UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

FAKULTÄT FÜR
ELEKTROTECHNIK,
INFORMATIK UND
MATHEMATIK

Vision Based Tracking in Team Sports

Von der Fakultät für Elektrotechnik, Informatik und Mathematik
der Universität Paderborn

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte Dissertation

von

M.Sc. Emad Monier Mosaad Ibrahim

Erster Gutachter:	Prof. Dr.-Ing. Ulrich Rückert
Zweiter Gutachter:	Prof. Dr. Hans Kleine Büning
Tag der mündlichen Prüfung:	28.06.2011

Paderborn 2011

Diss. EIM-E/278

Abstract

Object tracking is an important task in many computer vision applications including surveillance, gesture recognition, vehicle tracking, augmented reality, video compression, and medical imaging. The tracking becomes more complex when it involves human beings especially in the sports domain where there is a lot of interaction between players, which results in occlusion, clutter and dynamic changes in the scene. Moreover, tracking players in official games should be markerless due to the game rules which add more complexity to the problem.

In this thesis work a tracking system has been designed and implemented to track players in indoor sports such as basketball and handball. The system is able to record games and trainings using two high quality digital cameras to track the players offline, analyze the tracking results and produce visualizations for analysis results.

The use of template matching and particle filter techniques in tracking sport players has been investigated. A system for recording, tracking and visualization has been developed. Before the system is used in practice many tests have been performed to assure the validity and accuracy of the tracking.

In cooperation with the colleagues in the System and Circuit Technology group, the acquisition subsystem, in addition to the video capturing of the game, was able to record some physiological data such as heart beat rate, which is acquired by a special sensor designed in our group and transferred wirelessly to a computer for recording. The tracking of both position and physiological data has provided new possibilities for analysis of indoor sport games. This data has been used for analysis by Applied Mathematics and Sport Medicine research groups in addition to Paderborn basketball team.

Zusammenfassung

Objekterkennung und -verfolgung sind wichtige Aufgaben in vielen Anwendungen der Bildverarbeitung einschließlich der (Verkehrs-) Überwachung, Gestenerkennung, erweiterten Realität, Videokompression und medizinischen Bildgebung. Die Objektverfolgung wird komplizierter in Verbindung mit Menschen, vor allem im Bereich Sport, wo es viele Kontakte zwischen den Spielern gibt. Die Ergebnisse sind Vermengung, Unordnung und dynamische Veränderungen in dem Vorgang. Aufgrund der Spielregeln sollte die Objektverfolgung ohne Markierung erfolgen.

In dieser Dissertation wurde ein Objektverfolgungssystem konzipiert und umgesetzt, um Spieler von Hallensportarten, wie z.B. Basketball oder Handball, zu verfolgen. Das System ist in der Lage, Spiele und Trainingseinheiten mit zwei hochauflösenden digitalen Kameras zu erfassen, Spieler offline zu verfolgen, die Ergebnisse zu analysieren und Visualisierungen für die Analyseergebnisse zu produzieren.

Der Einsatz von Mustererkennung und Partikelfilter -Techniken zur Objektverfolgung von Sportlern wurde untersucht. Ein System zur Erfassung, Verfolgung und Visualisierung wurde entwickelt. Vor dem praktischen Einsatz des Systems wurden viele Tests durchgeführt, um die Gültigkeit und Genauigkeit des Systems zu zeigen.

In Zusammenarbeit mit Kollegen der Fachgruppe Schaltungstechnik wurde zusätzlich zum Video-System eine Komponente zur Aufnahme physiologischer Daten, wie der Herzfrequenz, entwickelt. Die Objektverfolgung bietet zusammen mit den physiologischen Daten neue Möglichkeiten für die Analyse von Mannschaftssportarten.

Contents

List of Symbols

$W_k(t)$	weight of the position used to calculate position prediction	33
f	frame rate	34
t	time variable	38
$\mathbf{S}_t$	unobservable state variable at time t	38
$\mathbf{Z}_t$	observable (measurable) evidence variables	38
$\mathbf{s}_{t-1}$	state model at time $t-1$	38
$\mathbf{z}_{0:t-1}$	all measurements up to time $t-1$	38
$p(\mathbf{s}_{t-1}\|\mathbf{z}_{0:t-1})$	posterior of state s_{t-1} given measurements $\mathbf{z}_{0:t-1}$)	38
$p(\mathbf{s}_t\|\mathbf{s}_{t-1})$	probability that a state changes from $\mathbf{s}_{t-1}$ to $\mathbf{s}_t$	38
$p(\mathbf{z}_t\|\mathbf{s}_t)$	likelihood of state $\mathbf{s}_t$ given measurement $\mathbf{z}_t$	38
N_p	number of particles	39
$w_{t-1}^{(i)}$	weight of particle i at time $t-1$	39
$\mathbf{x}_t = (x_t, y_t)^T$	position of the player at time t	40
$\mathbf{e}_t = (a_t, b_t)^T$	bounding ellipse of the tracked target (player)	40
$\mathbf{A}_t$	appearance matrix	40
l	target label	40
$\mathbf{s}_t^l = (\mathbf{x}_t^l, \mathbf{e}_t^l, \mathbf{A}_t^l)^T$	the state of target l at time t	40
$P(\mathbf{S}_t\|\mathbf{S}_{t-1})$	state transition model	41
$p(\mathbf{x}_t\|\mathbf{x}_{t-1})$	the evolution model of the position	41
$p(\mathbf{e}_t\|\mathbf{e}_{t-1})$	the translation model of the bounding ellipse	42
$\bar{\mathbf{h}}^l$	color histogram of the l-model	43
h_k^l	the probability of feature (bin) k in color histogram $\mathbf{h}$	44
C^l	normalization constant used in computing the color feature probability	44
$\Pi(.)$	weighting function that gives higher weights to the pixels closer to the center of the ellipse	44
δ	Kronecker delta	44
$b(x_a)$	a function that associates the given pixel to its corresponding histogram bin	44
$M(.)$	a mask function that gives a zero weight to background pixels	44

d_B	similarity between two histograms	44
$\rho(\mathbf{h}_1, \mathbf{h}_2)$	Bhattacharyya coefficient	44
S	Set of points in Euclidean space	51
Θ_{new}	threshold of a new cluster, used in c-means clustering	54
Θ_{merge}	threshold of merging two clusters, used in c-means clustering	54
r	distance to the center of the corrected (*Defish*) image in pixels	81
$\grave{r}$	distance to the center of the distorted (*Fisheye*) image in pixels	81
f	the focal length (in pixels)	81
$\grave{p} = (\grave{x}, \grave{y})$	pixel position in a distorted image	81
$p = (x, y)$	pixel position in an undistorted image	81
$\grave{p}_c = (\grave{x}_c, \grave{y}_c)$	center of the distorted image	81
$p_c = (x_c, y_c)$	center of the undistorted image	81
$p_h = (x_h, y_h)$	position of the head or the body of the player on the image	85
h_p	player height in meters	85
$p_f = (x_f, y_f)$	player feet position on the image	85
R_c	the relative resolution of the camera (pixel/meter)	86
H	height factor	90
E	tracking algorithm rate of correction. The number of corrections in one minute	98
N_e	Number of tracking errors in a video sequence	98
T_e	Time needed to manually correct one tracking error (seconds)	98

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The aim of CV (Computer Vision) is to enable machines to understand the contents of digital images. What by understanding meant is to extract information from image data which can be used for further analysis or as an input for other processes. There are many applications for CV, for example in industry it can be used to inspect defect objects. In the medical domain the CV techniques can be used to identify important phenomena or events in a medical image. Structure from motion is an application which concerns finding the three-dimensional structure by analyzing the motion of an object over time. Tracking of moving objects is of great importance in security applications. The focus of this thesis will be on using CV techniques to track players in team sports.

Tracking of moving objects in video sequences is an important task of many CV applications. Tracking is the problem of generating an inference about the motion of an object given a sequence of images [1]. In simple words it is the process of locating the object in each frame of the image sequence. Some typical applications of video-based tracking are:

- ***Visual surveillance:*** where people or traffic are tracked to detect unusual situations.

- ***Tracking of small laboratory animals:*** such as insects and rodents (i.e. mice and rats) in order to study interactions of natural multi-agent systems.

- ***Automatic video annotation:*** the aim is to add graphical contents to a video which contains moving objects or persons.

- ***Ambient intelligence:*** where tracking together with other technologies such as telecommunications helps to assist people in their everyday life.

- ***Cognitive systems:*** where tracking is used to learn more about dynamic properties of different objects in the environment.

- ***Tracking of robots:*** where the tracking is used to find the path of the minirobots in order to analyze and debug experiments.

- ***Analysis of sport events:*** to extract positional data of athletes during match or training which is used by sport experts to analyze the performance of the players.

Looking at the literature, it can be seen that many algorithms have been proposed to solve the tracking problem. They can be classified into two main categories: deterministic methods and stochastic methods.

Deterministic methods use an exhaustive search for local maxima of a similarity cost function between the template image and the searched image. On the other hand, stochastic methods use the state space to model the underlaying dynamics of the tracking system.

1.1 Goals and Motivations

Sport is attracting the minds of millions of people around the world. To CV researchers it is a rich and diverse domain which provides a huge source of footage from which to work. This thesis has two goals. First, to propose solutions for sport player tracking in indoor sport games based on CV techniques in order to acquire the position data in sufficient accuracy, in real time and automatically without user intervention. The position data should be accurate enough to be used for further analysis. Second, to provide a reliable software system that can be used by sport scientists and trainers to acquire tracking information.

Three primary motivations are powering the work of this thesis.

Firstly, sport scientists are very interested in being able to know how much distance the players have run and how fast they have moved during the game or the training session. This information would allow more specific training to be designed to suit individual players. Regarding CV, the tracking of sport players from video is a challenging domain in which players interact, occlude, make unpredictable quick body movements and move in non-linear fashion covering large distances.

Secondly, there is a lack of tracking systems that fulfill the requirements of sport scientists and trainers in terms of easily recording the games, tracking the players and providing visualization, summaries and statistics for the tracking data.

Thirdly, acquisition of positional data is basic for higher level analysis of team game interactions, stability of the team as a system and modeling cooperative, collaborative and adversarial actions of individual players and the whole team.

1.2 Problem Statement

This thesis work is a part of a larger project called *Intelligent Sports Wear and Automatic Analysis of Team Sports* which is under development at the System and Circuit Technology research group at the Heinz Nixdorf Institute, University of Paderborn. Figure 1.1 shows a block diagram of the project. The darker block *Tracking of Players in Video Data* is the focus of this thesis work. The whole project concerns the tracking of both internal and external conditions of the player during the game or training. Some work has been also done in other blocks such as online visualization of sensor data and offline visualization of the position data. The output of the player position tracking is coordinates of the players in meters. The two research groups *Sport Medicine* and *Applied Mathematics* at the University of Paderborn use position data to perform further analysis.

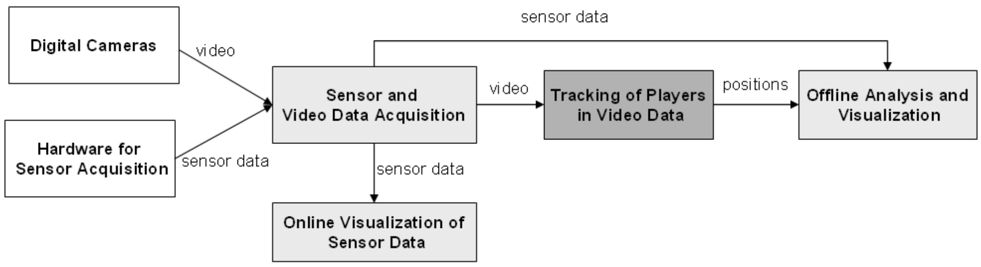


Figure 1.1: Block diagram of the project *Intelligent Software and Automatic Analysis of Team Sports*.

The analysis system shown in Figure 1.1 , consisting of a high resolution video system together with a wireless sensor network, is used for collecting position and physiological data of sport players during training or competitions. The combination of the two data streams provides a new type of performance analysis and visualization solution for indoor team sports.

A training session or game is captured by a video system consisting of two cameras which are placed at the hall ceiling. The video data is post-processed offline in order to identify positions of the players and to track all players on the field by means of CV techniques. A GUI (Graphical User Interface) has been developed in order to enable the user to interact with the tracking algorithm for error correction. From the tracking data the path of each player as well as the corresponding calculated information such as speed and distance can be obtained. Trainers need to instantly (during the game) know the speed and the distance covered by the players during the game. So, players are equipped with an additional off-the-shelf acceleration sensor which is able to transmit the data via the wireless network after which it can be visualized.

For the recording of physiological data of the athletes during training or a game, a compact wearable module has been developed [83]. It can be easily mounted onto the sport shirt. The heart activity is measured by electrodes that are integrated in the shirt. The signal generated by these electrodes is processed in real-time. The calculated heart rate is transmitted via a new wireless technology called ANT to a central computer. Additional physiological data, which can be optionally recorded, are skin temperature and skin conductance.

The difficulty of tracking moving objects in video comes from the fact that tracked objects are usually interacting, occluding and have a non-rigid shape. This is more clear when the task involves tracking humans where interactions and occlusions usually occur. The problem becomes more complex when it involves tracking players in sport activities, especially in team sports where multiple players interact, occlude, make sudden movements and try to move in an unpredicted manner in order to confuse the opponents. In the sports domain, especially in team sports, the tracking of players is a complex task due to different reasons related to problems in acquisition of the images such as:

- Using special camera lenses such as fisheye in order to capture the whole playing field will cause distorted images. This makes it more complex to calculate the positions.

- Errors in camera installation require calibration and correction because they will affect the calculations of player position in meter.

- The use of multiple cameras requires a fusion between different camera views.

There are also problems due to the interaction between players during the game, this includes:

- Unpredictable, nonlinear and fast-changing player movement in order to deceive players in the opponent team.

- Collision and blocking in order to prevent the opponent to pass or shoot the ball.

- The players in one team look the same, especially when using the overhead camera.

In addition there are traditional problems in CV such as changing lighting illumination and noise in images.

1.3 Contributions

The contributions of the thesis work include the following:

- Methods of tracking that are based on deterministic and stochastic techniques have been analyzed.
- New techniques for optimization of image pre-processing have been implemented.
- The implementation of the tracking algorithms has been done on different hardware platforms such as multiprocessors and graphic processors.
- Design of new test patterns and development of benchmark data that enable to prove the validity, accuracy and efficiency of tracking.
- A software system with a user friendly interface that enables normal user to record games, track the players, visualize and analyze the tracking data has been developed.

1.4 Thesis Organization

The rest of this thesis is organized as follows:

- Chapter 2 presents an overview of CV based tracking with a focus on sport player tracking.
- Chapter 3 presents the deterministic-based tracking methods that are used in tracking and basically the template matching techniques.
- Chapter 4 presents the probabilistic-based tracking methods with focus on Particle Filter based tracking.
- Chapter 5 presents the techniques used for handling several tracking targets.
- Chapter 6 shows the development of a software system that enables normal users to use the tracking algorithms developed during the course of this thesis.

- Chapter 7 presents an experimental study to show various testing results of the tracking algorithms as well as the proof of the validity, reliability and objectivity of the developed tracking methods.
- Chapter 8 gives the summary, conclusion and suggested future work.

Chapter 2

Sport Player Tracking Overview

This chapter presents an overview of the CV based tracking methods with a focus on sport player tracking. The chapter is organized as follows: Section 2.1 presents the tracking as a computer vision problem and with a brief overview of the stages of processing and basic definitions. Section 2.2 gives an overview on the state of the art in player tracking including background/foreground segmentation, encoding visual features and dealing with multiple players. Section 2.3 shows some commercial systems in the sports domain.

2.1 Tracking as a Computer Vision Problem

The CV system can be seen as a system consisting of several stages or steps which include image acquisition, image processing, scene analysis and result visualization. Figure 2.1 shows the steps of CV system.

The acquisition step means to obtain the images from cameras. In digital cameras various sensor types may be used such as CCD (Charge Coupled Device) and CMOS (Complementary Metal Oxide Semiconductor) sensors [65, 1] and different optical components including fisheye [1] and normal lenses.

[1] A wide-angle lens that takes in an extremely wide, spherical image.

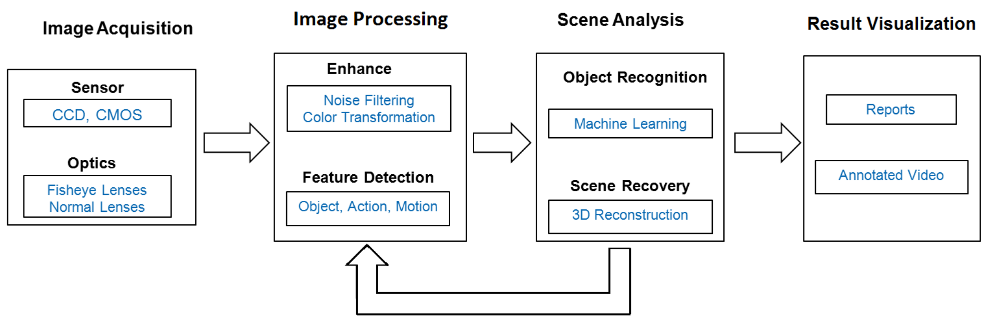


Figure 2.1: Stages of the computer vision system

In case of analog cameras the obtained analog signal should be digitized so that it can be processed by a computer.

The image processing step (also called pre-processing) is the step where the image is enhanced, for example, by filtering of noise or color transformation. Image processing also tries to find important features or parts of the scene where the scene analysis should be applied. The scene analysis step uses various methods and algorithms including machine learning techniques to recognize, track or reconstruct a 3D model of objects in the scene. A final stage could be the visualization and presentation of the results of the whole CV process. The feedback from the scene analysis step to the image processing step provide some information that could help to optimize the image processing step.

Tracking is a typical CV problem, where all the stages of CV are involved. Figure 2.2 is an adaptation of Figure 2.1 to show the steps of tracking players using CV. In the acquisition step shown in Figure 2.2 the cameras used to capture the images are digital CCD cameras with fisheye lenses. The details of the camera setup are described in Section 7.1.

In the image processing block there are several sub-steps which consider the color transformation from Bayer pattern to RGB (Red, Green, Blue) color space which is called *Debayering* (see Section 7.1.1). Another sub-step is white balancing, which enhances the colors by filtering the color channels. The background detection and subtraction step concerns, as described in Section 7.1.1, finding the foreground regions of the image by segmenting the image to motion and non-motion parts.

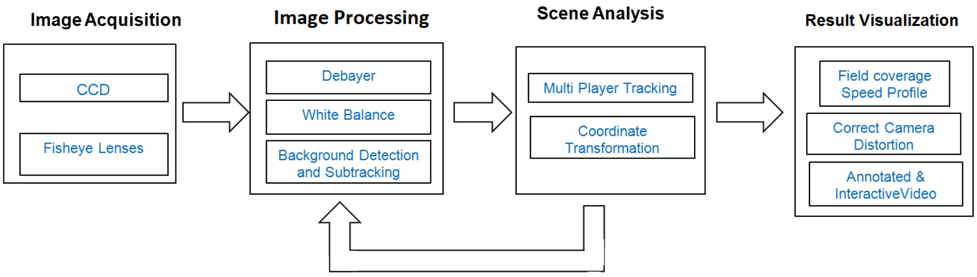


Figure 2.2: Stages of computer vision system for tracking players in team sports.

In the scene analysis step the tracking takes place where multiple players are tracked using the tracking algorithms described in Chapters 3 and 4. In order to acquire the positions of the players in meters on the playing field, the necessary coordinate transformation from image domain to real world domain is performed as shown in Section 7.1.2.

The visualization done in the last step concerns performing different statistics on the tracking data, for example distance and speed profile. Another important visualization is done by generating video with annotations of tracking data which can be static or interactive. Static means the video can be played by a normal video player, but the displayed annotations cannot be changed. Interactive means a special video player is used to play the video and the user can select the kind of visualization he needs. Three preprocessing steps are done before the video is ready for visualization, namely, *Debayering* (described above), *Defishing* and *Warping* the frames. *Defishing* is to correct the fisheye camera distortion caused by the fisheye lenses. *Warping* means to correct the errors caused by inaccurate camera installation which require a perspective transformation.

2.2 State of the Art

2.2.1 Background/Foreground Segmentation

Background/Foreground segmentation means to determine and separate the objects of interest from the image. In the application of sport player tracking, the objects of interest are the players, the referees and the ball. Everything else is considered as background. This separation will reduce the amount of processing needed for tracking by focusing the processing on small areas of the image. The simplest and fastest method to find foreground objects in video sequences, is to subtract each two subsequent frames. Although it is fast, it suffers from the extraction of unwanted regions from the background due to the change in illumination and the moving of non-interesting objects in the background such as trees, clouds and so on [92]. If an empty background can be obtained, it can be subtracted from each frame to get foreground objects but changes in lighting or small camera jitter can cause false background objects to be detected. The simplest approach of differencing two consecutive frames is used in [40] where it is assumed that when no motion is present the last found information about the position of the foreground object is used for tracking.

Seo et al. [97] has used a histogram-based model for the background color and used it to perform the background/foreground segmentation. The disadvantage of this method is that it assumes that the background color is homogeneous which is true in some cases like in tracking soccer players but in indoor sport games like basketball the playing field may contain advertisement. Histogram-based methods have been used by other researchers [36, 53, 49, 4, 55, 60] who track players in soccer matches except [60] who used histograms to detect background in handball and basketball, however only when the background had a similar color to the players shirts. Stauffer and Grimson [95] have developed a method for estimating the background which is based on modeling each pixel as a mixture of Gaussians. It is usually called background Gaussian mixture model. The disadvantage of the method comes from the huge amount of memory and time needed for modeling of all pixels. Therefore this method is usually used to get an initial estimate of the background model before the start of the tracking, then during the tracking the segmentation is done based on this initial estimate and with help from other features like, for example, color histogram. This technique has been used in soccer sport by Xu [74] and Figueroa [34]. Other statistical based methods have been used by [76, 96].

