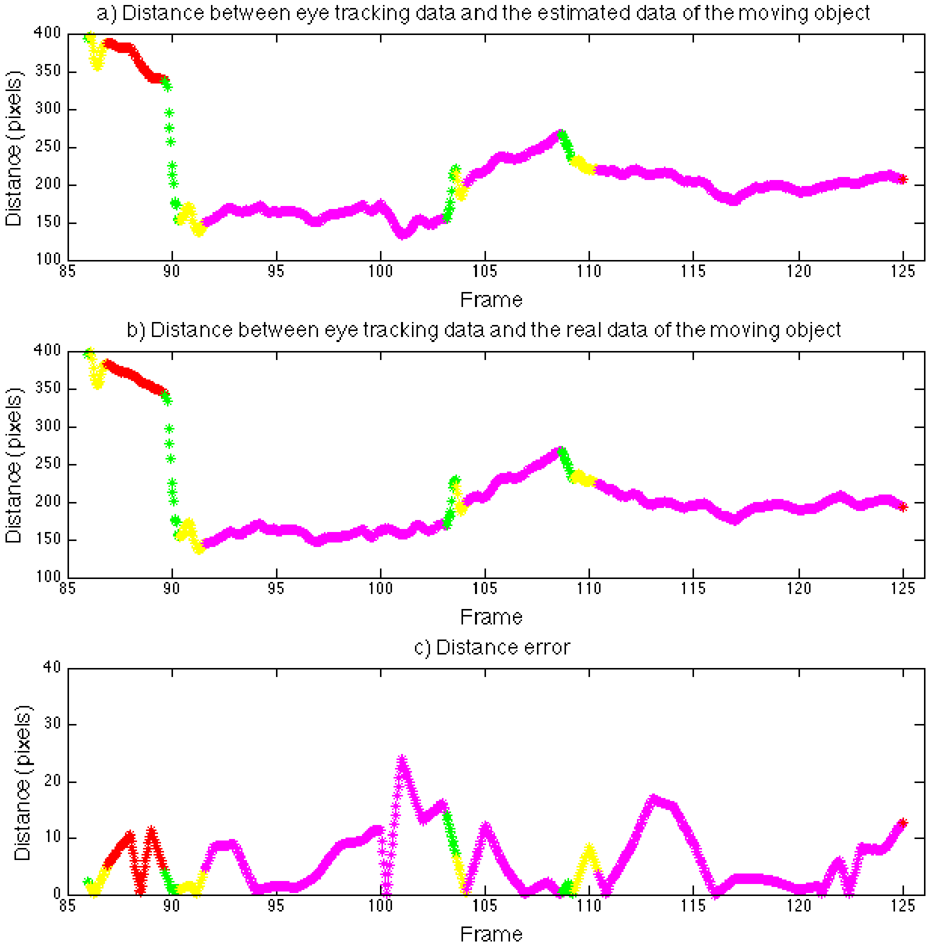


Figure 4.19: Result for the automatic evaluation of smooth pursuit detections when it is applied to the second object in the video 1. The smooth pursuit movements of the objects are marked with the colour purple. In a) the distance between eye tracking data and the estimated data of the moving object. In b) the distance between the eye tracking data and the manually measured data of the moving object. In c) the distance error between the values in figures a) and b).