2.2.2 Encoding Visual Features

The selection of suitable features to track is very important for successful tracking. The features used for tracking could be either the shape of the target or its color information. The widely used shape-based features are the active contours (snakes) which are introduced by Kass et al. [69]. Active contours are defined as energy-minimizing splines controlled (guided) by external constraint forces and influenced by image forces that attract it to lines and edges features. The main disadvantage of these techniques when used to track people is that the resulting contour may not represent the tracked target sufficiently enough because of missed features or noisy images. Further more it is poor for the handling of occlusions. Due to the drawbacks of this technique it has been improved by using color information of the target objects as in [103]. The technique is also costly in terms of computations. To speed up the computation Olszewska et. al.[46] has used a parametric active contour method based on B-spline and gradient vector flow formalisms [116]. More review on using contour-based methods can be found in [76]. In his work Needham [76] encoded the shapes of the indoor soccer players using a set of five pre-learned multiresolution kernels. A similar approach was adopted by Lu and Little [66], who used a pre-learned set of dense grids of histograms of oriented gradients [27] to track hockey players and recognize their actions. The early approaches to use color features was done by [40, 97]. Color templates that have been extracted from the estimated position of the player has been used to find the player in the next frame. Perš and Kovačič [84] tried to use both color and shape features in order to make the representation more robust. They encoded the player's shape by utilizing 14 binary Walsh-function-like kernels, while the player's color was encoded using his average color. Okuma et al. [51] has tried to develop a generic color-based detector for hockey players. A large number of manually extracted raw color areas containing the players has has been used to train a cascade of weak classifiers. Ok et al. [36] described the player by dividing the area where the player exists into two regions. One region conatins the shirt and the other contains the short. The mean values of the colors in the two regions have been used to represent the player. The class of visual representations that can be viewed as a generalization of this approach are the color histograms [98]. They have been successfully applied in many tracking applications in sports domain [51, 49, 85] as well as in the more general applications of visual tracking [80, 77, 105, 25]. Birchfield and Rangarajan [10] proposed a class of color histograms that also integrates the spatial information of the target's color. Measurements based on visual data are known to be inherently ambiguous. Therefore, some researchers [44, 53, 74, 97] enforce a spatial continuity of the players' positions on the court by using the Kalman filter [52]. However, the assumptions that the Kalman filter imposes on the measurement process and the target's dynamics are usually too unrealistic for visual tracking and so result in a degraded performance. For this reason, many authors [117, 51, 36, 49, 76] employ particle filters [73] instead. While these methods result in a more robust tracking than Kalman filter, they usually increase the computational complexity [53].

2.2.3 Multi Target Tracking

To correctly track more than one target the identities of the tracked objects should be maintained during the whole tracking process. The techniques to manage multiple targets can be classified, based on the excellent review in [58] into the following:

1. The detect-then-associate technique which, as the name suggests, perform first a detection step followed by an target-to-measurement association step. The problem with this class of technique is that the association step may require complex computation.

2. Concatenate all the states into a single joint state. This will enable the use of particle filtering techniques developed for single-target tracking [76, 80]. The disadvantage of this approach as stated in [58] is that the poor estimation of a single target may degrade the entire estimation and if the number of particles is increased to overcome this problem the computational cost will increase as well [54].

3. Track each target using separate trackers when the number of targets is already known. This will reduce the computational complexity with respect to the previous techniques but a confusion between near by targets may cause errors in tracking which require additional effort to overcome this difficulty. Some solutions such as histogram back-projection [97] and occlusion alarm probability [36] have been presented. However when the targets have similar visual features these techniques still fail when the targets are very near to each other.

2.2.4 Evaluation of Tracking Systems

In order to use the results from a tracking system to make further analysis or research it is important to evaluate the system to prove the correctness of the obtained tracking. Perš in his study to error and mistakes in automated player tracking [48] tried to investigate different error sources in the tracking process. Some patterns of players' movement have been designed and tracked using APAS manual tracking software [30] and used to find the best tracking algorithm parameters. In [60] Kristan et. al. used 273 manually tracked frames with frame rate 25 pfs for handball players moving in a specific pattern in order to evaluate his tracking technique. The tests done by these authors were not enough in terms of the length of the test video sequences and that they do not represent most of the playing situations.

2.3 Commercial Sport Video Analysis Systems

Because it is the most popular sport most of the commercial systems are dedicated for the analysis of soccer. Needham [76] has reviewed some commercial software systems in his PhD thesis which include the following:

- **SoftSport inc.** concentrates on passes of the ball in soccer sport. Two products are offered. The first is “Second Look 3P's (Player, Performance, Profile)” which provides a detailed player performance report during the whole game, including the completed and lost passes. The second is “Second Look” which analyzes the full ninety minutes of soccer match and provides many detailed reports on team performance and all individual player performances. It provides an overview of the game tactics and strategy. This system requires entering all the data to the software [39].

- **Match Analysis** allows a coach to send off a video of the game, and their specialists will methodically enter many events into the computer as they happen. Within a couple of days a glossy set of “Player Reports”, “Team Reports” and a “Coaches Summary” are sent out featuring events like, shots on-goal, losses on Pass vs Dribble and more [3].

- **ProZone** is a professional company which fits several cameras in the soccer ground and overnight manually marks the positions of each player on video footage. This provides information of the ground plane position of each player throughout the game [87].

- **TrackSYS** offers a software package “’The Observer Video-Pro’ (developed by Noldus Information Technology) used for video annotation and presentation. A set of events and players are defined before the games, and the game is annotated as it occurs, or afterwards from video. This data can be analysed to show the basic statistics of the game, as a time-event plot, or a series of video clips can be produced to display the results[102].

2.4 Discussion

In this chapter an overview on the sport player tracking using computer vision has been presented. The first part of this review is dedicated to presenting the player tracking in sports domain as a computer vision problem. The second part concerns the state of the art in player tracking based on computer vision.

The presented review is based on the excellent review of Kristan in his PhD [58]. The methods used in background detection and feature encoding subsections depends mainly on the quality of the images acquired from the camera. Although the work of Kristan [60, 58] and Perš [84, 47] tried to find a solution for tracking of indoor sports players, the acquisition of the video data is based on low resolution analog cameras of resolution 384×288 where the player is represented by 10×10 pixels. The low quality of the resolution of the video is one source of errors in the tracking process as explained in [48]. The noise that comes from digitalization of analog images and the artifacts resulted from video compression [48] require sophisticated techniques for background detection and cause poor features for the tracked target. The tests done to prove the quality of the systems were not enough because the test sequences used were not long enough and do not contain all the player interactions in real games. It has been noted from the literature review that there is no benchmark dataset from real game situations that could be used to evaluate and compare the different tracking systems.

Chapter 3

Deterministic Tracking

Deterministic tracking depends on a brute-force search mechanism where all possible positions of the target are tested. An iterative search is run in order to find the local maxima of a similarity cost function between a template image and a certain (searched) image [118]. Model based tracking algorithms incorporate a priori information about the objects to develop representations such as skin complexion, body blobs, kinematic skeleton, silhouettes or layer information [119, 17, 14, 100, 94, 22]. Appearance-based approaches apply recognition algorithms to learn the objects either in some basis such as the eigenspaces formed from observations or in kernel space [12, 7, 78].

In this chapter the deterministic methods for tracking will be discussed. It will be shown which methods are the most suitable for the tracking of sport players. The chapter is organized as follows. In Section 3.1 a simple deterministic tracking based on blob-detection is presented. Section 3.2 shows the proposed deterministic tracking method based on template matching tracking of player head.

3.1 Blob-based Tracking

Blob tracking in simple words is the tracking of the foreground objects (moving objects) in the scene based on geometric features. In the case of human tracking the simplest feature is the contour which contains the human. The

Algorithm 3.1 Blob-based tracking.

1. `specify the range of width` $\{w_{min}, w_{max}\}$ `and the range of height` $\{h_{min}, h_{max}\}$ `for the tracked object`
2. `for each frame` I_i `do`
   (a) `find the foreground mask` M_i `using Algorithm 7.1`
   (b) `find the edges in` M_i `using Canny Edge detection and the result is an edge image` C_i
   (c) `find the set of contours (blobs)` b_i `in` C_i `that match the ranges` $\{w_{min}, w_{max}\}$ `and` $\{h_{min}, h_{max}\}$
   (d) `perform tracking by associating each contour in` b_i `with the nearest path using algorithm 3.2`

detection of the contours is done based on the Background/Foreground segmentation result. The output of the detection step is a mask (called *foreground mask*) which marks the positions of the moving objects in the scene. After some filtering operations the mask is used as an input to an edge detection algorithm which in turn is used as input to the contour detection algorithm. Based on the size of the contours the blobs are classified to determine if they contain a player or not. After the contour detection, each new position should be associated with one of the paths found so far. The details of the algorithm can be seen in Algorithm 3.1.

The simplest method to associate each found contour in the current frame to another one in the previous frame is the nearest neighbor search which is explained in Algorithm 3.2. The algorithm is based on computing the distance matrix between the positions of the detected blobs and the last positions added to the paths. The smallest element in the matrix is found and its pair (blob and path position) are associated. The process is repeated after deleting the row and column of the minimum distance from the matrix until all blobs are associated. Figure 3.2 shows an example to the association of the current found blobs to the previous paths. The distance matrix is computed based on euclidean distance metric. Because the number of moving objects in the scene is limited the algorithm runs in linear time.

The Background/Foreground estimation is based on finding a model for

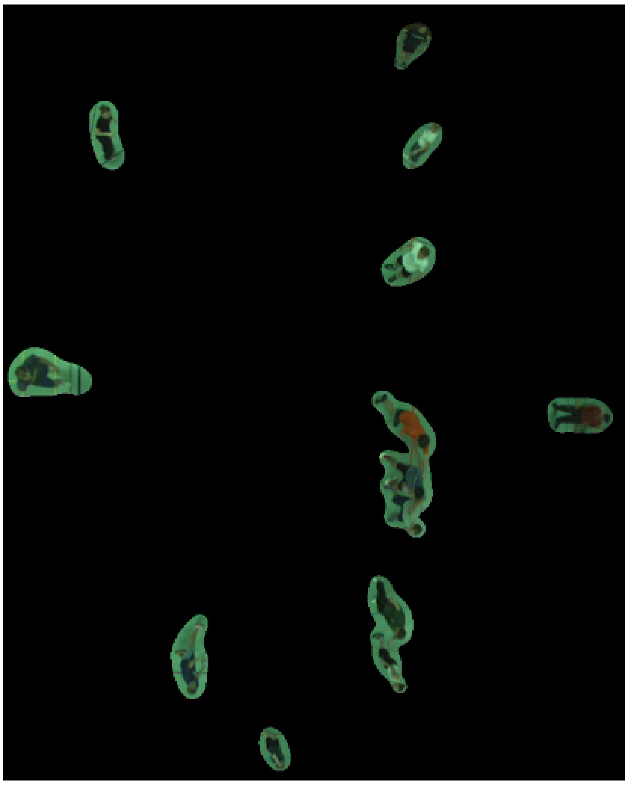


Figure 3.1: Blobs of the players in handball field.

the background and then using it to detect the moving objects by subtracting from each frame. This is explained in Algorithm 7.1. The drawback of this simple technique is that there is no strategy for handling *collisions* which occur when two or more players come very near to each other. Figure 3.1 shows the detected blobs from a handball training session. It can be seen that some blobs contain one player but others contain two or more players.

3.2 Template Matching-based Tracking

Templates are formed using simple geometric shapes or silhouettes [120]. It can also be sub-images which contain the objects of interest. Template Matching is one of the techniques used in the pattern recognition domain which is used to find parts of an image that match a template (reference) image. The template image is compared to all parts of the searched image

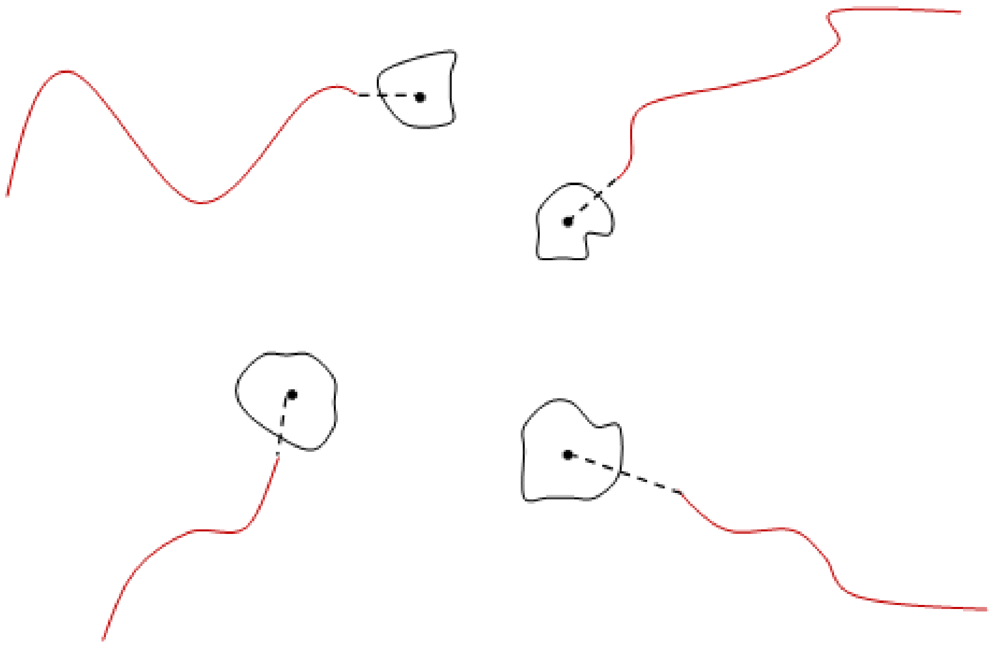


Figure 3.2: Association of blobs with paths.

Algorithm 3.2 Blob association algorithm.

1. `do until all blobs` b `are associated`
   (a) `compute the distance matrix` D `for the distances between all blobs` b `and the paths` p
   (b) `find the smallest element` d_{ij} `in the distance matrix` M
   (c) `associate blob` b_i `with path` p_j
   (d) `delete raw` i `and column` j `from` D `(by replacing with a large value)`

and a measure of similarity is computed in each comparison step. The result of the template matching procedure is a matrix in which the intensity of each pixel indicates the degree of similarity with the template. In order to find the position of the best match a maximum value search is applied on the result image.

An advantage of a template is that it carries both spatial and appearance information. Templates, however, only encode the object appearance generated from a single view. Thus, they are only suitable for tracking objects whose poses do not vary considerably during the course of tracking.

The basic algorithm of template matching uses a convolution mask (template), taken from a specific feature of the searched image. This technique can be easily performed on gray images or edge images. It is intuitively likely that the convolution output will be highest at places where the image structure (feature) matches the mask structure, where large image values get multiplied by large mask values. This method is normally implemented by firstly picking out a part of the search image to use as a template: The searched image is represented as $I(x, y)$, where (x, y) are the coordinates of each pixel in the search image. The template is represented as $T(x, y)$, where (x, y) is the coordinate of each pixel in the template. The template is centered over each (x, y) point in the searched image. A *matching function* between I and T is calculated over the whole area spanned by the template. As all possible positions of the template with respect to the search image are considered, the position with the highest matching value will most probably contain the searched feature.

Figure 3.3 shows an example of template matching. The left image is the searched image. The template is taken to be the red-bounded sub image from the searched image (shown in the middle in bigger size). The result of the template matching procedure on the searched image is shown on the right image where the high intensity pixels shows better matching than the less intensity ones.

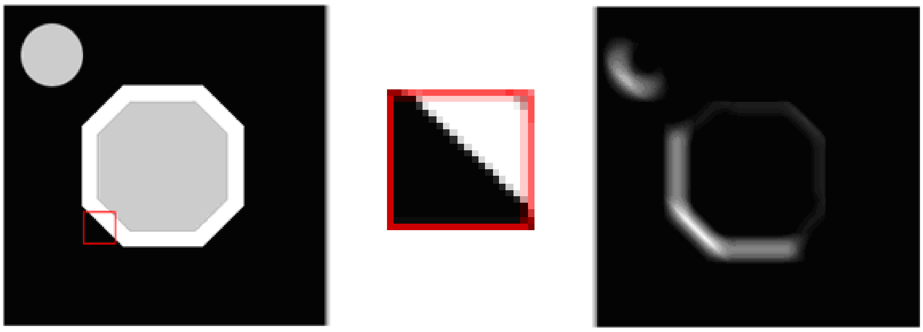


Figure 3.3: Example of template matching.

3.2.1 Matching Functions

The matching function (also called cost function) usually used is the Sum of Squared Differences (SDD) between the template and the current image as in [67, 37, 9]. More robust similarity measures have been applied and the mean-shift algorithm or other optimization techniques have been utilized to find the optimal solution [11, 25, 31, 24].

Other matching functions that are mainly based on statistical correlation try to find how much a dataset is similar to another one. If the size of the searched image I is $W \times H$ and the size of the template image T is $w \times h$ the result of the matching is a matrix $R(x, y)$ of size $(W - w + 1) \times (H - h + 1)$. The following is a list of matching functions that may be used with template matching:

1. Sum of Squared Differences: This is the simplest and fastest method where only the sum of the square differences (squared eculidean distances) between each pixel in the template and the corresponding pixel in the searched image is calculated as in equation (3.1) [15].

$$R(x, y) = \sum_{x',y'} [T(x', y') - I(x + x', y + y')]^2 \tag{3.1}$$

The result in equation (3.1) can be normalized with a normalization factor N_m.

$$R(x,y) = \frac{\sum_{x',y'} T(x',y') - I(x+x',y+y')]^2}{N_m} \tag{3.2}$$

where N_m is defined as follows:

$$N_m = \sqrt{\sum_{x',y'} T(x',y')^2 . \sum_{x',y'} I(x+x',y+y')^2} \tag{3.3}$$

2. Cross Correlation: By expanding equation (3.1) it results in

$$R(x,y) = \sum_{x',y'} [T(x',y')^2 + I(x+x',y+y')^2 - 2T(x',y')I(x+x',y+y')]$$

the term $T(x',y')^2$ is constant. If the term $I(x+x',y+y')^2$ is approximately constant then the remaining term is the cross correlation term.

$$R(x,y) = \sum_{x,y} [T(x',y')I(x+x',y+y')] \tag{3.4}$$

and the normalized form is

$$R(x,y) = \frac{\sum_{x,y} [T(x',y')I(x+x',y+y')]}{N_m} \tag{3.5}$$

3. Correlation Coefficient: The correlation coefficient between the template and the searched image is calculated as known from statistics as following

$$R(x,y) = \sum_{x,y} [T'(x',y')I'(x+x',y+y')] \tag{3.6}$$

Where $T'(x',y')$ and $I'(x+x',y+y')$ are the means of the template and searched image respectively and can be computed as following

$$T'(x',y') = T(x',y') - \frac{1}{wh} \sum_{x'',y''} T(x'',y'') \tag{3.7}$$

$$I'(x+x',y+y') = I(x+x',y+y') - \frac{1}{wh} \sum_{x'',y''} I(x+x'',y+y'') \tag{3.8}$$

where w and h are the width and height of the template image respectively. The correlation coefficient can also be normalized as described in equation 3.9

$$R(x,y) = \frac{\sum_{x',y'} T'(x',y').I'(x+x',y+y')}{\sqrt{\sum_{x',y'} T'(x',y')^2 . \sum_{x',y'} I'(x+x',y+y')^2}} \tag{3.9}$$

Using Equation 3.4 to compute the cross correlation has several disadvantages as described in [63]:

- if the image energy $I(x+x',y+y')^2$ varies with position, matching using equation 3.4 can fail. For example, the correlation between the template and an exactly matching region in the image may be less than the correlation between template and a bright spot.
- the range of the cross correlation $R(x,y)$ depends on the size of the template.
- equation 3.4 is not invariant to changes in image amplitudes such as those caused by changing lighting conditions across a sequence of images.

These difficulties are overcome by normalizing the image and template vectors to unit length. The template matching algorithm is usually applied to gray scale images. For color images the same procedure can be applied for each color channel and the result of matching each channel should be combined into one result. The flow chart in Figure 3.4 shows the use of template matching in tracking. Before the start of the tracking loop the background model is calculated using Algorithm 7.1. The initial position of the tracker is determined before the start of the tracking process. For each frame the mask of the foreground is calculated using subtraction from the background. The current frame is multiplied by the background mask to find the regions of interests in the image (ROIs). In the sport domain the background is constant (static) so it can be calculated easily and subtracting it from each frame helps the matching step to focus more on the foreground objects being tracked. The matching is done using equation 3.9 and the best matching position is found. Finally, the template is updated and then the whole process is repeated for the next frame.

There are still some pitfalls in the tracking algorithm presented in the flow chart which concerns the template update, size and the number of templates. In the following subsections these issues will be discussed.

3.2.2 Selection of Template

In order to use template matching for tracking some points should be taken into consideration.

- Looking at all possible orientations of the sport player during the game and setting template for all possible situations will require a large number of templates which is not practical.
- Due to the highly dynamic and changing nature of sport players' movements, the use of a static template is not optimal. So the template should be updated or there should be a number of templates used for tracking.
- The size of the template it is difficult to set a fixed size of the player template because according to the camera setup described in Chapter 7 the player shape will be changed according to his distance from the camera because of fisheye objectives. This can be seen in Figure 3.5 which shows a number of players in different situations and positions as seen by the camera. In this figure, where there is no scaling of the player shapes, it is clear that the player body size and orientation is changing. Figure 3.6 shows a screen shot from one a basketball game taken from one fisheye camera used in the experiments of this thesis work.

The proposed solution to handle the problem of different sizes and orientations of the players is to track the upper part of the player which is the head and part of the shoulders. As can be seen in Figures 3.5 and 3.6 the

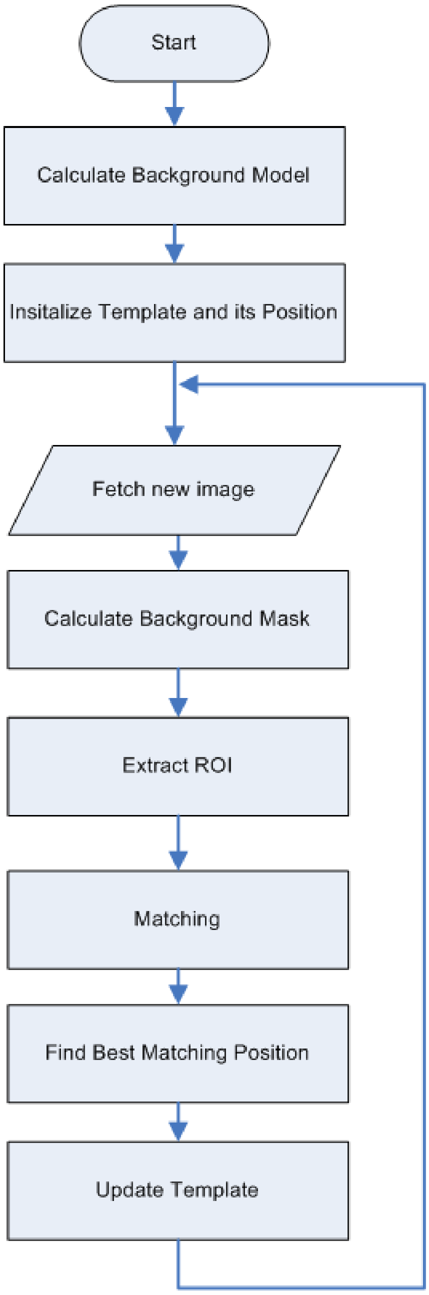


Figure 3.4: Flowchart of template matching tracking.

Figure 3.5: Player shapes in different positions on the playing field taken by overhead fisheye-lens camera.

Figure 3.6: A screen shot of basketball game taken by fisheye lens camera.

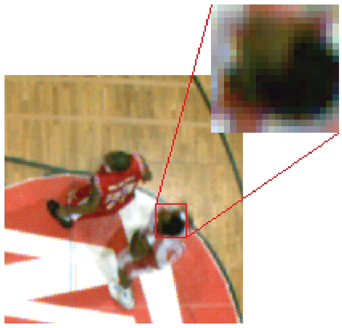


Figure 3.7: Example of player's head template.

heads of the players are visible in all orientations and its circular shape is not changing much as the whole body shape. So template matching tracking of the head plus a small part of the shoulder can be easier than tracking the whole body. Tracking the player head makes it easier to calculate the feet position as described in Section 7.1.2. The size of the template depends on the image resolution and should be determined before the start of the tracking process.

Figure 3.7 shows how a head template of size 20×20 of a basketball player's head looks like. The use of fixed size templates of the head may solve the problem of template size but the update of the template and the handling of multiple players is still unsolved. In Chapter 5 the whole issue of tracking multiple players is discussed in detail, the rest of this chapter will be dedicated to the updating of the template and the possible enhancements to the tracking algorithm.

Figure 3.8 shows the result of the template matching process. Images on the left are the searched images which are marked with red arrows to show the player under consideration. The images in the second column are the used templates. The third column is the result of the template matching where the best matching position is highlighted. The fourth column shows the amount of best matches.

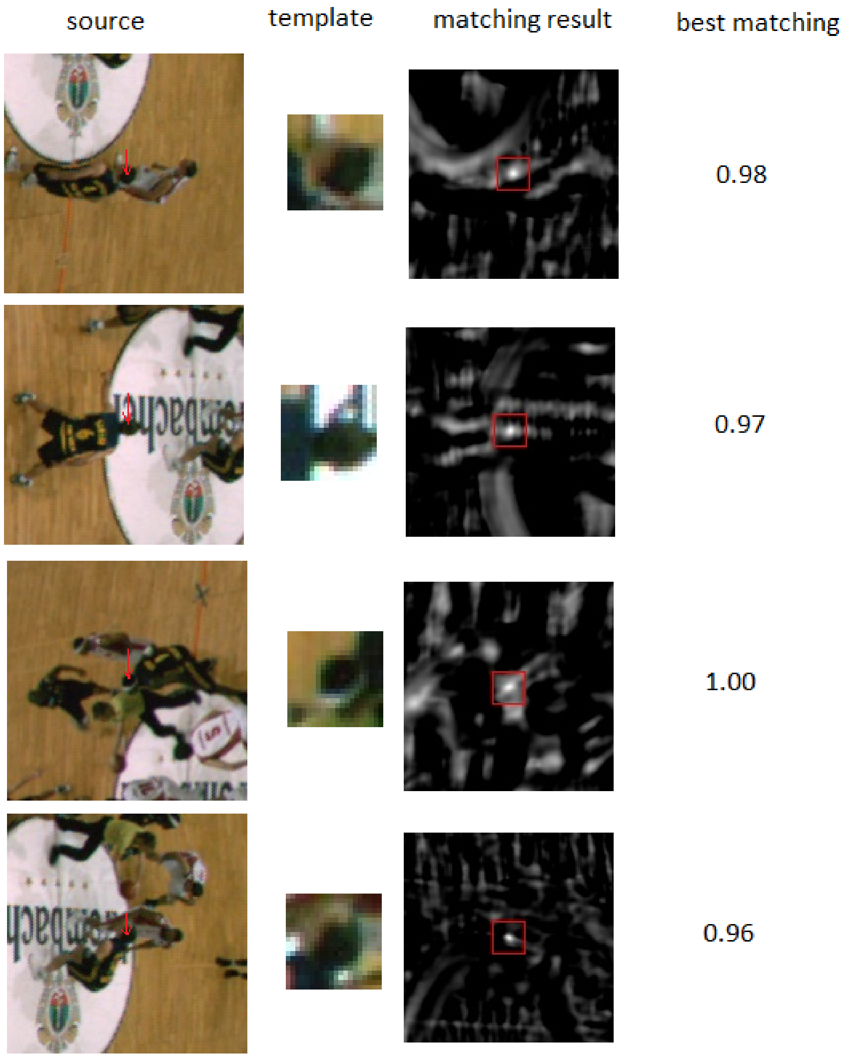


Figure 3.8: Example of the result of template matching with different players.

3.2.3 Head Template Update

There are two proposed methods in order to deal with the update of the head template problem. The first option is to use one template and update it according to the rule in Equation 3.10

$$T_t(x,y) = (1-\alpha)\hat{I}_{t-1}(x,y) + \alpha T_{t-1}(x,y) \tag{3.10}$$

$T_t(x,y)$ is the template at time step (frame) t and it is updated by taking a part from the old template $T_{t-1}(x,y)$ and a part from the sub-image which resulted in the best match $\hat{I}_{t-1}(x,y)$. The second option is to save a set of templates that will represent all the possible templates for the head. In this case the current template to be used for matching is computed as described in Equation 3.11

$$T_t(x,y) = \sum_i [w_t^i T_t^i(x,y)] + (1-\alpha)\hat{I}_{t-1}(x,y) \tag{3.11}$$

The current template is computed as a weighted sum of all templates plus a part of the best matched sub-image and $\alpha = \sum_i w^i$.

3.2.4 Enhanced Template Matching Algorithm

As shown above, template matching is used to find a template image in another image based on similarity function. In the tracking of human beings and more specifically in the sport domain, some logical rules which comes from physics or logic can be used to enhance the tracking to make it more efficient. These rules are described in details in Chapter 5 under the name Closed World Assumptions. One rule to be considered is the assumption that the first template is selected manually before the start of the tracking process and that the player cannot suddenly disappear from the view of the camera or change its position completely arbitrarily. Using these rules makes it easier to focus the search in some ROIs (Region Of Interests) which reduces the processing time.

According to physics laws, players cannot change their speed arbitrarily due to the inertia [1]. So if the speed of the player is taken into consideration when the player template is searched in the next frame the matching could be led to an area where the player is more likely to appear.

In order to use the speed of the player to predict the possible position in the next frame a model of the motion of the players should be used. In [59] It has been shown, based on [89], that two models of motion can be used to model players' movement, namely the RW (Random Walk) and the NCV (Nearly Constant Velocity) dynamic models. The two models describe the player's movements in different situations. The RW model will best describe the movement if it has more abrupt radical accelerations in different directions while the NCV model describes better the kind of motion when the player moves constantly in different directions. Taking into consideration also the player's inertia which will prohibit sharp changes in his velocity and direction a simple prediction scheme can be used.

The method used in [59] is based on a probabilistic tracking scheme for the sport players. It can be adapted for the deterministic tracking as follows: Assuming the current position of the player is p_t and the position in the next frame p_{t+1} is

$$p_{t+1} = p_t + \delta_t \tag{3.12}$$

where δ_t is the prediction of the change of distance. According to [59] who refers to [16, 33], δ_t can be calculated as a weighted sum of successive past differences between the last N_c positions.

$$\delta_{t+1} = \frac{1}{C} \sum_{j=t-N_c+1}^{t} (p_t - p_{t-1}) W_k(t), \tag{3.13}$$

where the weights are defined as:

$$W_k(t) = w_k w_{k-1} e^{-\frac{1}{2}\frac{(k-t)^2}{\sigma_0^2}} \tag{3.14}$$

and $C = \sum_{k=t-N_c+1}^{t} W_k(t)$ is a normalization constant. The first two terms in Equation 3.14 are the weights of the two positions under consid-

[1] The resistance of any physical object to a change in its state of motion or rest

eration. In the case of [59] the weight of the object state taken from the likelihood function of the state. In the case of the template matching this weight could be either set to 1 or to the amount of matching resulting from the matching function can be used.

The last term in Equation 3.14 is a Gaussian that assigns higher weights to more recent positions. The number of past positions N_p in [59] is chosen based on the assumption that the player is not able to change his speed sharply within half a second. This means to talk the past $f/2$ positions, where f is the frame rate. The parameter σ is set to $N_c/3$ based also on [59].

3.3 Tracking Algorithm

Algorithm 3.3 summarizes the single player template matching-based tracking technique described in this chapter. The algorithm does not take the tracking of multiple players into account. The details of the tracking multiple players will be presented in Chapter 5 and the experimental results of the template matching tracking will be presented in Chapter 7.

The output of the tracking should be coordinates of the foot in meters which are also referred to as real world coordinate transformation. When the player crosses one half of the playing field to the other half, the processing should correspondingly be transfered to the other half of the image. This require a kind of fusion between the two camera views. Both the real coordinate transformation and the fusion between camera views are described in Chapter 7.

3.4 Discussion

In this chapter it has been shown how deterministic tracking can be used to track players using overhead cameras. As a start the simple blob tracking has been used. Tracking single player gives excellent results when the player is not interacting with others. When collision between players occurs two or more players come to one blob and after separation it is difficult to associate

Algorithm 3.3 template matching tracking algorithm

```
compute the background using Algorithm 7.1
get first frame and initialize the template manually
For each input frame
```

1. `Pre-process the frame as described in 7.1.1`
2. `Subtract the frame from background model to find the foreground mask`
3. `Compute ROI for template matching using speed computed from Equation 3.13`
4. `Perform template matching`
5. `Transform the output coordinates of template matching to real coordinates using Equation 7.10`
6. `If player is on middle line transfer the processing to other half of the frame as described in 7.1.2`

the paths again. The time complexity of the blob detection depends on the implementation of the edge detection. Canny edge detection [19] has been used in the implementation. After the edge detection the connected components in the "edges image" are inspected to find contours which is the classified based on the player size on the image. Speaking about the space complexity or the amount of memory required for blob detection, several temporary images and data structures are needed to store the result of the edge detection and contour detection.

The second part of the this chapter shows how template matching can be used for tracking sport players. Investing the high quality of the images it is possible to track the head of the player with small part of the shoulders. In order to determine the best size of the template, experiments have been done in Chapter 7. Tracking the head has the advantage that no rotation or scaling of the head is required. Although the overhead cameras reduces the chance that the head of the player is totally invisible due to occlusion, in some situations when the player falls down or other taller player comes into a strong contact with him the head disappears from the scene. The multiple player tracking framework described in Chapter 5 reduces the number of errors that occurs due to occlusion. In some cases the tracker still needs to be manually corrected. The rate of error occurrence is inspected in Chapter 7.

Chapter 4

Probabilistic Tracking

As shown in the previous chapter, deterministic tracking depends on a brute-force search mechanism. Although some information about the movement and the nature of the target can be used to guide the search there are still uncertainties which need to be dealt with. In the sports domain, the players try to move in a fast and unpredictable way in order to confuse or escape from the opponent. In order to deal with such uncertainties, a probabilistic frame work has been used by Russel [91] to perform tracking. Early approaches to deal with uncertainties are based on Kalman Filtering [52] which relies on the assumption that the involved distribution is Gaussian.

In order to deal with non-linearity and non-Gaussianity, approaches that make use of Bayesian filters and Monte Carlo Simulation methods have been developed by Mackay [71]. Gordon [35] introduced a re-sampling phase in the sequential Bayesian filter algorithms. These techniques are used in the Artificial Intelligence domain under the name "survival of the fittest" and in the control field as *particle filtering*. In the CV domain these methods were first used by Isard and Blake [43, 41] under the name *CONDENSATION*.

Reviews that show how widely these techniques have been used in recent years have been done by Doucet [29] and Arulampalam et al. Other work can be found in [17, 28, 42, 45, 68, 75, 104, 70, 88, 80, 50]. Further, comprehensive treatments are given in [2, 8]. However, several drawbacks remain as stated by King and Forsyth [56]. Despite the great number of improvements that have already been introduced, many open issues prevent stating that particle filters are able to solve unconstrained tracking problems.

This chapter shows the use of a probabilistic framework based on particle filters for tracking sport players. The methods presented in this chapter are based on the work of Isard and Blake [43, 41] concerning the use of particle filter in CV and the work of Kristan [60, 58], Perš [84, 47] and Rowe [90] who applied particle filters in the tracking of humans. Section 4.1 gives an introduction to particle filter in general. Section 4.2 shows how can the particle filter be used for the tracking sport players in terms of modeling both the player appearance and his motion and the likelihood function based on color feature. Section 4.3 presents the single player particle filter-based tracking algorithm. Finally, a discussion of the chapter contents is presented in Section 4.4.

4.1 Particle Filter

From a probabilistic point of view, the tracking problem involves dealing with stochastic processes. These are series of time-slices describing the state of all the entities within the scene [90]. Each time-slice consists of a set of random variables. Two kinds of variables can be distinguished, namely unobservable state variables at time t, denoted as $\mathbf{S}_t$, and observable (measurable) evidence variables, denoted as $\mathbf{Z}_t$. The interval between time-slices depends on the frame rate. Given a state model $\mathbf{s}_{t-1}$ at time $(t-1)$ and all the measurements up to $(t-1)$ $\mathbf{z}_{0:t-1}$, the posterior $p(\mathbf{s}_{t-1}|\mathbf{z}_{0:t-1})$ can be estimated by the recursion of Equations 4.1 and 4.2 using new measurement $\mathbf{z}_t$

$$prediction : p(\mathbf{s}_t|\mathbf{z}_{0:t-1}) = \int p(\mathbf{s}_t|\mathbf{s}_{t-1})p(\mathbf{s}_{t-1}|\mathbf{z}_{0:t-1})d\mathbf{s}_{t-1} \tag{4.1}$$

$$update : p(\mathbf{s}_t|\mathbf{z}_{0:t}) = \frac{p(\mathbf{z}_t|\mathbf{s}_t)p(\mathbf{s}_t|\mathbf{z_{0:t-1}})}{\int p(\mathbf{z}_t|\mathbf{s}_t)p(\mathbf{s}_t|\mathbf{z}_{0:t-1})d\mathbf{s}_t} \tag{4.2}$$

The recursion for the posterior thus requires a specification of a dynamical model describing the state evolution $p(\mathbf{s}_t|\mathbf{s}_{t-1})$, and a model that evaluates the likelihood of any state given the observation $p(\mathbf{z}_t|\mathbf{s}_t)$.

The Conditional Density Propagation (CONDENSATION) algorithm is used in the implementation [41]. CONDENSATION intended to track a human contour which moves in cluttered background, given a video signal as data. CONDENSATION has been simply described by Kristan [57] as follows: The posterior at time-step $(t-1)$ is represented by a finite particle set of N_p particles with normalized weights $w_{t-1}^{(i)}$

$$p(\mathbf{s}_{t-1}|\mathbf{z}_{0:t-1}) \approx \{\mathbf{s}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N_p} \tag{4.3}$$

so that the sum of all weights is one. At time-step t, the particles are re-sampled according to their weights in order to obtain an unweighted representation of the posterior

$$p(\mathbf{s}_{t-1}|\mathbf{z}_{0:t-1}) \approx \{\tilde{\mathbf{s}}_{t-1}^{(i)}, \frac{1}{N_p}\}_{i=1}^{N_p} \tag{4.4}$$

and are then simulated according to the dynamical model $p(\mathbf{s}_t^{(i)}|\tilde{\mathbf{s}}_{t-1}^{(i)})$ to obtain a representation of the prediction $p(\mathbf{s}_t|\mathbf{z}_{0:t-1}) \approx \{\mathbf{s}_t^{(i)}, \frac{1}{N_p}\}_{i=1}^{N_p}$. Finally, a weight is assigned to each particle according to the likelihood function $w^{(i)} = p(\mathbf{z}_t|\mathbf{s}_t^{(i)})$, all weights are normalized, and the posterior at time-step t is approximated by a new particle set $p(\mathbf{s}_t|\mathbf{z}_{0:t}) \approx \{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N_p}$.

In order to use CONDENSATION for the tracking of sport players the following should be specified:

- model for the player (state model)
- model for the motion of the player
- model for the state transition (evolution)
- likelihood function

In the next Section 4.2 it will be shown how the three models and the likelihood function are specified and implemented.

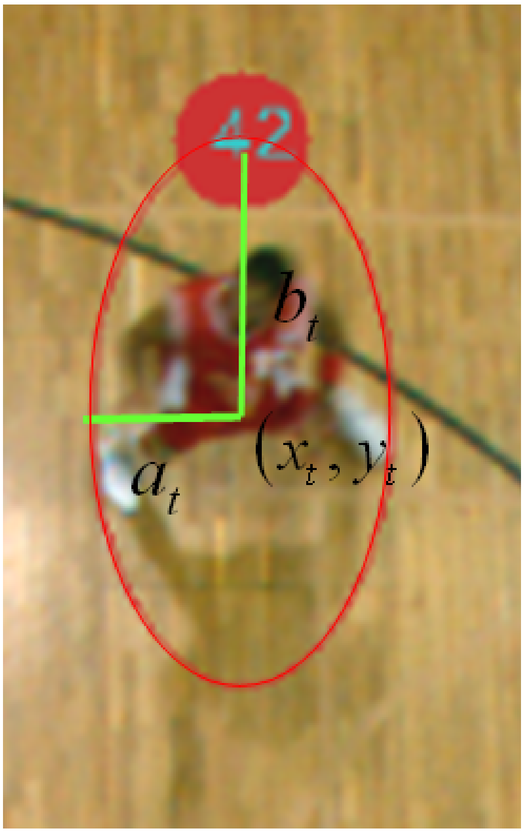


Figure 4.1: The player model.

4.2 Particle Filter-based Player Tracking

4.2.1 Modeling the Player

The target (player) motion is characterized by its position at time t, $\mathbf{x}_t = (x_t, y_t)^T$ which is here the center of the ellipse as shown in Figure 4.1. The aspect model is given by the bounding ellipse and an appearance matrix. The bounding ellipse is denoted by $\mathbf{e}_t = (a_t, b_t)^T$. The appearance matrix denoted by $\mathbf{A}_t$, stores the pixel intensity values within the bounding ellipse. A label l associates a specific appearance model to the corresponding state, allowing multiple-target tracking. Therefore, the l-target's state is defined as $\mathbf{s}_t^l = (\mathbf{x}_t^l, \mathbf{e}_t^l, \mathbf{A}_t^l)^T$. In the case of player tracking the label l can be the player t-shirt number or the player name.

4.2.2 Modeling Player Motion Dynamics

Modeling the dynamics of the tracked target is an important part of the probabilistic tracking. Well-modeled target motion dynamics can cause many improvements to the performance of the tracker. One important improvement could be the accurate estimation of the target's position which will result in a smaller number of tracking errors. Using a large number of particles will result in better estimation of the target state. On the other hand large number of particles may cause more cost on computing the likelihood probability. So another improvement that comes from a good dynamic model is to efficiently use a smaller number of particles by leading them to the regions where it is more likely to contain the current state of the target.

Several independence relationships are assumed in order to determine the transition model. It is considered that both aspect and dynamic models are independent and that the position only depends on the previous position and the speed on the previous one, and so does the bounding box and the appearance. Therefore, the transition model can be split [90]:

$$\begin{aligned} P(\mathbf{S}_t|\mathbf{S}_{t-1}) &= P(\mathbf{X}_t, \mathbf{E}_t, \mathbf{A}_t|\mathbf{X}_{t-1}, \mathbf{E}_{t-1}, \mathbf{E}_{t-1}) \\ &= P(\mathbf{X}_t|\mathbf{X}_{t-1})P(\mathbf{E}_t|\mathbf{E}_{t-1})P(\mathbf{A}_t|\mathbf{A}_{t-1}) \end{aligned} \tag{4.5}$$

Players usually aim to act in an unpredictable fashion to confuse the opponent, which implies that the dynamics should be modeled by a Brownian motion [111]. The player's motion, however, is restricted by his task and laws of physics; i.e. a player's task is to travel from region A to region B, and during that traversal the position cannot be changed instantly and arbitrarily due to the effects of the inertia. Therefore, the transition (evolution) model of the position $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ can be defined as

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = N(\mathbf{x}_t; \mathbf{x}_{t-1} + d_t, \Lambda_t) \tag{4.6}$$

Where $N(.; \mu, \Sigma)$ is a normal distribution with mean μ and covariance matrix $\sum$ and d_t is a drift constant at time t for all particles. By the same way the translation model of the bounding ellipse can be written as:

$$p(\mathbf{e}_t|\mathbf{e}_{t-1}) = N(\mathbf{e}_t; \mathbf{e}_{t-1} + d_t, \Lambda_t) \tag{4.7}$$

The drift constant d_t is calculated the same way as δ_t (see Section 3.2.4) from the past $f/2$ states.

4.2.3 Color-Based Likelihood Function

The likelihood function or model $p(\mathbf{z}_t|\mathbf{s}_t)$ gives the probability density function of the measured feature $\mathbf{z}_t$ given the current state of the target $\mathbf{s}_t$. This function is used to evaluate the particles to give them weights that are needed for the re-sampling step in the CONDENSATION algorithm. The color features in the appearance model $\mathbf{A}_t$ are used as the measurable features. Hence, the appearance $\mathbf{A}_t$ is given by a matrix whose elements are the pixels' intensity values. The target appearance is represented by color histograms.

Histograms are broadly used to represent human appearance, since they are claimed to be less sensitive than other representations, such as color templates, to rotations, camera point of view, non-rigid targets and partial occlusions [90].

A histogram is a statistical description of a random variable space that captures the occurrence frequencies of a variable in different event classes. For color, the event classes are regions in color space [106]. Color histograms represent the color distribution of images by counting the number of pixels in the range of color class pixels according to their color range class in color space.

A standard color histogram is derived by approximating the intensity distribution independently for each red (R), greed (G) and blue (B) color band through an N-bin histogram.

Standard color histograms project the color distribution of an image to three color bands. This projection removes the spatial 3D information of the RGB tuples in color space. The spatial relation of the color information is retained if the color values of the image pixels are not considered as independent but as tuples in color space. A histogram then discretizes the tuples in

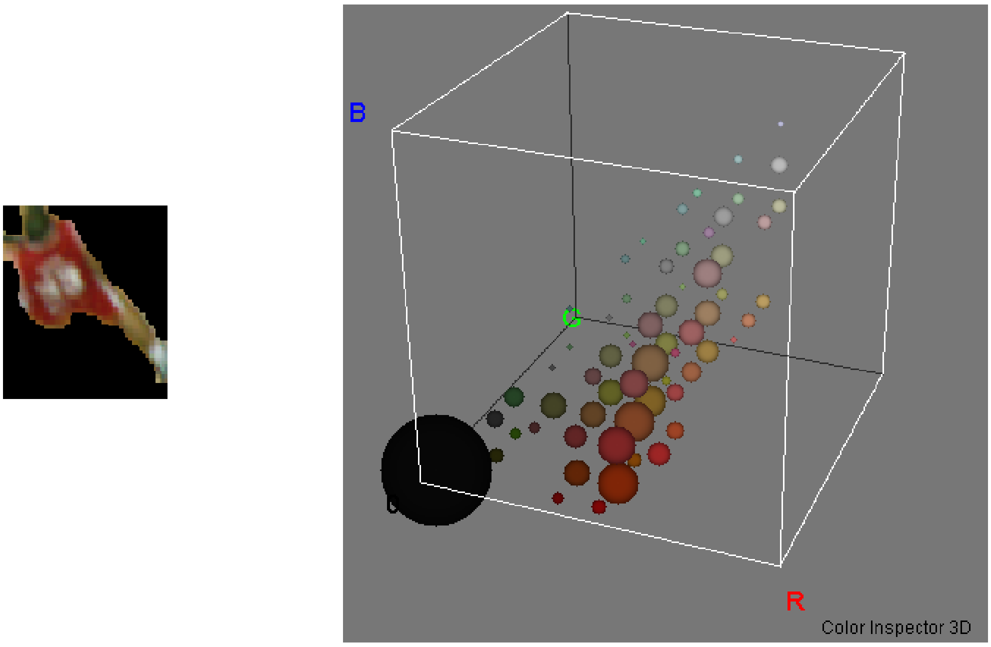


Figure 4.2: Three-dimensional color histogram for an image of a player.