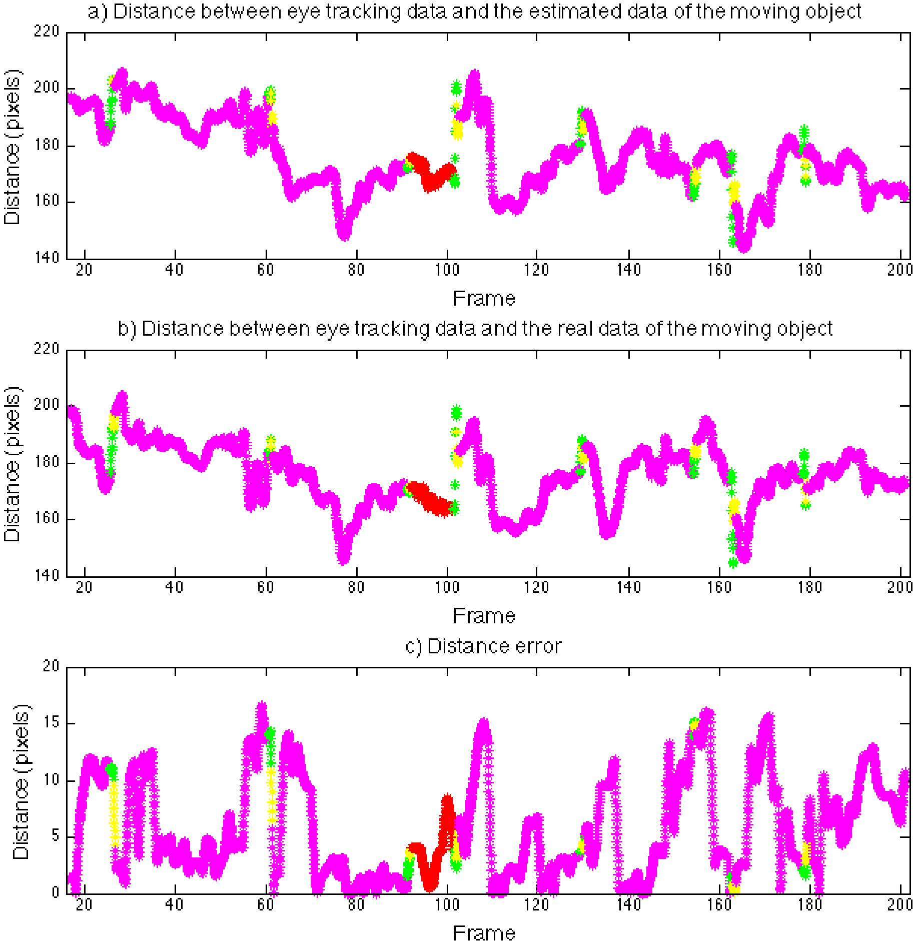


Figure 4.20: Result for the automatic evaluation of smooth pursuit detections when it is applied to the object in the video 4. The smooth pursuit movements of the objects are marked with the colour purple. In a) the distance between eye tracking data and the estimated data of the moving object. In b) the distance between the eye tracking data and the manually measured data of the moving object. In c) the distance error between the values in figures a) and b).

Chapter 5

Discussion

In the development process of the detection and tracking method of moving objects in video, the major drawback was that it took a long time to run the programs for every single frame. This made every test of the program very time consuming and it limited the number of test that were possible to preform and also the number of frames included in every test. Therefore, it was difficult to make sure that the method would still have the same behaviour after running a long time.

The performance of the Guassian mixture model (GMM) gave adequate result when the background was static and the background and the moving object differs in colour. If the background is dynamic, it seems to be able to adapt to it to some extent, but the method still suffers from false detections. The GMM was not able to handle camera shakes movement.

The Tracking feature point method (TFPM) was able to reduce the number of false detections due to camera shakes and other movements. The method only calculates the translational movements of the camera. Therefore, one improvement that could be made to the method is to also include calculations of the rotational camera movements.

The Mean shift algorithm (MSA) seemed in most cases to be able to decide which of all detections of moving objects that belong to the same objects in two subsequent frames. The method is updating the object information based on the current frame only. If the updating of the object information instead depended on a number of past frames, it would probably give more reliable and stable results. The moving objects' feature characteristic probability and area could then be updated in a recursive way. The MSA sometimes fails when the moving object is under occlusion and when moving objects are passing each other and becomes mixed up. The performance of the method would probably be improved if another method was added, like the Kalman filter or Particle filter.

A major influence of the resulting performance had the Removal of false detection (RFD), because it decreased the number of the false detections significantly in some cases. This method was added in the end of the development process because it turned out to be necessary due to all the false detections that the detection and tracking method suffered from.