N^k equally sized bins, where k denotes the dimension of the color space.

In RGB color space a 3D color histogram which consists of N^3 equally sized bins preserves the spatial information of the color pixels. 3D color histograms define an equivalence function on the set of all possible colors, that means two colors are the same if they fall into the same bin. Figure 4.2 shows a 3D color Histogram for a player image [1]. The player image is masked by the background mask, so the bin which contains the black color is big and should be ignored in the calculation of the histogram.

The histogram of the l-model is given by:

$$\bar{\mathbf{h}}^l = \{h_k^l; k = 1 : K\}, \tag{4.8}$$

where K is the number of bins, and the probability of each feature is given by:

[1]Generated by ImageJ 1.43u software, written by Wayne Rasband, National Institute of Health, USA, http://rsb.info.nih.gov/ij

$$h_k^l = C^l \sum_{a=1}^{M} M(x_b)\Pi(b(x_a))\delta(b(x_a) - k), \tag{4.9}$$

where C^l is a normalization constant required to ensure that $\sum_{k=1}^{K} h_k^l = 1$, $\Pi(.)$ is a weighting function that gives higher weights to the pixels closer to the center [26, 60], δ the Kronecker delta, $\{x_a; a = 1 : M\}$ the pixel locations and $b(x_a)$ a function that associates the given pixel to its corresponding histogram bin. $M(.)$ is a mask function that gives a zero weight to background pixels, it is computed as described in section 7.1.1. The l-labeled target's state is then defined as $\mathbf{s}_t^l = (\mathbf{x}_t^l, \mathbf{e}_t^l, \bar{\mathbf{h}}^l)^T$.

The presence of a player in a given state is evaluated by comparing some candidate histograms to a reference histogram of the tracked player. The similarity between two histograms can be computed using the following metric [90, 50]:

$$d_B = \sqrt{1 - \rho(\mathbf{h}_1, \mathbf{h}_2)}, \tag{4.10}$$

where

$$\rho(\mathbf{h}_1, \mathbf{h}_2) = \sum_{k=1}^{K} \sqrt{h_1 h_2} \tag{4.11}$$

is known as the *Bhattacharyya coefficient*. Therefore, similar histograms have a high Bhattacharyya coefficient, which should correspond to high sample weights.

Due to nonuniform lighting and change in players pose while moving across the field, the texture of the players may change. Kristan [58] suggests that the appearance model for tracking people should be updated. Thus the player reference model should be adapted. This adaption will be applied on the appearance model of the player which is the histogram. The reference histogram at time-step t is adapted based on [58] as follows:

$$\mathbf{h}_t^l = \alpha \mathbf{h}_{\mathbf{s}_t}^l + (1 - \alpha)\mathbf{h}_{\mathbf{s}_{t-1}}^l \tag{4.12}$$

where $\alpha \in [0, 1]$ is the adaptation rate.

4.3 Tracking Algorithm

After determining the required models for CONDENSATION, as shown above, it can now be used for tracking sports player. For completeness, a listing of CONDENSATION is shown in Algorithm 4.1. The details of the algorithm regarding for example the re-sampling requires explaining the whole statistical background which is out of scope of this thesis. More details about CONDENSATION and the statistical background behind it can be found in [41]. The whole process of tracking requires image pre-processing and post-processing of the output in addition to the tracking *engine*. So CONDENSATION is shown in Algorithm 4.2 as a step of the tracking algorithm.

The complexity of the CONDENSATION algorithm depends on two main parameters, namely the number of particles and the computation of the likelihood function for evaluating each particle by giving a weight. The likelihood function depends on computing a three-dimensional histogram for each particle and measuring the distance between it and the reference histogram. The complexity of computing the histogram for an image (or sub-image) depends on the image size and the number of channels as well as the number of histogram bins. The more bins each histogram for each channel has, the better the representation of the color distribution and the longer the time needed for computations is. In Chapter 7 experiments have been done in order to find the best number of bins for computing the histogram and the best number of particle for the CONDENSATION.

Algorithm 4.2 shows the processing steps of tracking one player based on CONDENSATION as the tracking kernel. The total time needed for the tracking of one player includes the pre-processing described in Section 7.1.1, the tracking kernel and the post-processing of the tracking data. In order to track multiple players using CONDENSATION the multiple player tracking framework described in Chapter 5 should be used.

Algorithm 4.1 CONDENSATION algorithm adapted from [41].

1. From the "old" sample-set $\{\mathbf{s}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N}$ at time step $t - 1$, construct a "new" sample-set $\{\mathbf{s}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N}$
2. construct the n^{th} of N new samples as follows:
   (a) Select a sample $\mathbf{s'}_t$ as follows:
      i. generate a random number $r \in [0, 1]$, uniformly distributed.
      ii. find, by binary subdivision, the smallest j for which $c_{t-1}^{(j)} \geq r$
      iii. set $\mathbf{s'}_t^{(n)} = \mathbf{s}_{t-1}^{(j)}$
   (b) Predict by sampling from $p(\mathbf{s}|\mathbf{s}_{t-1} = \mathbf{s'}_t^{n})$ to choose each $\mathbf{s}_t^{(n)}$
   (c) Measure and weight the new position in terms of the measures feature $\mathbf{z}_t$: $w_t^i = p(\mathbf{z}_t|\mathbf{s}_t = \mathbf{s})$ then normalize so that $\sum {w_t^{(n)}}_n = 1$ and store together with cumulative probability as $\mathbf{s}_t^{(n)}, w_t^{(n)}, c_t^{(n)}$ where

$$\begin{aligned} c_t^{(0)} &= 0, \\ c_t^{(n)} &= c_t^{(n-1)} + w_n^i (n = 1, ..., N). \end{aligned}$$

3. once the N samples have been constructed: estimate, the moments of the tracked position at time step t as a weighted mean $\mathbf{s}_t$ obtaining the mean position using $\sum_{n=1}^{N} w_t^{(i)} \mathbf{s}_t^{(i)}$

4.4 Discussion

This chapter concerns the tracking of a single sports player using probabilistic model in order to deal with the non-linearity and the uncertainty in the sports domain. CONDENSATION algorithm which is based on particle filter techniques has been used in the CV domain for tracking. In order to use CONDENSATION for tracking in sports domain a model of a player (state), his motion, state transition and likelihood function should be found. The state of the player has been modeled as an ellipse and the visual features

Algorithm 4.2 Single player tracking algorithm using CONDENSATION.

```
compute the background using Algorithm 7.1
get first frame and initialize the state of the tracker manually
For each input frame
```

1. `Pre-process the frame as described in 7.1.1`
2. `Subtract the frame from background model to find foreground mask`
3. `Compute ROI for the tracker from previous frame`
4. `Run the CONDENSATION described in Algorithm 4.1 on the sub image determined by the ROI`
5. `Transform the output coordinates of template matching to real coordinates using Equation 7.10`
6. `If player is on middle line transfer the processing to other half of the frame as described in 7.1.2`

inside the ellipse are modeled as a three-dimensional color histogram. Three dimensional histograms have been chosen to preserve the spatial information of the RGB color tuples in the color space. The transition model of the player motion and the ellipse size is modeled as a normal distribution. The likelihood is modeled as the Bhattacharyya distance between the state histogram and the reference histogram.

In order for the tracking algorithm presented in this chapter to be used in practice some parameters including the number of particles and the number of histogram bins need to be determined. Chapter 7 shows the experiments done to find the best parameters as well as the experiments done to evaluate the tracking itself in terms of accuracy and running time.

Chapter 5

Managing Multiple Players

In Chapters 3 and 4 the tracking of single players using deterministic template matching technique and probabilistic based technique has been introduced. In the previous work review in Chapter 2 the multiple target tracking problem can be tackled using several strategies in which each strategy has drawbacks with respect to the computational efficiency or accuracy of tracking. Using information about the *environment* of the tracking (who will be tracked, where the tracking takes place and so on) can be very helpful for the tracking algorithms and for managing multiple players. The idea of using meta-data to help accomplish a CV task has been introduced to the CV literature by Intille and Bobick [40] in 1994 under the name *closed-world tracking*. Perš in [84] and Kristan in [60] have used the same concept to track sports squash and handball players using low resolution analog cameras.

In this chapter a general framework for using closed-worlds for tracking of multiple sports players is presented. This framework gives enhancements to the way the closed worlds are used in sports tracking domain. The enhancements include not only the tracking itself but also optimization of the image pre-processing. The rest of this chapter is organized as follows: Section 5.1 gives the definition of the closed worlds concept and its application in tracking of multiple targets. Section 5.2 presents new closed world assumptions that can be used for enhancing the multi-target tracking shown in Section 5.1. Section 5.3 shows how closed world assumptions can be used for optimization of the image pre-processing. Section 5.4 gives a discussion and a summary of the chapter.

5.1 The Concept of Closed Worlds and Its Use in Tracking

The closed worlds is a concept that has been introduced to the CV community by Intille and Bobick [40] in 1994. The main idea of this concept is that for a given region in space and time, a specific context is adequate to explain that region (i.e. determine all objects within the region). This region is called the closed world and the context is a boundary in space of knowledge, outside of which knowledge is not helpful in solving the tracking problem. The closed world assumptions in [60] and [57] which have been used for tracking in sports domain are defined as follows (CWA stands for Closed World Assumption):

- (CWA1) The camera overlooking the playground is static, and positioned so that its optical axis is approximately perpendicular to the floor.
- (CWA2) The court (playing field) is bounded, and its model can be calculated.
- (CWA3) The players' textures can vary during the game; however, they are known at the beginning.
- (CWA4) A player cannot change his position completely arbitrarily due to the effects of inertia.
- (CWA5) At a given time two players cannot occupy the same position.

The closed-world assumption (CWA1) which concerns about the camera position implies that the players are viewed from above. Using CWA1, it can be assumed that a complete occlusion of a player is not likely to happen, because one player cannot be located on top of another during a regular match. So CWA5 is based on CWA1. A model for the background can be

Algorithm 5.1 Using space partitioning in tracking [60].

```
Initialize Tracker Positions
For t = 1,2,3,...
```

1. `Sort the trackers according to their weights`
2. `Initialize the seeds of partitioning to the current position estimates`
3. `For each tracker do`
   (a) `construct a partitioning using the current seeds`
   (b) `construct the a partition mask for the current tracker`
   (c) `run single player tracking algorithm`
   (d) `update the seed of the current tracker`

calculated as described in Algorithm 7.1 based on CWA2. CWA5 has been used by Kristan [60] in order to track multiple handball players. To track N players using CWA5 as described in [60], the search space is divided into N disjoint regions. Each region contains only one tracker. Based on this kind of partitioning the tracking can be done using Algorithm 5.1.

Here it is assumed that the input to the partitioning procedure is a set of tracker positions. These sets of positions are taken from current estimates of the tracker positions. The estimates can be computed from the past positions and based on the player motion model. In the implementation of [60] this kind of partitioning has been achieved using Voronoi Diagrams [6].

Voronoi diagram is a special kind of decomposition of a metric space determined by distances to a specified discrete set of objects in the space, e.g., by a discrete set of points. It is named after Georgy Voronoi, also called a Voronoi tessellation, a Voronoi decomposition, or a Dirichlet tessellation (after Lejeune Dirichlet) [114]. Voronoi diagrams are defined in [114] as follows: Let S be a set of points in Euclidean space with all limit points contained in S. For almost any point x in the Euclidean space, there is one point of S closest to x. The word *almost* is used to indicate exceptions where a point x may be equally close to two or more points of S. If S contains only two points, a and b, then the set of all points equidistant from a and b is a hyperplane. That hyperplane is the boundary between the set of all points closer to a than to b, and the set of all points closer to b than to a. It is the perpendicular bisector of the line segment from a and b. In general, the set of all points closer to a point c of S than to any other point of S is the interior of a (in some cases unbounded) convex polytope called the Dirichlet domain or Voronoi cell for c. The set of such polytopes tessellates the whole space, and is the Voronoi tessellation corresponding to the set S. If the dimension of the space is only 2, then it is easy to draw pictures of Voronoi tessellations, and in that case they are sometimes called Voronoi diagrams. Figure 5.1 shows how Voronoi Partitioning is used for tracking.

5.2 Enhancement by Local Partitioning Strategy

The experimental study in [60] has been done on sequences of video with low quality and small image size of 348x288, so the construction of Vornoi masks may not require a lot of time. Taking into consideration that the higher resolution digital cameras can be used for tracking the construction of such partitioning masks could be costly. Also the processing steps in Algorithm 5.1 may run for several times in the same frame and for each update of the tracker position the Voronoi diagram will be calculated. This means that the cost for constructing Voronoi masks will increase for higher quality images.

In the following an additional set of closed world assumptions will be introduced and its use together with the above set of assumptions to optimize the multi-player tracking will be explained. The additional closed world assumptions are:

- (CWA6) The distance between players during playing can be more than the distance that one player can achieve between two frames.

- (CWA7) An estimation of the player's next position can be computed.

- (CWA8) The initial position of the player is known at the beginning.

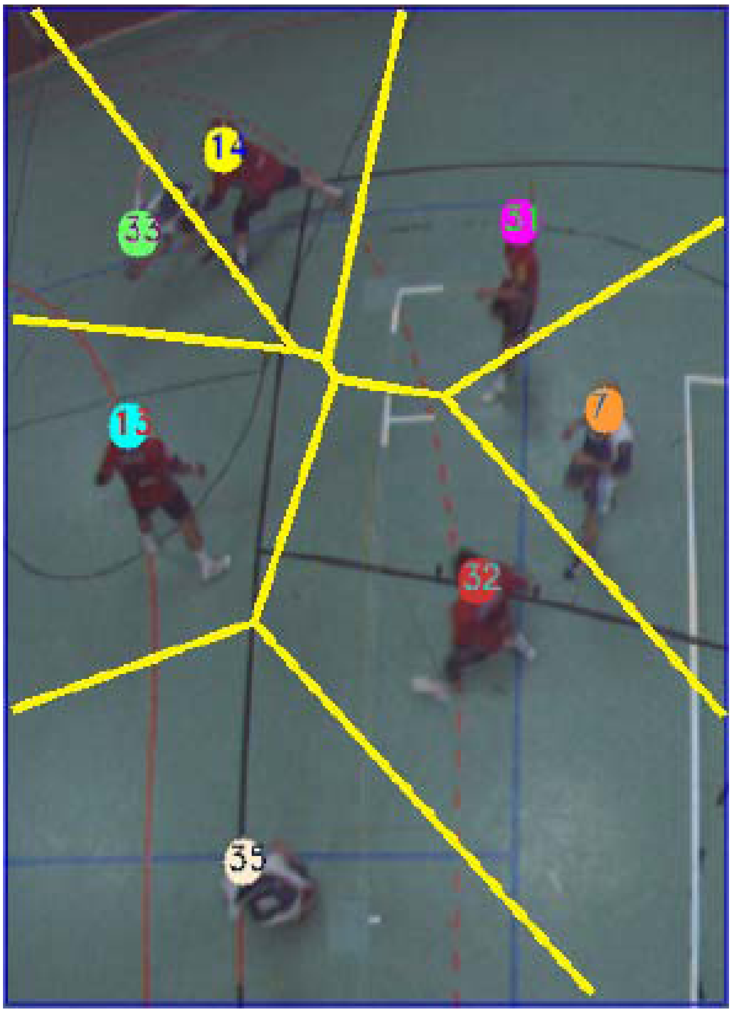


Figure 5.1: Voronoi partitioning for multiple player tracking.

- (CWA9) The player's height is known.
- (CWA10) The camera position and parameters are known.

Using the closed world assumption CWA6 and CWA7 it is not necessary to compute Voronoi masks for the trackers which are far away from others. If the estimated distance for this tracker is still greater than the distance that can be achieved in one frame, it will be a waste of time to construct the mask for this tracker. Also using the closed world assumptions CWA6 and CWA7 the order of tracker update is computed based on the distance between trackers which determines which one is in a critical situation (is more likely to confuse other trackers). Using this information can help to first update the tracker which may cause confusion. It is better to first update the trackers, which are nearer to others. The trackers are clustered based on the distance between them where each group of trackers are treated independently. Each group of trackers represents a possible conflict in tracking, so the processing will be concentrated on this group. For each group a separate partitioning is performed. Groups that only contain one tracker do not require any partitioning.

Taking into account the assumptions CWA6 and CWA7, the modified algorithm for multi-player tracking can be shown in Algorithm 5.2.

The grouping of the trackers is actually a clustering process where the number of clusters is not known, yet the width of the cluster is. The width of the cluster can be determined based on the maximum speed of the target between two frames. The adaptive c-means clustering algorithm [93] is very relevant for solving this problem. Only two parameters Θ_{new} and Θ_{merge} should be determined before the clustering begins. Θ_{new} is the threshold used to determine if a new cluster should be added and Θ_{merge} is the threshold to determine if two clusters should be merged.

The adaptive c-means clustering algorithm has been used to analyze tracking data taken from a basketball game in order to acquire an estimation of the average number of position data clusters in each frame. The data set consists of position data of 10 players from 2 teams in 16043 frames with a frame rate of 30 frame/second. Using the values of 150 and 200 for Θ_{new} and

Algorithm 5.2 Modified voronoi partioning based multi-target tracking.

```
Initialize Tracker Positions
For each input frame
```

1. `Compute the position estimates of the trackers`
2. `Based on tracker estimated positions cluster the trackers into groups`
3. `For each tracker group`
   (a) `Initialize the seeds of partitioning to the estimated positions`
   (b) `Sort the selected trackers according to distance matrix`
   (c) `For each tracker (starting from min to max) do`
      i. `construct the partitioning using current seeds`
      ii. `construct the partitioning mask for the current tracker`
      iii. `run tracking algorithm`
      iv. `update the seed of current tracker`

Θ_{merge} respectively, it is found that the average number of clusters in each frame is 5.91 and the standard deviation is 1.42. Regarding the number of samples (player positions) in each cluster it was found that 57.6% of the total clsuters contain only one sample, 26.2% contain 2 samples, 8.8% contain 3 samples, 4.8% contain 4 samples and less than 3% contain between 5 and 8 samples. It is clear that in more than half of the number of clusters that partitioning is not required and in more than 25% only a simple paritioning between two trackers is needed.

To partition a sub-image with 2 trackers the intersection between two rectangles can be calculated as shown in Figure 5.2.

5.3 Optimizations Based on Closed World Assumptions

Using CWA7 and CWA8 more optimization can be done regarding the preprocessing described in Section 2.1. These optimizations include the preprocessing of only the sub-image where the tracking should take place. Be-

Algorithm 5.3 The adaptive c-means algorithm

`Estimate the thresholds` Θ_{new} `and` Θ_{merge}
`set` $K = 0$
`choose a data point` x
`calculate the distance` $d_j = d(x, m_j), j = 0, 1, ..., k$
`find` $j^* = \arg\min_j d_j$
`if` $d_{j^*} > \Theta_{new}$ `or` $k = 0$

1. $m_k = x$
2. $k = k + 1$
3. `set` $n_{j^*} = 0$

`else`

1. $n_{j^*} = n_{j^*} + 1$
2. `replace` m_{j^*} `by` $m_{j^*} + (\frac{1}{n_{j^*}})(x - m_{j^*})$
3. `calculate` $D_l = d(m_l, m_{j^*}), l = 0, 1, ..., k$
4. `find` $l^* = \arg\min_{l \neq j^*} D_l$
5. `if` $D_{l^*} < \Theta_{merge}$
   - $merge(m_{l^*}, m_{j^*})$
   - $k = k - 1$

`goto: choose data point`

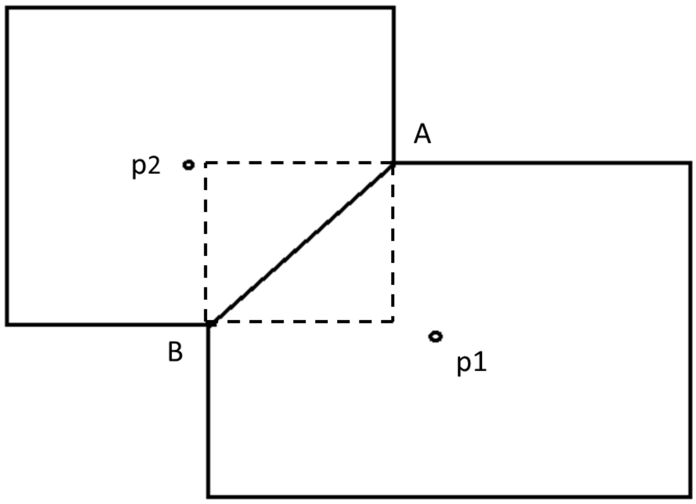


Figure 5.2: Partitioning a sub-image containing two trackers.

cause an estimation of the player position can be calculated (CWA7), the size and location of the sub-image which may contain the player in the next frame can be calculated and used to run the pre-processing only on this ROI.

CWA9 and CWA10 are used to calculate the player's position in meters and to perform the tracker transfer from one camera view to the other as described in detail in Sections 7.1.2 and 7.1.2.

5.4 Discussion

In this chapter the multi-player tracking in the context of closed worlds has been introduced. Kristan [60] introduced 5 closed world assumptions and used them in the tracking of multiple sports players by partitioning the search space into disjoint regions where each region contains only one tracker. Another 5 closed world assumptions have been introduced in this chapter and used to enhance the multiple player tracking as well as the image pre-processing. Clustering of position data from basketball game has been used to show that only in 16.6% of the frames partioning of space to more than three partitions is needed and in 25% only a simple partioning for two trackers is needed. In Chapter 7 experiments will be discussed to show the effect of using closed world assumptions in tracking accuracy and effiecieny.

Chapter 6

Sport Performance Analyzer Software

In this chapter the development of a software system that enables normal users to use the tracking algorithms described in this thesis will be explained. SPA (Sport Performance Analyzer) is the name of the software system. SPA is the output of the project *Intelligent Sports Wear and Automatic Analysis of Team Sports* (see Section 1.2). It is a system for acquisition, analysis and visualization of both physiological and position data of indoor sports players to provide a tool of players' performance analysis. The intended users of SPA are team coaches, sports science researchers, sports medicine researchers and other researchers who are interested in the human motion analysis or the analysis of the team interactions as a dynamic system.