The RFD could therefore be improved with a method that is better developed. Since the aim of this thesis is to detect and track moving objects that the eyes could follow with the eye movement smooth pursuit, one improvement of the method could be to evaluate the detected objects trajectories. Based on the trajectories traits, determine which object that could belong to moving objects and which ones are more likely to be noise. In combination, the size change could also be investigated when a moving objects area does not change as irregularly as the area of noise can change.

Because of the Manual modification of the result (MMR) method was added, false detected objects did not affect the end result. Therefore, the aim of the detection and tracking method was more to detect the entire area of the moving objects rather than decrease the false detections. Furthermore, if the moving object was divided into several detections or the detection was lost during a number of frames the method was able to compensate to some extent for that.

In the proposed method, only the width and height of the moving objects have been estimated. But, in the Object segmentation method (OSM), all the pixels that are included in the objects are determined and could be used to create binary images of the objects. It would then have strong enough edges so that an Active contour-based tracking method, that is used to calculate the contour of the objects, would unlikely fail.

Overall, the result from the detection and tracking method of moving objects in video sequences was able to detect and track the moving object in most part of the videos that were used in the evaluation process. The entire area of the objects sometimes fails to be detected specially when the video scenarios are more complex.

The automatic evaluation of smooth pursuit detections method showed a similar result when using the data of the moving objects that was manually determined and when the data that was estimated with the proposed method. For evaluating how well the eyes are following a moving object, the distance of the point of gaze and the centre position were measured. This may give misleading results, because the eyes may not always follow the centre position of an object but some other part of the object. If the contour was estimated for the object, the automatic evaluation of smooth pursuit detections method could be developed further by calculating the distance between the point of gaze and the object area.

Chapter 6

Conclusion

The purpose of this thesis was to develop a method for automatic evaluation of smooth pursuit detections when viewing moving objects in videos. The first and main part of the work was to develop a method that detected and tracked moving objects in videos, based on investigations of already existing methods in field. A major issue in the developing process was that the computational cost was very high for the method. So, the ability to test the method was limited, due to that every test was so time consuming. The detection and tracking method of moving objects in video sequences gave adequate results when the background was static and the colour of the moving object differs from the background. As the environment becomes more dynamic and complex, the method suffers from false detections and sometimes fails to detect the entire object area. Since the method did not give sufficient results in all video scenarios, a manual modification of the result was performed trying to ensure that the result could still be used for the evaluation of eye tracking data.

The second part of the thesis was to develop an automatic evaluation of a smooth pursuit detection method based on the detected and tracked moving objects and eye tracking data that has already been previously annotated. In the method, the distance between the points of gaze and the moving objects centre point was measured. The mean distance was calculated to 183 pixels. The eyes may not always follow the centre position of an moving object, but rather some other part of the object. Hence, the method gives more satisfactory result when the objects are small.The evaluation of how well the eyes are following a moving object could therefore be developed further, by estimating the contour of the objects and then calculate the distance between the point of gaze and the object area.

Chapter 7

References

[1] J. Shajeena and Dr. K. Ramar. A Novel Way of Tracking Moving Objects in Video Scenes, 2011.

[2] Nicole S. Love and Chandrika Kamath. An Empirical Study of Block Matching Techniques for the Detection of Moving Objects, 2006.

[3] C. Kamath, A. Gezahegne, S. Newsam and G. M. Roberts. Silient Points for Tracking Moving Objects in Video, 2005.

[4] Qiang Chen, Quan-Sen and Pheng Ann Heng. Two-Stage Object Tracking Method Based on Kernel and Active Contour, 2010.

[5] Jiaming Zhang and Chi Hau Chen. Moving Objects Detection and Segmentation In Dynamic Video Backgrounds, 2007.

[6] Sen-Ching S. Cheung and Chandrika Kamath. Robust techniques for background subtraction in urban traffic video, 2004.

[7] Liang Zhang, Xiangming Wen, Wei Zheng and Bo Wang. An Algorithm for Moving Semantic Objects Trajectories Detection in Video, 2010.