The rest of this chapter will be organized as follows: Section 6.1 gives an idea about the whole system through data flow between the different modules in the system. Section 6.2 explains the software architecture of the SPA. Sections 6.3, 6.4 and 6.5 explain in more detail the recording, tracking and visualization modules respectively. In section 6.6 SPA utilities for testing of the tracking algorithms are presented. Finally Section 6.7 gives the conclusion of the chapter.

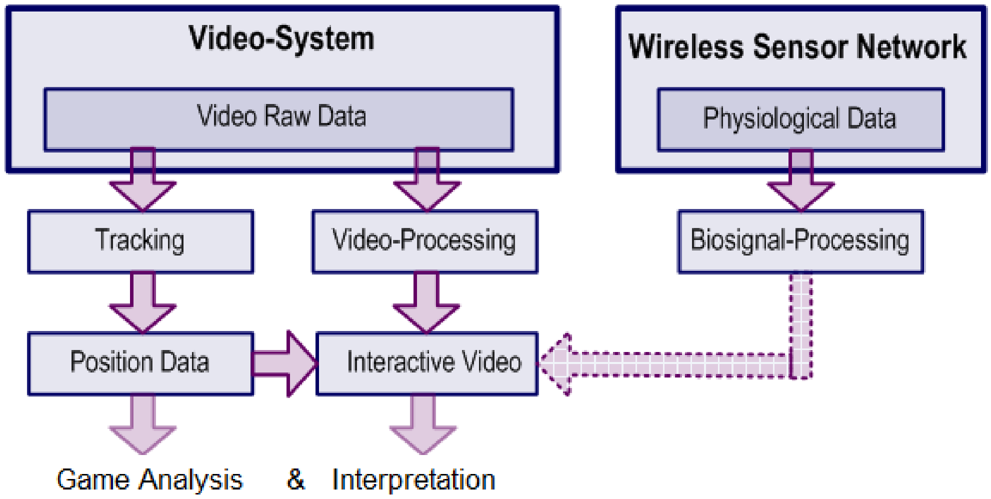


Figure 6.1: Data flow diagram for the SPA software system [83].

6.1 Structure of SPA

SPA has three main modules, namely the video-sensor data acquisition module (Video-System and Wireless Sensor Network), the vision-based tracking module and the analysis-visualization module. The best way to describe SPA is a data flow diagram that shows the flow of the data between the different modules. This is shown in Figure 6.1. The acquisition module is responsible for recording bio-signal data and video streams. In Figure 6.1 the data acquisition module is shown in two components. The first component is the video system which is composed of two high-quality digital cameras that produces raw (uncompressed) video. The video stream is used as input to the tracking and the visualization modules. The second component of the acquisition module is the wireless sensor network which is used to acquire the physiological data of the players.

The processing of the bio-signal acquired from the sensor is done on-board before it is sent to the recording computer via a wireless network. Bio-signal data, video data and position data are inserted to the analysis and visualization module which produces different visualizations such as graphs and interactive video with annotated information. The video tracking module works offline to provide the position data of the players. It takes as input two video data streams and produces the positions of the players in real world coordinates (meters) which can be further analyzed to acquire more information about the movement of the players.

The recording of video and physiological data streams is synchronized so that to make further analysis of the data easier. Because the amount of physiological data is much less compared to the video data it can be processed and visualized online. So for example during the sports event the heart rate can be monitored and the trainer can decide to substitute a player based on his heart rate profile.

6.2 Software Design

In order to develop software products to be used by a large number of users, it needs a systematic process of analysis, design, development, test, maintenance, and documentation. The targeted users of SPA are researchers in the sports domain and team trainers. Therefore, a systematic development of such a system is a must.

Software engineering techniques have been used during the development of the SPA system. Software engineering can be defined as the systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software. Details of the phases of the SPA Software development can be found in [101, 32]. In the following sections a brief description of the design and the user interface and tools used for development will be shown.

Figure 6.2 shows the software architecture of the SPA. The acquisition module is responsible for the recording of both video and sensor data. The GUI (Graphical User Interface) will provide the acquisition module with the necessary information from the Game-Team-Sport module or for short the database module. This information includes the specification of the sport hall (such as height of the camera) and the kind of game (for example handball or basketball). This information is important to specify the video-frame size. For the sensor data it is important to get the team and player names from the database in order to assign the current sensors to the players. A

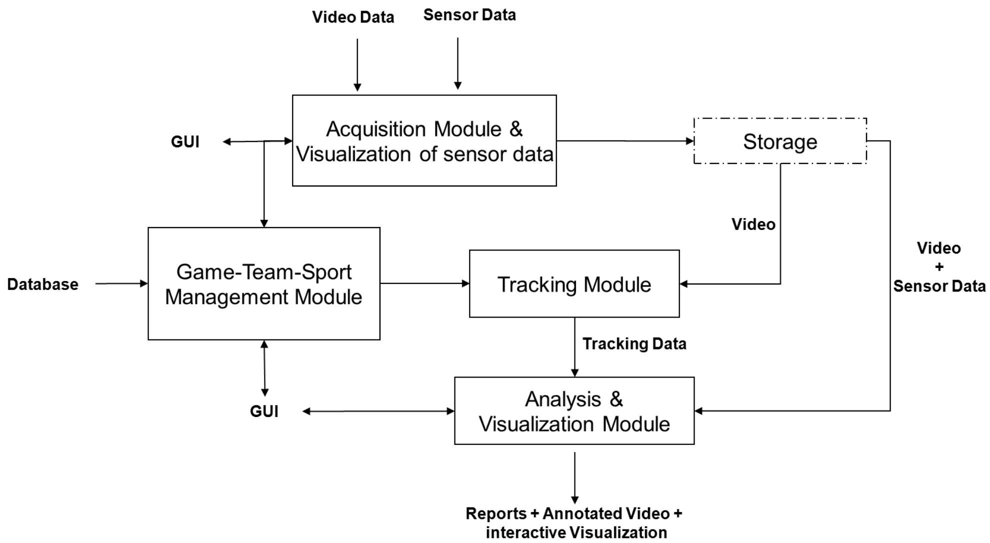


Figure 6.2: SPA software structure.

user friendly interface will allow the user to easily access this information and to control the recording process with all possible events like *stop*, *start* and *pause* functionalities. Both the video and sensor data are stored for further analysis and visualization. The video data can be read and processed for tracking by the tracking module which outputs the tracked player positions. The Analysis and Visualization module is responsible for generating reports and visualizations which help the users to interpret the tracking output and physiological data and draw conclusions about the players' performance during the game or the training session. The Game-Team-Sport Management module provides all other modules with the necessary information about players, teams, camera setup, games and so on.

6.3 Recording Video and Bio-signals Data Module

The recording module is responsible for recording video and/or physiological data. The video data is recorded from one or two cameras while the recording of physiological body signals is done by a wireless sensor network. In this

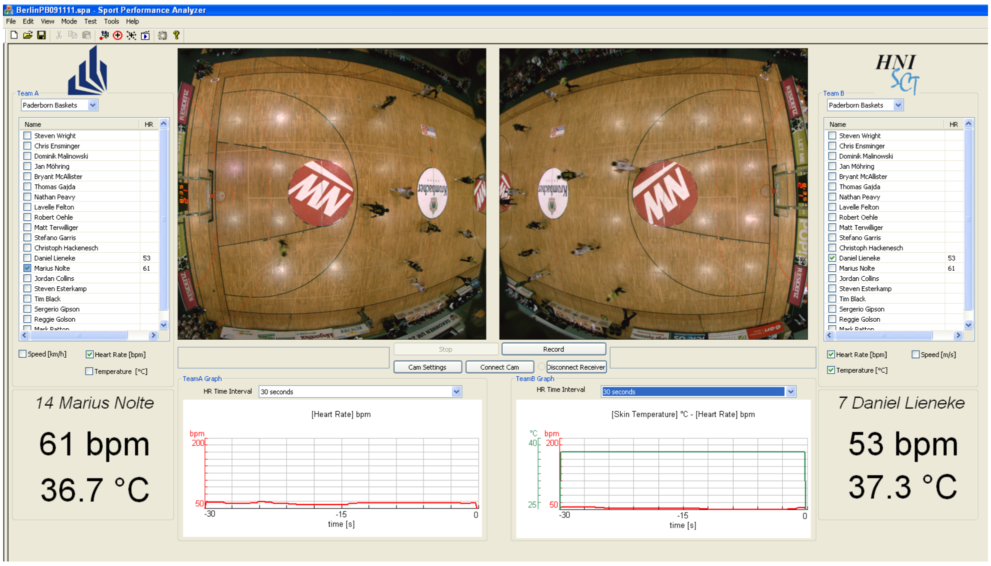


Figure 6.3: Screen shot of SPA recording module user interface.

section only a brief description of the sensor network with more details on the software part will be introduced. A screen shot of the recording module in the SPA software can be seen in Figure 6.3. Because of the small amount of data (relative to video data) received from the sensors an online visualization while recording is possible. This includes graphs with a variety of colors for different players and signals coming from the sensors.

6.4 Tracking Module

The tracking module performs the tracking algorithm on the inserted images and puts out the tracking data, which is saved in separate files for each tracker. At the beginning of the tracking process, the user should specify the initial position of the tracker and which player shall be tracked. The player information, such as name, shirt number, height and team are read from the database.

From a software design point of view the tracking module is the most complex module in the whole SPA. Figure 6.4 shows a simple class diagram of the tracking module where only the important classes are shown. To give

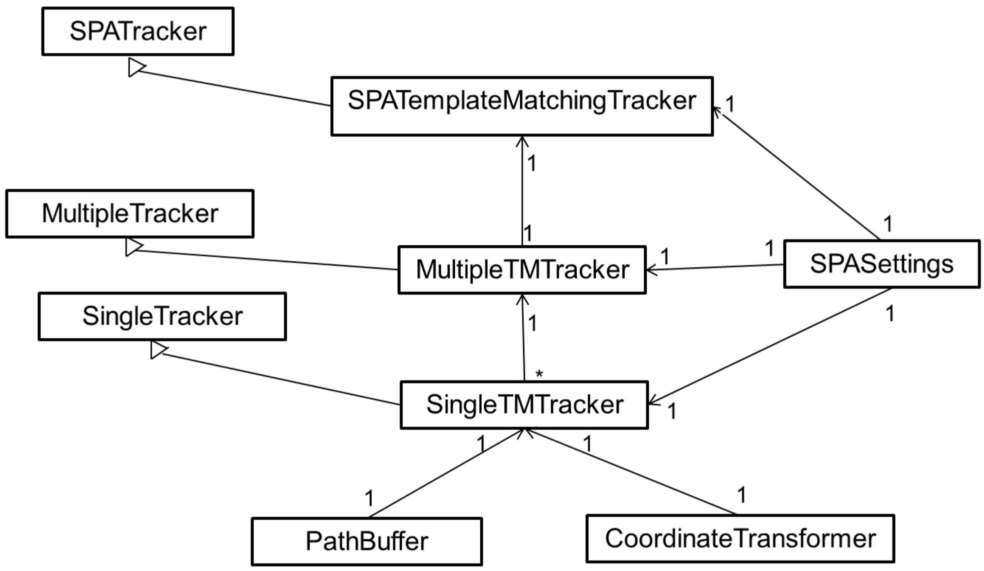


Figure 6.4: Class diagram of the SPA tracking module

an idea how the tracking works, in principle, there should be a main class which runs as a thread. It is inherited from the base class *SPATracker* where every new tracker should be inherited from this class.

In Figure 6.4 the class *SPATemplateMatchingTracker* is an example of a tracker based on Template Matching and is inherited from *SPATracker* class. This *main* class is responsible for pre-processing (preparing) the image, managing the user interaction events and showing the images after tracking with the current positions of the tracker overlayed on the displayed image.

SPATemplateMatchingTracker is also responsible for managing an object from *MultipleTMTracker* class (is inherited from *MultipleTracker* class) which in turn takes care of a collection of *SingTMTracker*. *MultipleTracker* will allocate memory space for new trackers and also implements the multiple tracker handling technique. Every *SingleTMTracker* object is responsible for tracking one player using template matching techniques. Every *SingleTMTracker* has also its own *PathBuffer* which is a special circular buffer that keeps a specific number of position data points in memory in case the user wants to step back in video to see how the tracking was in the last, for example, 30 frames. When the position data is *old enough* it will be written

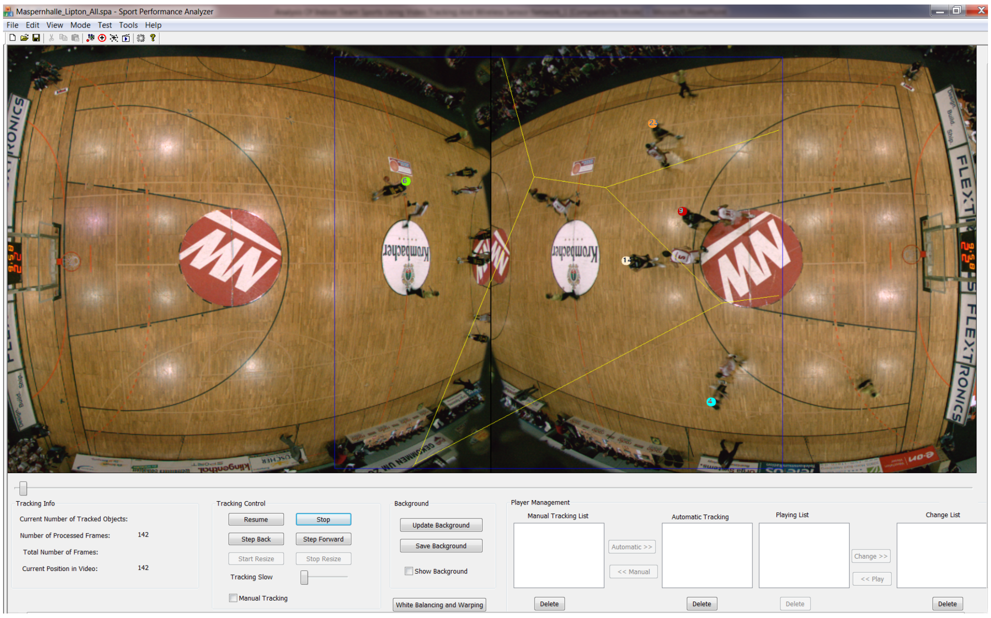


Figure 6.5: A screen shot for the tracking mode user interface.

on the hard drive. The *CoordinateTransformer* object will do the necessary transformation of the position data from image coordinate to real world coordinates.

Figure 6.5 shows a screen shot from the tracking mode. Various controls are available for the user in the tracking mode, including start, stop, pause, step back and step forward. It also allows the possibility to substitute players and shows a list of substituted and playing players. In case of very accurate manual tracking the tracking user interface allows to track specific players manually, which means that to manually correct the tracker in each frame.

The tracking algorithm parameters and all other parameters like video and output files can be configured from the settings dialog as shown in Figure 6.6. All the settings in the settings dialog can be saved as a project so the user can stop tracking at any time and continue later.

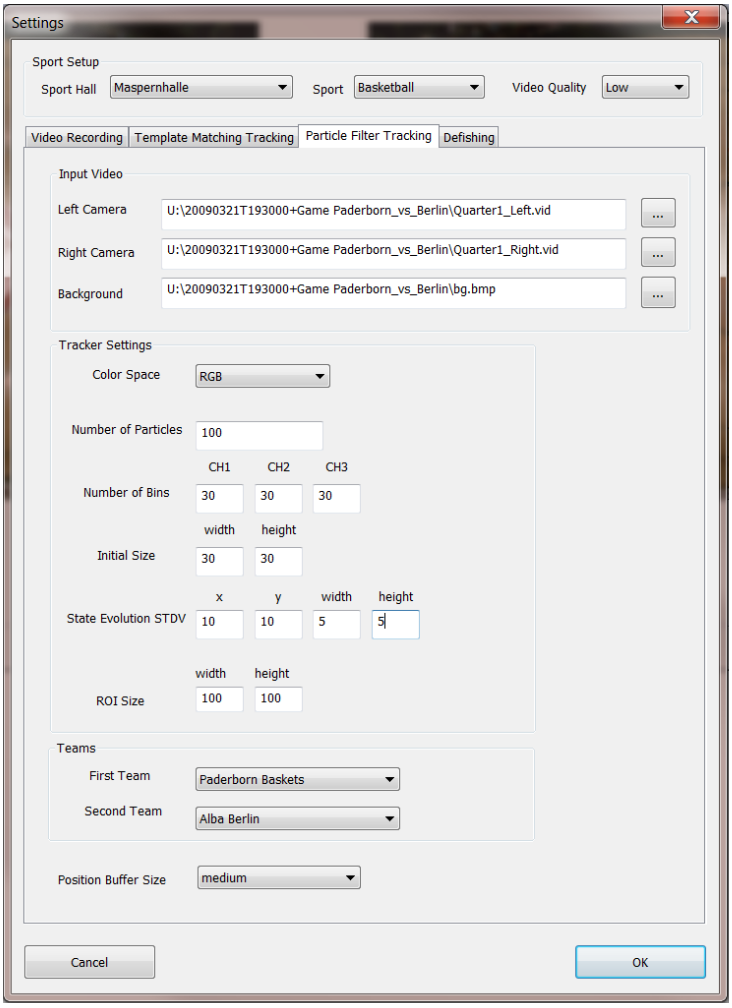


Figure 6.6: A screen shot form the SPA settings dialog.

Figure 6.7: Screen shot for the interactive visualization tool.

6.5 Analysis and Visualization Module

The visualization mode in the SPA allows the user to view and generate videos with augmented tracking information. The generated videos can be of different resolution and quality based on the user selection. The visualization module includes some classes which are responsible for correcting the camera fisheye lens distortion and correcting the small errors of the camera installation which needs *Warping* of the whole image. Because the warping and undistortion are operations which depend solely on the pixel coordinate and not on the pixel value, lookup tables are used in order to reduce the time needed for processing.

The visualization done using the SPA visualization module is a *static visualization*, i.e. the user cannot select the information that he would like to see during the video displays. An interactive visualization tool [99] also developed at the Systems and Circuits research group especially for the visualization of robot tracking data in the Teleworkbench project [107] has been used after some adaptations to *interactively* visualize the player tracking data, including the position and sensor data. Figure 6.7 shows a snapshot from this visualization tool.

Important data for the trainers and sport scientists include speed profile of the player(s), field coverage (covered distance), average speed. A special MatLab tool has been developed to help generate different visualizations and statistical reports. The details of the tool can be found in [32].

6.6 Utilities for Testing

SPA software includes utilities to track players manually in order to acquire highly accurate tracking data of the players. Manual tracking can also be used to generate ground truth test data which is used as benchmark data for testing tracking algorithms. Benchmark data is used to evaluate the performance of the tracking in terms of accuracy. In order to test different parameters of the tracking algorithms a tool which takes the benchmark files as input is developed. It lets the tracking algorithm run with a different a set of values for a specific parameter and produces the required statistics to evaluate the tracking algorithm.

6.7 Discussion

This chapter gives a brief overview of the SPA software which was developed during the course of this thesis work. SPA is also a part of the sport project at the System and Circuit Technology research group. An idea about the design of the software together with screen shots from different modes has been presented. The details of the software parts that are not directly related to the domain of this thesis can be found in the appropriate references referred to in the thesis.

SPA is a tool that enables users to acquire a new combination of data about the sports players. The combination of physiological and position data provides new possibilities for analysis of indoor sport games. The acquisition, tracking, visualization and analysis of these combination of data is done in one system through a user friendly graphical user interface.

SPA is being used for the analysis of official basketball games played

Paderborn and the handball games played in Soest. It is also used for the analysis of training sessions of handball at the University of Paderborn. The analysis results are used by the department of Sport Medicine and the department of Applied Mathematics at the University of Paderborn for further research at the levels of Bachelor, Master and PhD in these departments.

Chapter 7

Experimental Study

In the previous chapters methods for tracking sports players based on template matching and particle filters have been presented. The proposed algorithms have parameters, such as template size (in template matching), number of particles and number of color histogram bins (in particle filters) that still need to be determined. The tracking algorithms need also to be evaluated in terms of running time, memory requirements and accuracy. In order for the SPA software system to be used to support sport scientists and team trainers to perform experiments and studies or to evaluate the performance of the athletes during competitions and training sessions, it should satisfy some requirements regarding the validity, objectivity and reliability of the tracking.

This chapter presents a comprehensive experimental study on the tracking algorithms and their implementation in the SPA software. The chapter explains the experimental setup and the various numerical results of the tests done for the evaluation of the tracking algorithms. The rest of the chapter is organized as follows: Section 7.1 explains in detail the experimental setup which includes the video data acquisition, pre-processing of image data, post processing of position data, implementation and running times of pre- and post- processing. Section 7.2 gives a description of the datasets used for testing. Section 7.3 presents definitions of the performance measurements and the quality criteria of the tracking system, it also shows the hypotheses about the system that need to be proved. Sections 7.4, 7.5 and 7.6 give the numerical results including tests for the parameters of the template matching and particle filter tracking algorithms, respectively. Section 7.7 shows the evaluation of the numerical results of the tests used to evaluate the hypotheses presented in Section 7.3. Section 7.8 gives discussion and concluding remarks on the chapter.

7.1 Experimental Setup

The selection and installation of the image acquisition hardware is of great importance because all further processing is based on the input image and its quality. Figure 7.1 shows the different possible camera installations in the sports halls. Using one camera installed on the side of the sport hall (Figure 7.1(a)) will cause total or partial occlusions to occur more often. This requires more than one camera to solve this problem (Figure 7.1(b)). In this case one player will appear in more than one camera view which requires data fusion between the different views. The configuration which uses overhead cameras has been chosen because it reduces the occlusion and keeps the player in the view of the camera all the time. In this configuration two cameras are needed (one for each field) in order to cover the full playing field.

The two cameras are stationary. The selection of the two cameras has been carefully done after testing different cameras and lenses. Because most of the sport halls are not high enough, fisheye lenses have been used in order to catch the required view on the image sensor. Figure 7.2 shows the camera installation configuration. The acquired image from this configuration is shown in Figure 7.3.

The use of fisheye lenses causes the problem of distorting the acquired images, deeming it insufficient for viewing the video with or without visualizations. Thus, undistortion is required in order to present a better visualization of tracking results. On the other hand, because it requires interpolation, undistortion will cause losing or *distorting* of some image features that could be important for the tracking algorithm. For this reason it is preferred to

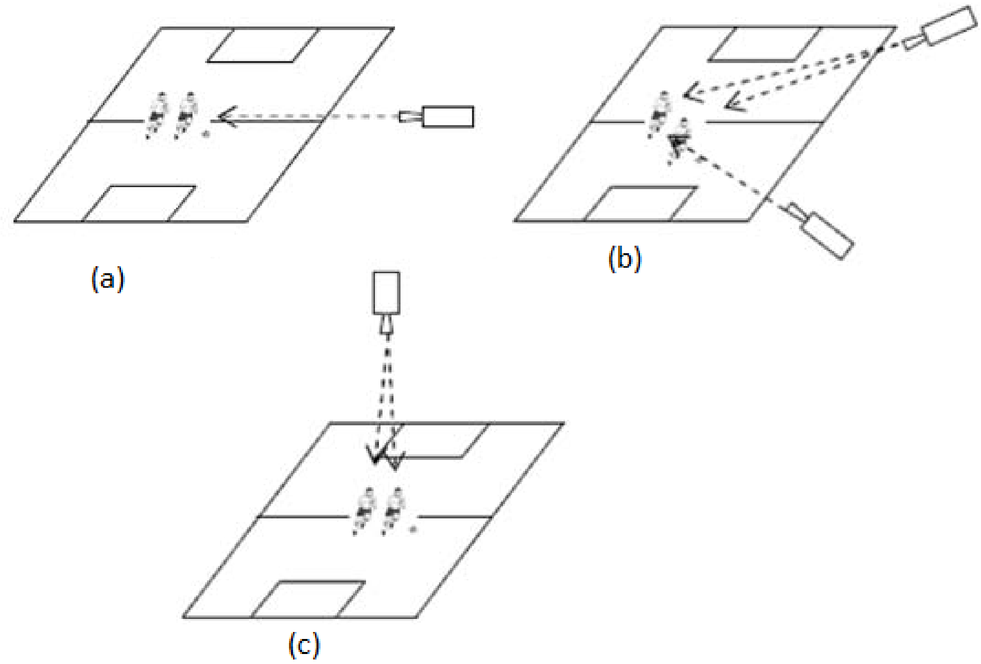


Figure 7.1: Possible camera installations in the sports hall.

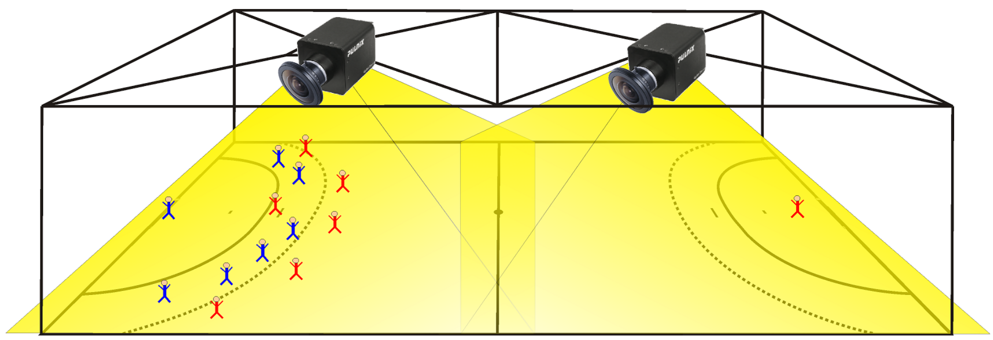


Figure 7.2: Camera installation in the sports hall [83].

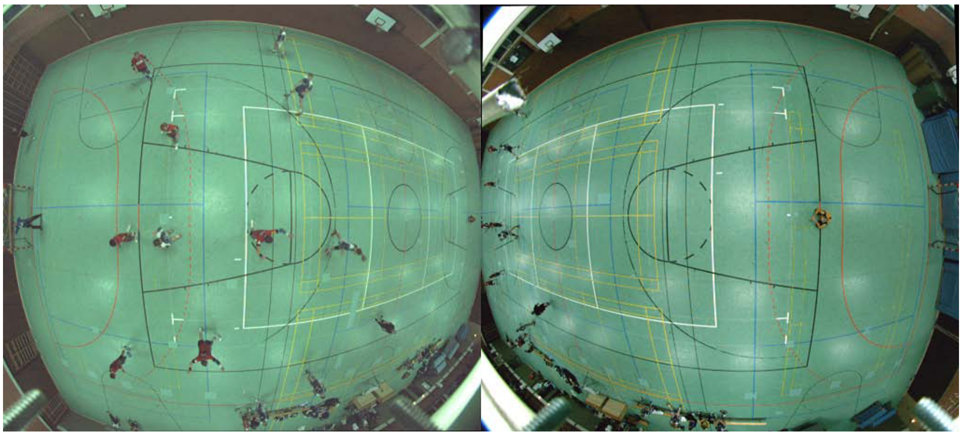


Figure 7.3: Images obtained from the two cameras installed in the University of Paderborn sports hall.

track the players in the fisheye image.

The images acquired from the two cameras are Bayer images which are produced by applying special color filters over the photo sensors of the camera. Further processing is needed in order to acquire normal color RGB (Red, Green, Blue) images. More details about processing Bayer images is given in Section 7.1.1.

The installation of the cameras has been done in three sports halls. The first is the sports hall of the University of Paderborn, the second is the Maspernhalle which is the main indoor sports hall of Paderborn and the third is in the Bördehalle in Soest. The specifications of the cameras are shown in Table 7.1.

The specifications of the computer system used for the testing are as follows:

- Processor: Intel Core i7-950 Series.
- Graphic Card: NVIDIA GetForce 480 GTX
- RAM: 8 GB (DDR3-1333)
- Mainboard: ASUS P6T WS PRO
- Operating System: Windows 7 Enterprise 64 Bit

	Uni-Paderborn	**Maspernhalle**	**Bördehalle**
Image Size (pixel)	1392×1040	1392×1040	1392×1040
Sensor	bayer pattern 2/3” CCD - 8 bit	bayer pattern 1/2” CCD - 8 bit	bayer pattern 2/3” CCD - 8 bit
Frame Rate	30 fps	30 fps	30 fps
Connection	Gigabit Ethernet	Gigabit Ethernet	Gigabit Ethernet
Objective	Fisheye f=2.7mm	Fisheye f=2.7mm	Fisheye f=2.7mm
Pixel Size	6.45 μm	4.65 μm	6.45 μm
Camera Hieght	7.14m	9m	7.25m

Table 7.1: Technical Specifications of the Cameras Installed in the University of Paderborn Sport Hall, the Maspernhalle Sport Hall and Bördehalle

7.1.1 Pre-processing of Video Data

The image processing step shown in Figure 2.2 concerns the preparation of the image before it is inserted to the tracking algorithm. The preparation may include image color space transformation such as conversion from Bayer-Pattern to RGB or from RGB to gray or HSV color space. Another part of pre-processing may be the correction of the different camera distortion including the fisheye effect and errors in camera orientation. An important pre-processing step is background/foreground segmentation in which parts of the image where motions occur are extracted or identified. This helps to reduce the search effort in the tracking algorithm. In the following the pre-processing implemented for the tracking of the sport player in this thesis will be presented together with the implementation results.

Color Space Conversion and Enhancement Filtering

The used cameras produce Bayer images which need further processing to generate images in RGB format. The Bayer images are produced by using CFA (Color Filter Arrays), which are a set of filters placed over the image sensors to capture color information.

Color filters are necessary because the typical photo sensors detect light intensity with little or no wavelength specificity and therefore cannot separate

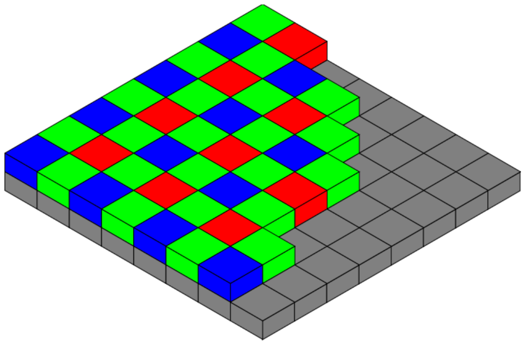


Figure 7.4: The arrangement of Bayer color filters on the pixel array of an image sensor [110].

color information. Most camera sensors use the Bayer filter as CFA. Bayer filter mosaic was invited by Dr. Bryce E. Bayer of Estman Kodak [109]. Figure 7.4 shows how the Bayer color filters are arranged over the image sensor.

The filter pattern is also called GRGB or other permutation such as RGGB because it has 50% green, 25% red and 25% blue. The raw output of Bayer-filter cameras is referred to as a Bayer pattern image [109]. Since each pixel is filtered to record only one of three colors, the data from each pixel cannot fully determine color on its own. To obtain a full-color image, various demosaicing algorithms can be used to interpolate a set of complete red, green, and blue values for each pixel. Different algorithms requiring various amounts of computing power result in varying quality of final images. This can be done in-camera, producing a JPEG (Joint Photographic Experts Group) or TIFF (Tagged Image File Format) image, or outside the camera using the raw data directly from the sensor [109].

The algorithms used to demosaic the Bayer pattern will be referred to as *Debayering* algorithms. Debayering algorithms range from simple algorithms with low quality output to sophisticated ones with high quality output. The simplest method is nearest-neighbor interpolation which simply copies an adjacent pixel of the same color channel. It is unsuitable for any application where quality matters, but can be useful for generating previews given limited

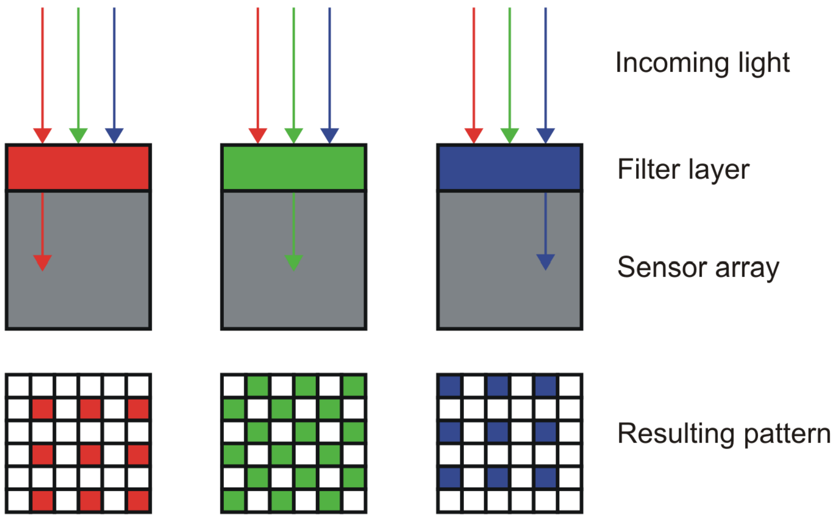


Figure 7.5: Cross section of image sensor with filter layer and the resulting pattern[112].

computational resources [113].

Another simple method is bilinear interpolation, whereby the red value of a non-red pixel is computed as the average of the two or four adjacent red pixels, and similarly for blue and green. More complex methods that interpolate independently within each color plane include bicubic interpolation and spline interpolation. Bilinear interpolation has been used in this work for debayering. More complex debayering algorithms exploit the spatial and/or spectral correlation of pixels within a color image [21]. Spatial correlation is the tendency of pixels to assume similar color values within a small homogeneous region of an image. Spectral correlation is the dependency between the pixel values of different color planes in a small image region.

White Balance

When capturing a scene by camera the lighting plays an important role in the quality of the scene image. If the lighting is too strong or too weak the image should be enhanced by means of white balancing. In principle one wants to scale the RGB color channels so that the objects that are believed to be neutral (white) appear so [108]. For example, if a part of the image which is believed to be white has the following values for RGB: $R' = 240$, $G' = 230$, $B' = 200$ and assuming the maximum value of all color components is 255 then the white balance is simply done by multiplying each red component by $255/R'$, each green component by $255/G'$ and each blue component by $255/B'$.

Correction of Camera Geometric Distortions

During the camera installation in the sports hall small errors in the position or the orientation of the camera in the range of millimeter may cause bigger errors during the exact computations of the player positions for the tracking process. In addition, after the installation the position of the camera may be changed due to ball hits, for example. In order to correct such errors a calibration is needed. In the calibration process, four points of known positions on the image are taken and used to build the transformation matrix in order to be used for the correction of each pixel of the frame.

As described in 7.1 the lens used for the cameras produces a fisheye effect on the image. The tracking is done on the fisheye-distorted image which needs to be corrected when calculating the player position in meters. The correction is also needed for generating the visualization of the tracking information (for example, annotated video). The details of the correction of the fisheye distortion are explained in 7.1.2.

The two operations of correcting camera orientation and fisheye-distortion are based on the pixel-position, not the pixel value, so LUT's (Look Up Tables) can be calculated once and then used for each image saving a lot of processing time. In the tracking case it is only required to correct one pixel for each tracker which means 15 look up operations at most when tracking all the handball players and the referee.

Background Detection and Subtraction

Background subtraction is used to segment the scene into motion and non-motion parts. The segmentation is performed by comparing the current frame with an estimated background model. The motion parts occur where

Algorithm 7.1 Background estimation algorithm.

```
For each input frame
```

1. `convert image to gray scale`
2. `smooth the image by Gaussian filter`
3. `compute difference between current and previous image`
4. `binarize the image by thresholding`
5. `perform dilation on the binary image`
6. `copy the non-motion areas to the background image model`

a significant change between the current frame and the background takes place. In the application of sports player tracking the background model can be taken as the image of an empty (with no players) field but if the lighting conditions changed the background model may not be valid for detection of moving foreground objects and need to be updated. Therefor, in this thesis work, the background model is built using a number of frames from the recorded video before the start of the tracking process. Algorithm 7.1 shows the procedure for the background estimation.

The idea of the algorithm is to compute the difference between each two successive frames and copying the non-motion (zero-difference) pixels to the background model. The input to Algorithm 7.1 is a color image which is converted to a gray scale image. First, a Gaussian filter is applied to smooth the image. In order to detect the motion parts the difference between two successive frames is computed. A threshold operation is then applied in order to mark each pixel as foreground or background which means it converts the gray scale image to a binary one. In order to fill the holes in the background or foreground regions a morphological dilation filter is applied.

If the image is of size $N \times N$ and assuming that the size of the filter is constant then the time complexity of the filtering operation is $\mathbf{O}(N^2)$. Image subtraction and thresholding are of $\mathbf{O}(N^2)$ complexity. Dilation is a basic operation in the area of mathematical morphology. It is usually applied on binary images, but there are variations that work on gray scale images.

The basic effect of the operator on a binary image is to gradually enlarge the boundaries of foreground pixel regions (i.e. white pixels). Thus areas of foreground pixels grow in size while holes within those regions become smaller. Being a kind of filtering that operates on each pixel of the image makes its complexity also $\mathbf{O}(N^2)$.

For the space complexity the resulting image is of the same size as the input image. Thresholding the image needs no temporary image to store the result. Other operations like smoothing require an image to store the result. Each step in the algorithm, except the thresholding step, requires a temporary image of the same size as the input image to save the result.

Algorithm 7.1 is used to build a background model offline before the start of the tracking. It has been shown that between 200 and 300 frames (i.e between 6 and 10 second when the frame rate is 30 fps) can be used to build a good background model. In indoor sports the lighting conditions have no sharp changes during the game because the light is artificial and non-moving, unlike sun light, so the background model dose not need to be updated.

7.1.2 Post-processing of Position Data

The output of the tracking algorithms are pixel positions on the image coordinate system. In order to calculate further information such as distance and speed of the players the pixels positions should be calculated to feet positions in the real world coordinate in meters. The calculations of the meter positions of the player are not simple for three reasons. First, due to the geometric distortions of the image on which the tracking takes place coming from two sources: the distortion due to the fisheye lenses and the errors in the camera orientation during its installation. Second, the tracker is actually tracking a point of the player body which is the head in case of template matching tracking and the center of the body in particle filter tracking. So a geometric model should be found to correctly calculate the position of the feet. Third, the tracker should be transfered from one camera view to the other one when the player crosses the middle line of the playing field. Figure

7.6 shows the steps of post-processing needed in order to calculate the positions of the players in meters. It is not shown how the tracker is transfered from one camera view to the other. This will be explained in detail in this section. The tracking takes place on the fisheye image (step 1), the position is corrected to coordinate on the undistorted (defished) image.

Defishing: Correcting Fisheye Lens Distortion

The correction of fisheye lens distortion is needed for the calculation of the player feet position in meters and for generating the video used for the visualization. The transformation from distorted to undistorted image is based on the following equation [115]:

$$r = f \tan(\frac{\grave{r}}{f}) \tag{7.1}$$

Where r is the distance to the center of the corrected (*Defish*) image, $\grave{r}$ is the distance to the center of the distorted (*Fisheye*) image and f is the focal length (in pixels). This transformation is based on the assumption that the lens is spherical and the distortion is only radial and not tangential. Based on radial distortion assumption the following relation holds:

$$\frac{r}{\grave{r}} = \frac{x - x_c}{\grave{x} - \grave{x}_c} = \frac{y - y_c}{\grave{y} - \grave{y}_c} \tag{7.2}$$

where $\grave{p} = (\grave{x}, \grave{y})$ is a pixel position at the distorted image and $p = (x, y)$ is the position of the mapping of $\grave{p}$ on the undistorted image and $\grave{p}_c = (\grave{x}_c, \grave{y}_c)$ is the center of the distorted image and $p_c = (x_c, y_c)$ is the center of the undistorted image.

Solving Equation 7.2 for x and y we get the following:

$$x = (\grave{x} - \grave{x}_c)\frac{r}{\grave{r}} + x_c \tag{7.3}$$

$$y = (\grave{y} - \grave{y}_c)\frac{r}{\grave{r}} + y_c \tag{7.4}$$

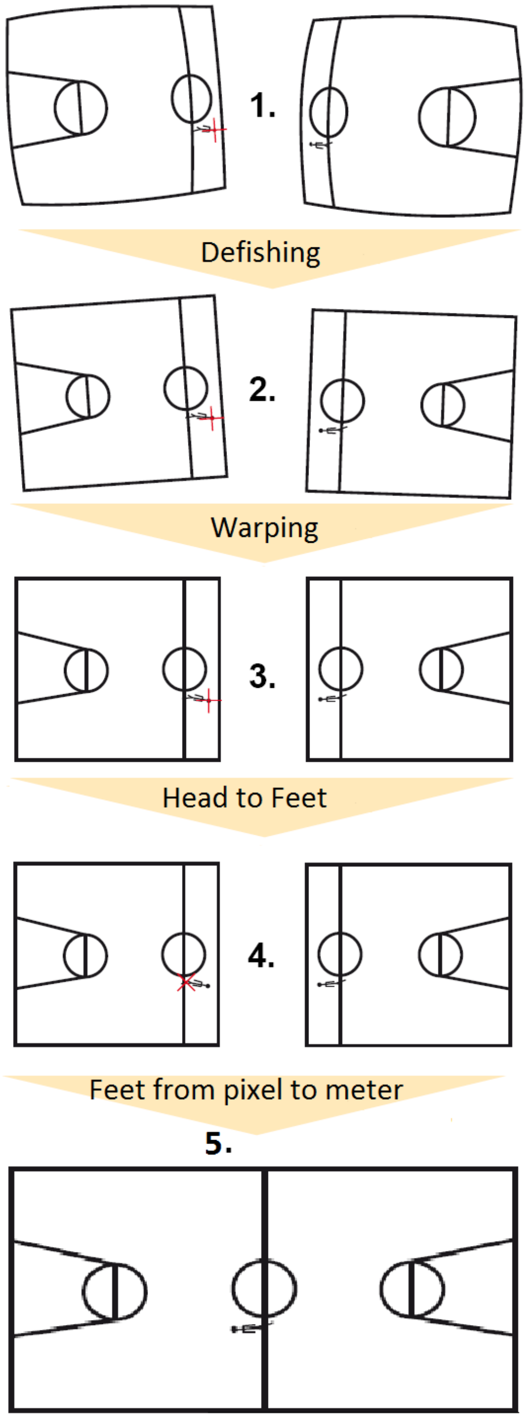


Figure 7.6: Post-processing of tracker output.

where

$$\grave{r} = \sqrt{(\grave{x} - \grave{x}_c)^2 + (\grave{y} - \grave{y}_c)^2} \tag{7.5}$$

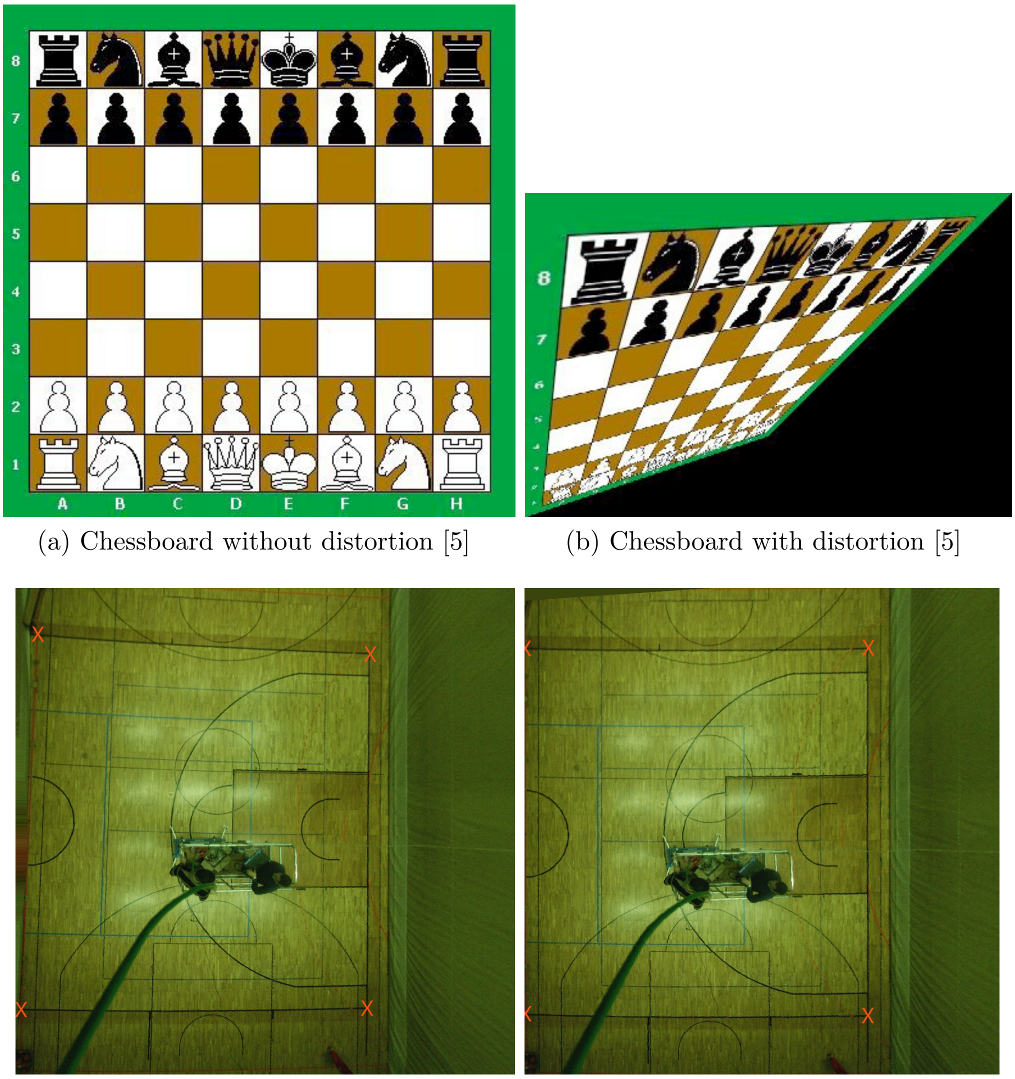
Warping: Correction of Perspective Camera Distortion

The common types of geometric distortions are caused from changing the image by shifting, scaling or rotation. Perspective distortion is caused by false camera positioning which is the case in our camera setup where the camera is fixed and its optical axis should be perpendicular to the playing field. The errors in positioning occur by changing the camera orientation when, for example, it is hit by ball or when it is not correctly installed. The perspective distortion of the image causes inaccurate calculation of the player position. Examples of perspective distortions are shown in Figure 7.7 which shows both distorted and undistorted images of chessboard and Maspernhalle sports hall.