[8] Tao Liu and Xiao-ping Cheng. Improved Mean Shift Algorithm for Moving Object Tracking, 2010.

[9] Saira Saleem Pathan, Ayoub Al-Hamadi and Bernd Michaelis. Intelligent Feature-guided Multi-object Tracking Using Kalman Filter, 2009.

[10] Weeratunga S. and C. Kamath. An investigation of implicit active contours for scientific image segmentation, 2004.

[11]] Weng Muyn, He Mingyi and Zhang Yifan. An Adaptive Implementation of the Kernel-Based Object Tracking Method, 2006.

[12] Ahmad Ali and Dr Sikander Majid Mirza. Object Tracking using Correlation, Kalman Filter and Fast Means Shift Algorithms, 2006.

[13] Da Tang and Yu-Jin Zhang. Combining Mean-shift and Particle Filter for Object Tracking, 2011.

[14] Simone Palazzo and Concetto Spampinato. Object Tracking: State of the Art and Online Performance Evalution, 2011?.

[15] Li Ying-hong, Pang Yi-gui, Li Zheng-xi and Liu Ya-li. An Intelligen Tracking Technology based on Kalman and Mean Shift Algorithm, 2010.

[16] Zhiwen Chen, Jianzhong Cao, Yao Tang and Linao Tang. Tracking of Moving Object Based on Optical Flow Detection, 2011.

[17] Shengluan Huang and Jingxin Hong. Moving Object Tracking System Based On Camshift And Kalman Filter, 2011.

[18] Ebrahim Emami and Mahmood Fathy. Object Tracking Using Improved CAMShift Algorithm Combined with Motion Segmentation, 2011.

[19] Ben-Hsiang Do and Shih-Chia Huang. Dynamic Background Modeling Based On Radial Basis Function Neural Networks For Moving Object Detection, 2011.

[20] Meng Liu, Chengdong Wu and Yunzhou Zhang. A Review of Traffic Visual Tracking Technology, 2008.

[21] Jong Sun Kim, Dong Hae Yeom and Young Hoon Joo. Fast and Robust Algorithm of Tracking Multiple Moving Objects for Intelligent Video Surveillance System, 2011.

[22] Chen Peijang. Moving Object Detection Based on Background Ex-

traction, 2009.

[23] Yunfei Wang, Zhaoxiang Zhang and Yunhong Wang. Moving Object Detection in Aerial Video, 2012.

[24] Linnéa Larsson, Master's Thesis: Event detection in eye-tracking data, 2010.

[25] Mingxin Jiang, Min Li and Hongyu Wang. A Robust Combined Algorithm of Object Tracking Based on Moving Object Detection, 2010.

[26] Simone Palazzo and Concetto Spampinato. Object Tracking: State of the Art and Online Performance Evaluation, 2011.

[27] Aiyn Yan, Jingjiao Li, Aixia Wang and Jiao Wang. Color Moving Object Segmentation Based on Mixture Gaussian Models, 2010.

[28] Ping Gao, Xiangju Sun and Wei Wang. Moving Object Detection Based on Kirsch Operator Combined with Optical Flow, 2010.

[29] Chris Stauffer and W.E.L Grimson. Adaptive Background Mixture Models for Real-Time Tracking, Computer Vision and Pattern Recognition, 1999.

[30] Yizong Cheng. Mean Shift, Mode Seeking, and Clustering, 1995.

[31] Zdenek Kalal, Krystian Mikolajczyk and Jiri Matas. Forward-Backward Error: Automatic Detection of Tracking Failures, 2010.

[32] Bruce D. Lucas and Takeo kanade. An Iterative Image Registration Technique with an Application to Stereo Vision, 1981.

[33]Jianbo Shi and Carlo Tomasi. Good Features to Track, 1991.

LUND
UNIVERSITY

Series of Master’s theses
Department of Electrical and Information Technology
LU/LTH-EIT 2015-453

http://www.eit.lth.se