In order to correct the perspective distortion a mapping from the distorted image (source) to the undistorted (destination) image should be found. This mapping is usually called *warping* which is a combination of a set of two dimensional transformations (shifting, scaling, etc.). The mapping can be described as $\acute{p} = M * p$ where p is the pixel in the source image, $\acute{p}$ is a pixel in the destination image and M is a transformation matrix. So the warping is in principle a matrix multiplication for each pixel coordinate in the image. In order to find the transformation matrix a set of points from the source image with known positions in the destination image are used to calculate the transformation matrix. In our camera setup the set of points in the source image can be the four points marked with red shown in Figure 7.7c and their mapping in the destination image are shown in Figure 7.7d. The details for the computations of the transformation matrix can be found in [5].

Calculating Player's Feet Position in Meters

After correction of the fisheye and geometric distortions of the image the position of the feet of the player should be found and transformed to meters. In template matching tracking the tracked object is head of the player while



(a) Chessboard without distortion [5]

(b) Chessboard with distortion [5]

(c) Maspernhalle with distortion

(d) Maspernhalle without distortion

Figure 7.7: Example of image warping of chessboard (a and b) and from the camera installed in Maspernhalle sports hall in Paderborn (b and c).

in particle filter it is the body of the player. Figure (7.8) shows the position of the camera and the player. In the figure the position of the mapping of player's head on the image is denoted as $p_h = (x_h, y_h)$. In case of particle filter this point is the center of the player body. In the following equations p_h refers to either the head or the body position. This can be calculated according to Equation 7.4 as follows:

$$\begin{aligned} x_h &= (\grave{x}_h - \grave{x}_c)\frac{r}{\grave{r}} + x_c \\ y_h &= (\grave{y}_h - \grave{y}_c)\frac{r}{\grave{r}} + y_c \end{aligned} \tag{7.6}$$

Looking at the triangle in Figure (7.8) the following equations can be derived using elementary geometry:

$$\frac{h_c}{h_p} = \frac{(x_h - x_c)}{(x_h - x_c) - (x_f - x_c)} = \frac{(y_h - y_c)}{(y_h - y_c) - (y_f - y_c)} \tag{7.7}$$

where h_p is the player height and the point $p_f = (x_f, y_f)$ is the feet position. In order to calculate the feet position Equation (7.7) should be solved for x_f which results in:

$$\begin{aligned} x_f &= x_h(1 - \frac{h_p}{h_c}) + x_c\frac{h_p}{h_c} \\ y_f &= y_h(1 - \frac{h_p}{h_c}) + y_c\frac{h_p}{h_c} \end{aligned} \tag{7.8}$$

and substituting for x_h from equation (7.1.2)

$$\begin{aligned} x_f &= (\grave{x} - \grave{x}_c)\frac{r}{\grave{r}}(1 - \frac{h_p}{h_c}) + x_c, \\ y_f &= (\grave{y} - \grave{y}_c)\frac{r}{\grave{r}}(1 - \frac{h_p}{h_c}) + y_c \end{aligned} \tag{7.9}$$

Finally, to get the meter coordinates the the point (x_f, x_f) should be divided

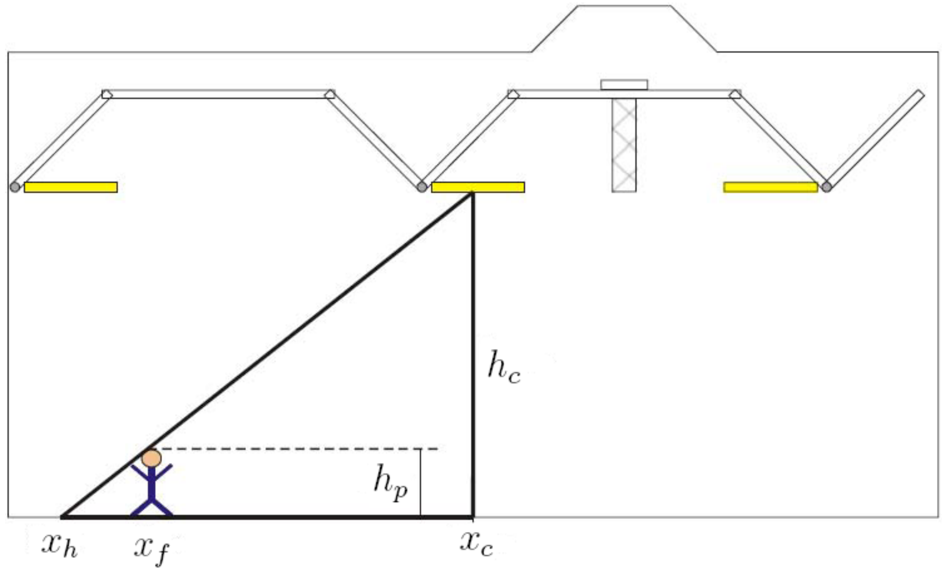


Figure 7.8: Calculating real world coordinates of players (own publication [115]).

by the resolution of the camera as follows:

$$\begin{aligned} x_r &= x_f/R_c, \\ y_r &= x_f/R_c \end{aligned} \tag{7.10}$$

where R_c is the relative resolution of the camera which has a unit of (pixel/meter). R_c gives how many pixels can represent one meter. It is calculated based on the camera height, camera sensor size and sensor pixel size.

Fusion Between two Camera Views

When the players move from one half of the playing field to the other, they are actually moving from one camera view to the other, so the tracker of each player should be transferred to the other view correctly. This is not a trivial task taking into consideration that the tracking is done on the fisheye distorted image. In order to deal with this problem an overlap between the two camera views is used in which the player in one view near to the middle line also appears in the other camera view. In the following two methods for the tracker transfer between the two camera views will be presented.

The first method for tracker transfer is based on testing the real world coordinates of the player to check if he is over the middle line of the playing field. In case the player is on the middle line, the distance between his image in one view to the axis that separates the two views of the merged frame (the big image formed from the two cameras) will be equal to the distance between his image on the other view to the same axis. In Figure 7.9 an example of a player crossing the middle line is shown, where the middle line of the field and the axis separating the two views are highlighted by red and blue color, respectively. The distance between the tracker and the axis is shown as a white line and the distance between the position where it should be transfered is shown as dashed white line. When the player is over the middle line the two distances are equal, so the tracker can be transfered by performing mirror transformation operation on the axis as shown in the figure. One pre-assumption to this method is that the camera orientation errors described in Section 7.1.1 should be corrected for each image. This will impose additional high computational cost in each tracking iteration because each pixel in the two camera images should be corrected in each iteration. In case of high quality cameras as the one used in this work, this means matrix multiplication and memory copy of about 2 million pixels (each camera is about 1 MB resolution). If LUT are used, the cost will be in lookup operation and memory copy which will also be an overhead on the whole tracking process.

The second method for the fusion between the two camera views also depends on testing real world coordinates of the players to see if the player is on the middle line. If the player is over the middle line, the position of the tracker on the other view is calculated by doing the inverse transformation from real coordinates in meters to the image coordinates in pixels. Figure 7.10 shows the steps for tracker transfer. In the figure the fisheye undistortion is referred to as *Defishing*, the correction of the camera installation errors is referred to as *Warping* and the inverse of it is *Unwarping*. Steps from 1 to 4 are done on the left side to find the position of the foot in meters, steps 5 to 8 are used to find the tracker position by doing the inverse of steps from

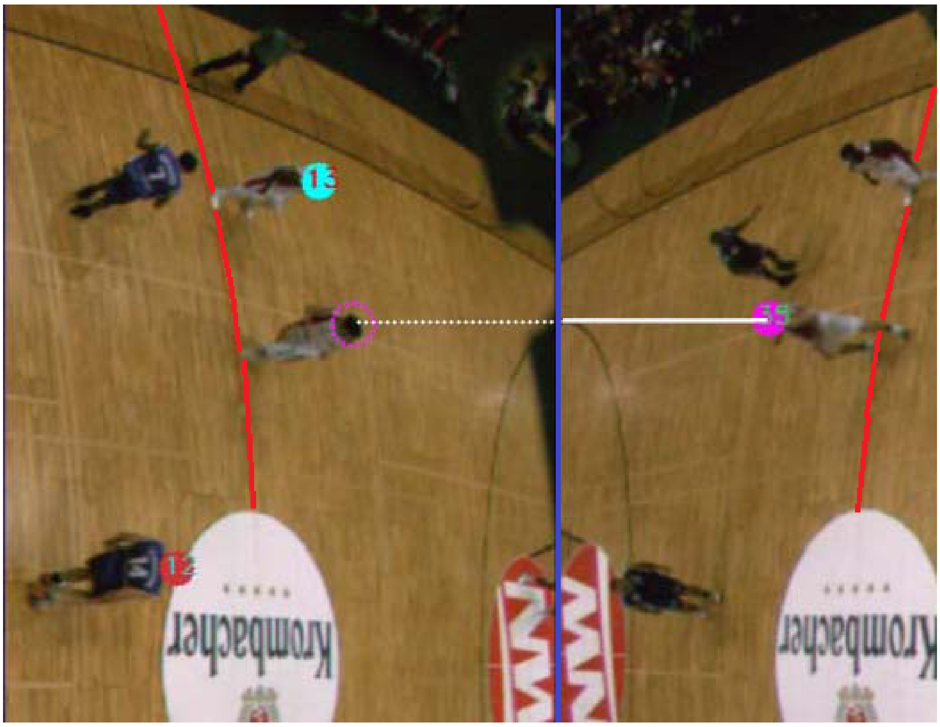


Figure 7.9: Fusion between two camera views.

1 to 4. As long as the foot position is not on the middle line only the steps from 1 to 4 are used.

Player Height Factor Analysis

The height of the player is an important parameter in the calculation of the player feet position in meters as seen above. The successful transfer of the tracker from one view to the other is highly dependent on the accuracy of the feet position. Template matching tracking uses the head of the player. Because the camera perspective differs with the player position on the playing field h_p should not be taken as the full height of the player. In particle filter tracking algorithm, the body of the player is tracked, so the h_p should also not be taken as the full height of the player. It should be taken as some point on the body. To conclude, the player height h_p should be a factor from the full player height. Therefore, it should be carefully chosen for each tracking algorithm using the overhead camera setup.

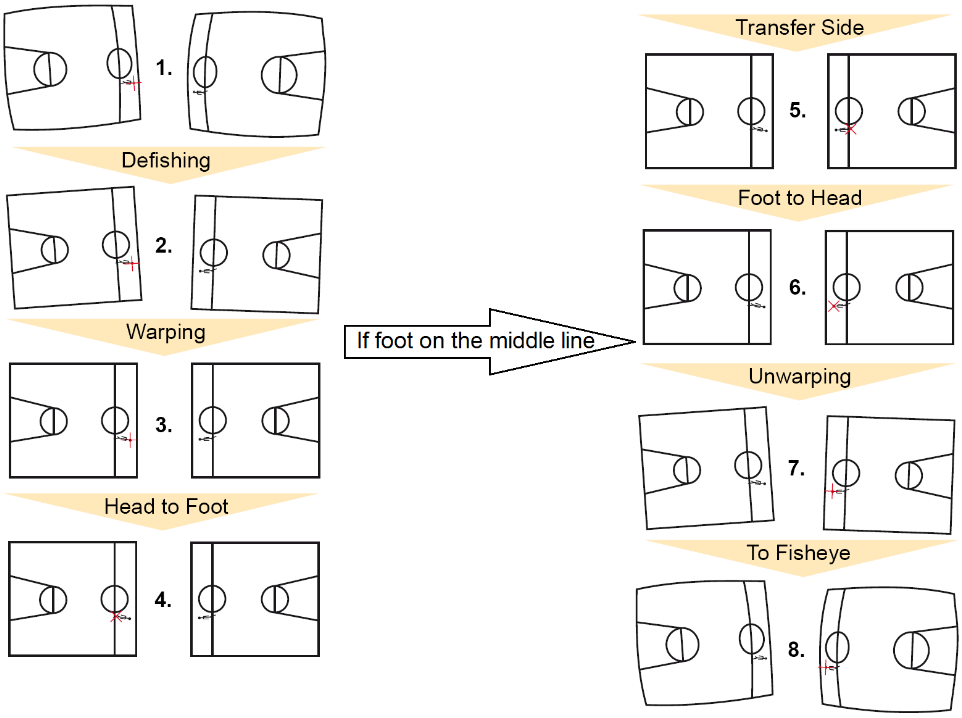


Figure 7.10: Tracker transfer between two camera views [82].

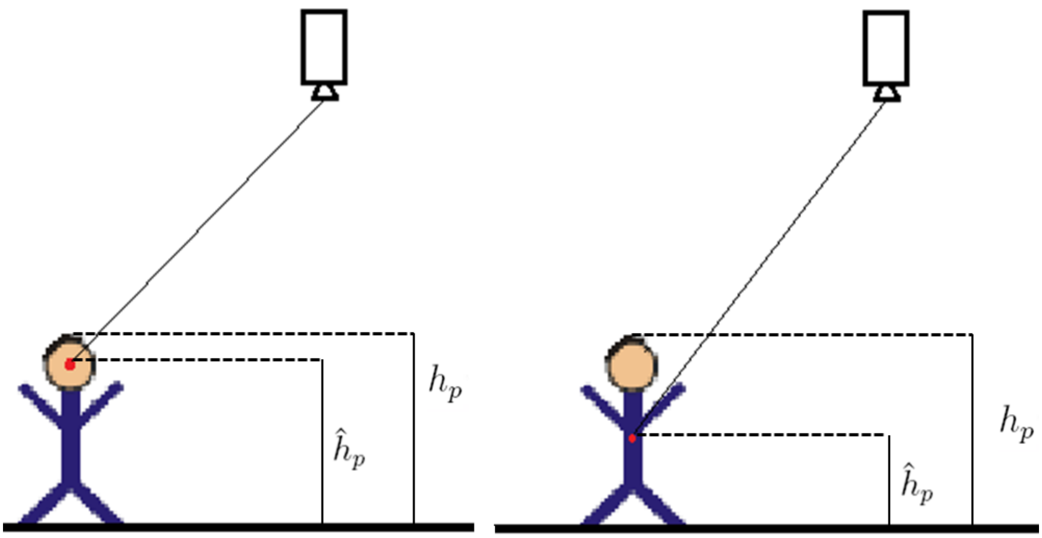


Figure 7.11: The difference between the actual height of the player and the height used for the calculations of feet position. Template matching on the left and particle Filter on the right.

Figure 7.11 shows the difference between the actual height of the player h_p and the height that should be used for the calculations of the feet position $\hat{h}_p$. In the left part of the figure, the red dot on the player's head shows the position of the best matching in case of template matching algorithm. On the right part of the figure the tracked position (also marked as red dot) using particle filter tracking is shown. The relation between h_p and $\hat{h}_p$ can be represented by the following equation:

$$\hat{h}_p = H h_p \tag{7.11}$$

where H is the factor that determines the amount that should be taken from the player's height when calculating the feet position which will be called *Height Factor*.

In order to find the best height factor the benchmark dataset described in Section 7.2.2 has been used. The RMSE (Root Mean Square Error) between the manually tracked feet positions and the feet position that result from the tracking have been computed. The RMSE is a measure of the differences between values predicted by a model or an estimator and the values actually observed from the object being modeled or estimated. It is calculated as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{N}^{i=1} |d_i|^2} \tag{7.12}$$

where d is the distance in meters between the player's manually tracked feet position and the player's feet position produced from the tracking. Figures 7.12 and 7.13 show the RMSE computed using the benchmark dataset for template matching and particle filter tracking, respectively. In the two figures the RMSE is computed for 9 players and for 20 different values of the height factor in the range of .05 to 1.0.

In Figure 7.12 it can be seen that the best value for the height factor which gives the minimum RMSE for most the players is between 0.75 and 0.85. The value 0.8 of height factor gives the minimum RMSE for 5 players while each of the values 0.75 and 0.85 gives minimum RMSE for 2 players.

Applying the same analysis on Figure 7.13 the range of effective height factor is between 0.55 and 0.7. The value of 0.55 gives the minimum RMSE for 5 players. Each of the values 0.6 and 0.65 gives one minimum and 0.7 gives the minimum for 2 players.

7.1.3 Implementation and Running Times of Pre- and Post- Processing

Image pre-processing has been implemented using the open source Computer Vision Library OpenCV[1] from Intel. Figure 7.14 shows the steps of the image pre-processing.

Making use of the closed world assumptions described in Chapter 5 the processing is optimized as shown in Figure 7.15 where the processing is done only on the regions where the trackers are at the current frame and making use of the possibility to predict the tracker position in the next frame as described in Section 3.2.4. Figure 7.15 shows how the white balance for example can be optimized to be done only on the tracker positions. This kind of optimization can be done on the background subtraction algorithm.

When the trackers are near to each other it may be better to compute one Region Of Interest (ROI) for all trackers. In each iteration of the tracking the processing ROI is computed for each tracker or for all trackers based on the distance matrix between all the trackers. An example of this can be seen in Figure 7.15 where on the left hand side the trackers are far enough away from each other so that each has its own ROI. On the right the trackers are nearer so that it's optimal to have one ROI for all trackers.

The processing time of one frame without using CWA optimizations is on average $67ms$ and with optimizations it goes down to $16ms$. So making use of the CWA assumptions makes the pre-processing 4.18 times faster.

7.2 Description of Test Datasets

In this section the video datasets used for testing of the SPA system and the its tracking algorithms will be introduced. The datasets include videos

[1] http://opencv.willowgarage.com

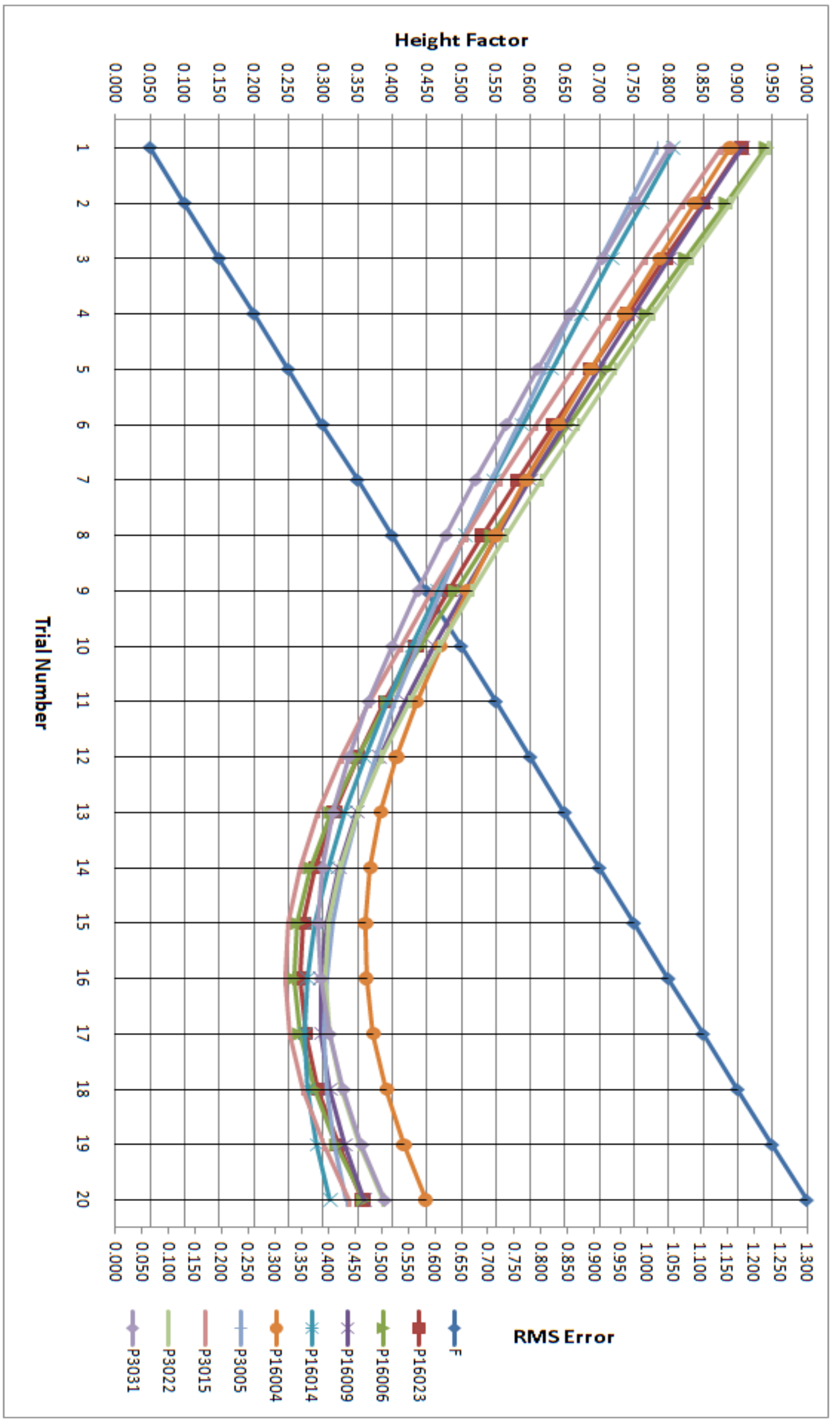


Figure 7.12: Height factor analysis for the template matching tracking

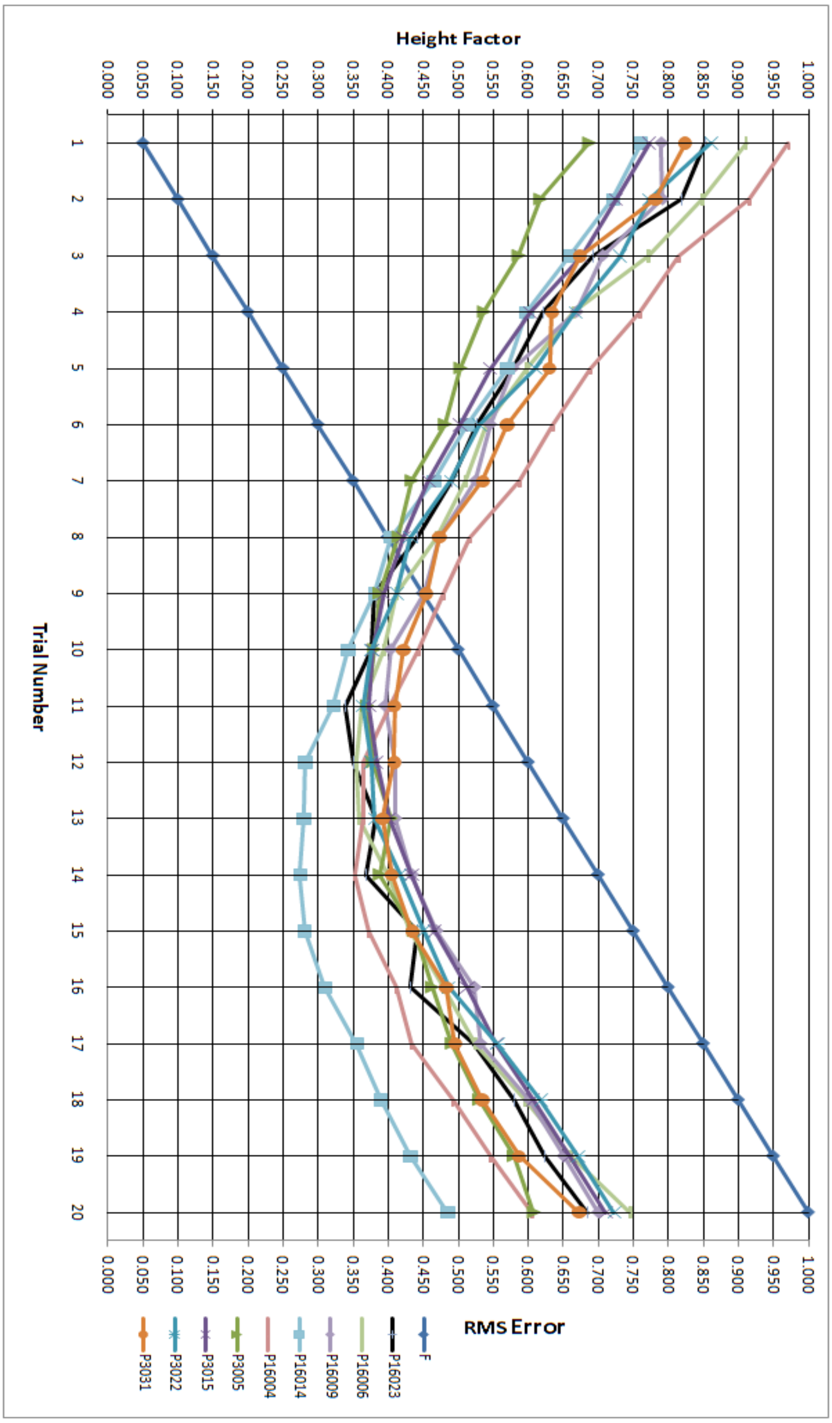


Figure 7.13: Height factor analysis for the particle filter tracking.

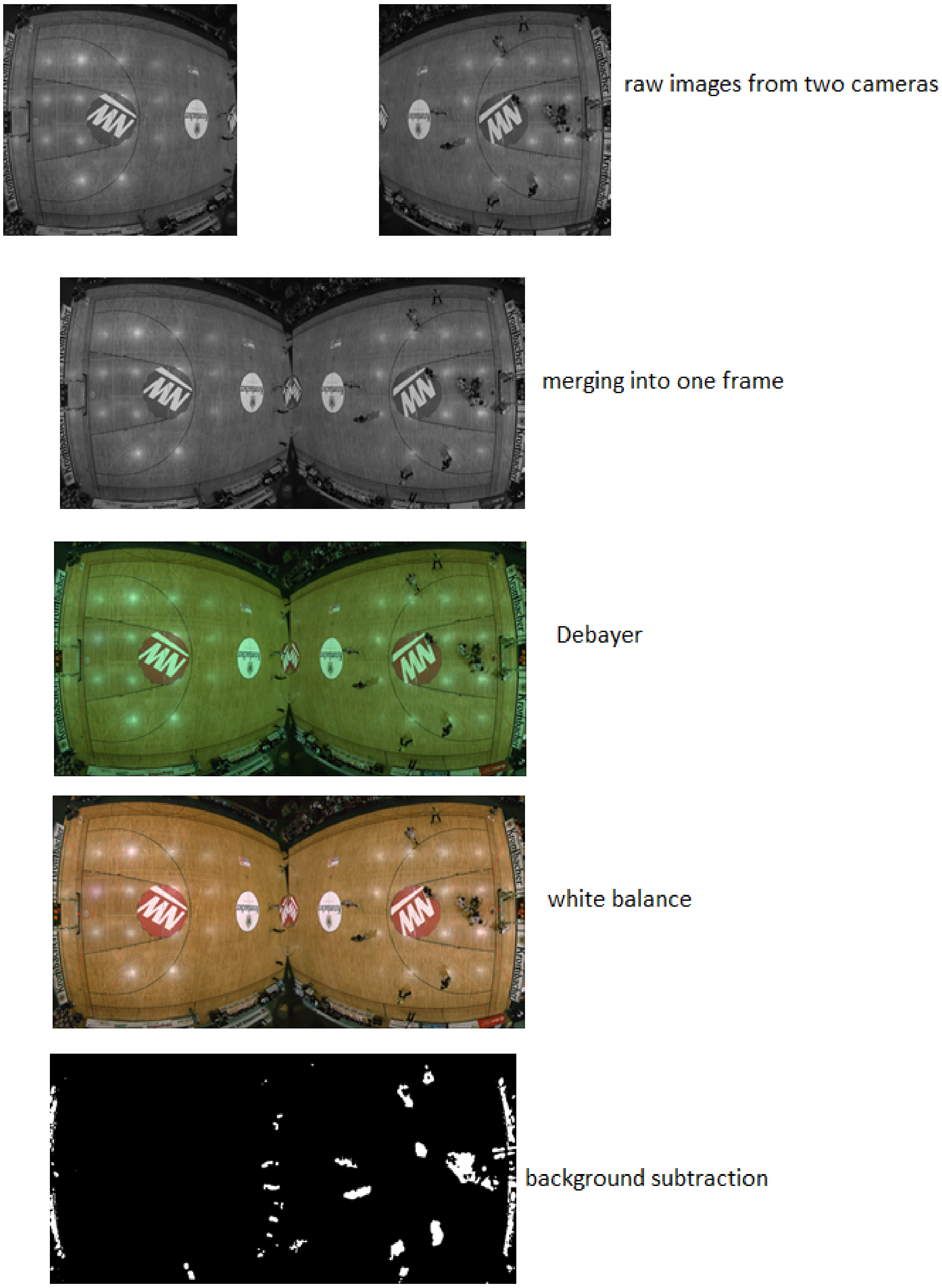


Figure 7.14: Steps of image pre-processing.

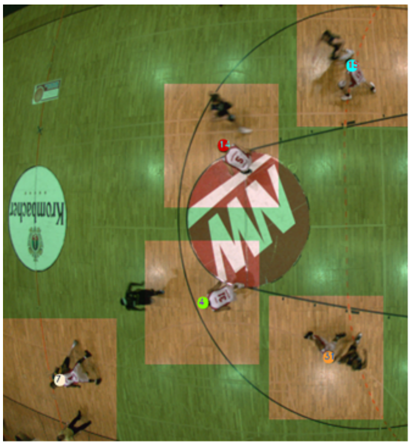
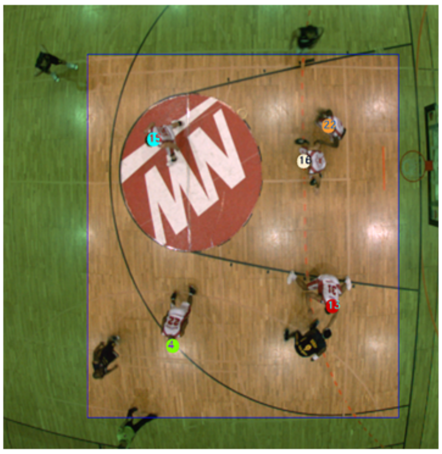



Figure 7.15: Image pre-processing optimization.

recorded from own camera installations in different sports halls, a benchmark dataset taken from own recorded data, video dataset from handball world cup taken with overhead analog cameras and own recorded video datasets that have been used for the evaluation of the SPA system quality criteria (see Section 7.7.1).

7.2.1 Own Recorded Datasets

The camera system shown in Section 7.1 is installed in three sports halls in the City of Paderborn and near by. The first installation took place in the University of Paderborn sports hall where recordings of students handball or basketball trainings can be recorded. The second installation was in the Maspernhalle sports hall in Paderborn where games from German first league basketball team Paderborn Baskets can be recorded. The third installation is in the sports hall Bördehalle in Soest where the games of the handball team Soester TV can be recorded.

7.2.2 Own Benchmark Datasets

In order to measure the accuracy of the tracking a benchmark dataset has been produced. The software developed during this work has the feature of allowing manual tracking. This means correction takes place in almost

Figure 7.16: Example of manual tracking of head, body and feet positions.

every frame. The produced trajectories from manual tracking are used as a ground truth for further experiments. Although this takes more time, a correct tracking will be generated. The generated trajectories should not be as long as a complete game or even quarter. A few minutes will be enough under the condition that it will include different situations in the game such as fast attack, play making, sprints and so on. The benchmark data has been taken from a game of basketball recorded in Maspernhalle and includes 2444 frames. Head, body and feet positions have been manually tracked in each frame for each player in the two teams.

7.2.3 Handball World Cup Dataset

This video dataset is a recording of a handball game of the World Cup 2007 between Germany and Slovenia. It has been recorded by two overhead analog cameras with resolution of 720×576 and a frame rate of 25 fps. The video data is divided into two parts one for each half. The first half has 59997 frames and the second has 62197 frames. The video has been digitalized, converted to SPA video format and used for tracking [86].

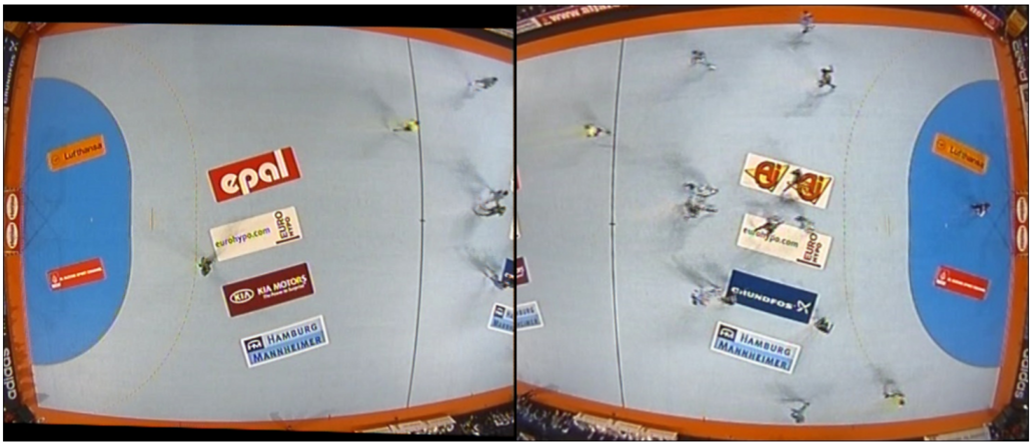


Figure 7.17: Screen shot from the handball World Cup game between Germany and Slovenia, taken from two overhead cameras and merged to form one image [86].

7.2.4 Test Quality Criteria Datasets

These are datasets used to test the quality criteria of the tracking system. The datasets consist of video recordings for subjects (test persons) who are instructed to move in specific patters and with different speeds. The covered distance and the speed of the subjects are known so they can be compared with the output of the SPA tracking. The tests are explained in details in Section 7.7.1.

7.3 Performance Measurements and Quality Criteria of The Tracking System

The quality of the algorithms is usually measured based on processing time complexity and memory requirements. In the evaluation of complex systems such as the SPA tracking system other quality criteria regarding the verification of the validity of the system are required. The tracking algorithm may fail to find the correct position of the target which is considered as an error in tracking. So measures such as the number of errors or the rate of error occurrence should be used to evaluate the tracking algorithms.

7.3.1 Performance Measurements

Errors from the tracking system occur when players move their extremities or if they move vertically (for example when jumping). The tracking algorithm tries to track part of the player's body or the whole body. For example, the template matching tracking implemented in this work is a head based tracking. The tracker may fail to match the head exactly. This happens due to collisions between players which may lead to partial or complete occlusion of the tracked part. Another source of error is the inaccurate estimation of the background mask. This may cause that the tracked part of the player is considered as background and is masked out from the matching results.

In order to keep error free tracking manual intervention (corrections) from user is required. The number of manual interventions (corrections) is a measure of the quality the tracker. The rate of correction is the number of corrections done in a specific amount of time (for example in one minute). In the evaluation of tracking algorithms done in this work the number of corrections done per minutes is used. This error measurement will be called E. The number of errors during the whole video sequence (basketball game quarter or handball game half) will be denoted as N_e.

User intervention in tracking process affects the rate of frame processing. The time of error correction is important for that estimation of the total time of processing one quarter or one game. It depends on the graphical user interface of the SPA and how easy it is for the user to correct an error. To correct an error the user should pause the tracking, may go back in the video for several frames until the occurrence of the error, correct by dragging the tracker to the correct position of the tracker and then let the tracking run normally. So another important measure of tracking is the time needed to correct one tracking error T_e.

To make it possible to evaluate the tracking algorithms a benchmark or ground truth dataset is needed. Using the manual tracking capability in the SPA software, the benchmark dataset described in Section 7.2.2 has been created.

7.3.2 Definition of Quality Criteria

One main use of the video tracking in the sports domain is to evaluate the performance of the players by coaches or by scientists in the coaching and sport science domain. According to Bös [13] the main quality criteria in the classic test theory are the *Objectivity*, *Reliability* and *Validity*. Objectivity is the degree in which the test results are independent of the investigator (examiner). A test is completely objective when it gives the same results by different investigators with the same subjects (test persons). Bös [13] based on Calrke [23] has used a correlation coefficient as a measure of objectivity and called it *Objectivity Coefficient*.

Reliability is defined by Lienert et. al. [64] as "The degree of reliability is a reliability coefficient which indicates to what extent under the same conditions and using the same subjects the obtained results of the test will be the same. In other words, in what extent the test results can be reproduced" Lienert define the validity of the test as the degree of accuracy with which the test has the one feature that it claims to measure, even actually measures. A test is totally valid if the produced result is exactly what occurred in reality. If it is applied to the tracking, a tracking system would be valid if the system produced exactly the paths that the subjects have really ran.

7.3.3 Definition of Evaluation Hypotheses

In order for the tests of validity, reliability and accuracy to be performed, the following hypotheses have been set:

- Hypothesis to prove the validity of the system:
  - H1 The system produces the exact positions of the player on the playing field.
    - H1.1 Player positions on the field are correct when the players are not moving (still-stand)
    - H1.2 Player positions on the field are correct when the players move on position.

- H.2 The system determines the exact running paths and velocity gradients.
    * H2.1 There is no distance produced when players still-stand on position.
    * H2.2 There is no distance produced when players make sport-specific actions on position without moving.
    * H2.3 The exact distances and speeds are computed when players make shuttle-run longitudinal.
    * H2.4 The exact distances and speeds are computed when players make shuttle-run transverse.
    * H2.5 The exact distances and speeds are computed when players make quadratic-run.
    * H2.6 The exact distances and speeds are computed when players make circular-run.
    * H2.7 The exact distances are computed when players make sprint with maximum speed.
    * H2.8 The exact speeds are computed when players make sprint with maximum speed.
    * H2.9 The exact distances and paths are computed when players make zick-zack run longitudinal.
    * H2.10 The exact distances and paths are computed when players make zick-zack run transverse.

- H3 Hypothesis to prove the reliability:
    - H3.1 The system produces the same distance when repeating the same tests of square-run.
    - H3.2 The system produces the same distance when repeating the same tests of circular-run.

- H4 Hypothesis to prove the objectivity: Two independent investigators will get the same results when evaluating the same data sets.

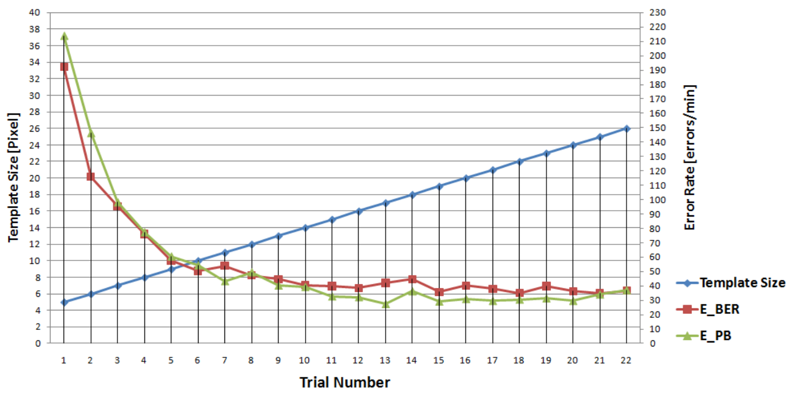


Figure 7.18: Template size analysis for the template matching tracking.

7.4 Results of Template Matching-based Tracking

The Template Matching-based tracking described in Chapter 3 has been implemented and used as a tracking module in SPA software. In this section the evaluation of the tracking is presented based on the performance measurements described above.

7.4.1 Parameters of Tracking Algorithms

Template Size

A fixed size template is used to track the head of the player and part of the shoulders. In order to find the best template size an experiment has been done to evaluate the tracking performance regarding the error rate E.

Figure 7.18 shows the results of the template matching for the two teams in the benchmark data using different template sizes. Template size around 20×20 (in pixels) gives lower number of errors and when template size is larger, the number of errors is not better. The reason for that is that the size of the player head can be sufficiently represented by 20×20 template.

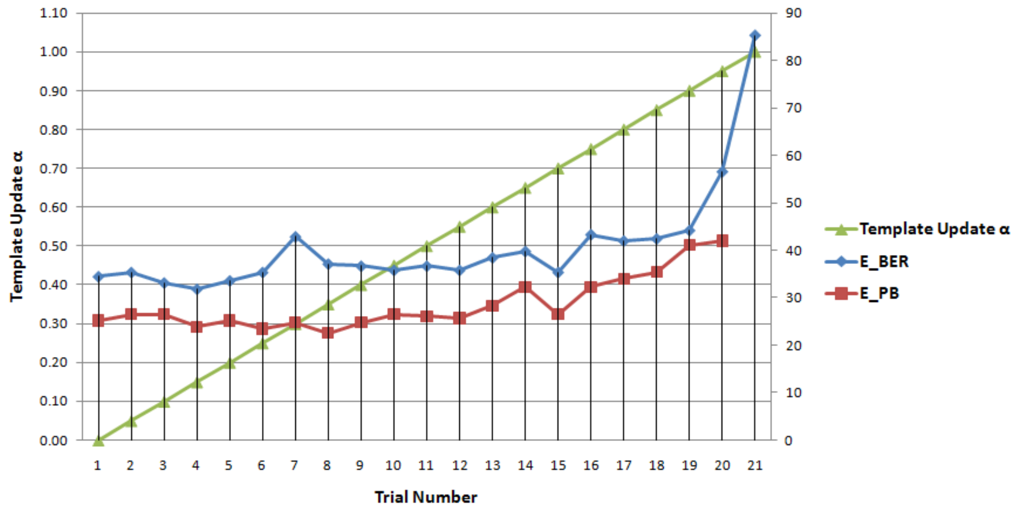


Figure 7.19: Analysis of the template update parameter.

Template Update

Figure 7.19 shows the analysis of update parameter α which is used in Equation 3.10 to determine the amount of update of the template. The figure shows the error rate for several runs of the template matching using different values of α. Values of α between 0.1 and 0.2 give the lowest values of error rate E.

7.4.2 Experimental Results

Template matching has been used to track most of the recorded videos from basketball games. Table 7.2 shows the results of the tracking of one team in 40 basketball quarters using template matching.

For each tracked player in the quarter each correction event done by the user is recorded as well as the time needed for correction. Statistics regarding the correction rate E and the correction time T_e have been done. Table 7.2 shows the correction rate E for all players and the average correction time T_e. The average value of E is 32.7 ± 11.6 with a minimum and maximum of 15.7 and 70.9, respectively. The average value of T_e is 2.87 ± 1.07 with a minimum and maximum value of 1.3 and 6.013 respectively.

Quarter	E	T_e	Quarter	E	T_e
1	70.86	4.88	21	21.60	2.66
2	23.29	2.80	22	35.03	2.96
3	17.90	3.49	23	27.87	3.57
4	21.16	3.73	24	42.10	1.52
5	16.00	2.93	25	30.89	1.65
6	22.25	3.40	26	21.65	1.55
7	15.73	2.66	27	36.22	5.34
8	24.33	2.89	28	43.35	2.58
9	19.25	3.01	29	43.79	2.70
10	43.83	2.15	30	38.29	1.89
11	48.40	2.47	31	26.62	1.89
12	60.19	1.88	32	34.25	2.16
13	34.76	4.43	33	30.90	2.11
14	42.50	3.88	34	35.81	2.67
15	26.17	3.20	35	32.19	3.18
16	38.29	1.70	36	38.44	4.56
17	27.69	1.32	37	33.58	2.31
18	39.23	1.62	38	19.23	3.08
19	21.26	1.79	39	39.24	6.01
20	40.32	2.61	40	23.87	3.66

Table 7.2: Template matching results for tracking of 40 basketball quarters.

Player	Playing Time(Min)	N_e	E	T_e
1	33.33	294	8.32	1.19
2	33.33	239	7.17	1.71
4	32.23	260	8.06	1.22
8	33.33	183	5.49	1.8
11	7.39	57	7.71	1.05
13	33.33	326	9.78	1.31
14	33.33	112	3.36	1.22
15	33.33	299	8.97	1.04
19	33.33	146	4.38	1.23
41	19.33	158	8.17	0.96

Table 7.3: Results of tracking the first half of the handball game between Germany and Slovinia in the world cup of 2006.

Another video dataset has been used to test the template matching tracking. It is the first half of the handball game between Germany and Slovenia. The result of the tracking is shown in Table 7.3. First column shows the player number. The second column shows the playing time for each player in minutes. The third column shows the number of tracker corrections N_e for each player during his playing time. The fourth column shows the rate of error correction E per minute for each tracker (player). The last column gives the average time T_e, in seconds, needed to correct the tracker. The error rate E for all the players in the whole half of the game is 35.48 error/min and average time for correction is 1.2877 seconds.

7.5 Results of Particle Filter-based Tracking

7.5.1 Parameters of Tracking Algorithms

Number of Particles

In order to find the optimal number of particles, an experiment has been done to measure the rate of errors against the number of particles. The tracking has been done for one team of the benchmark data with different number of particles from 2 particles to 50 particles with increment of 2. Figure 7.20

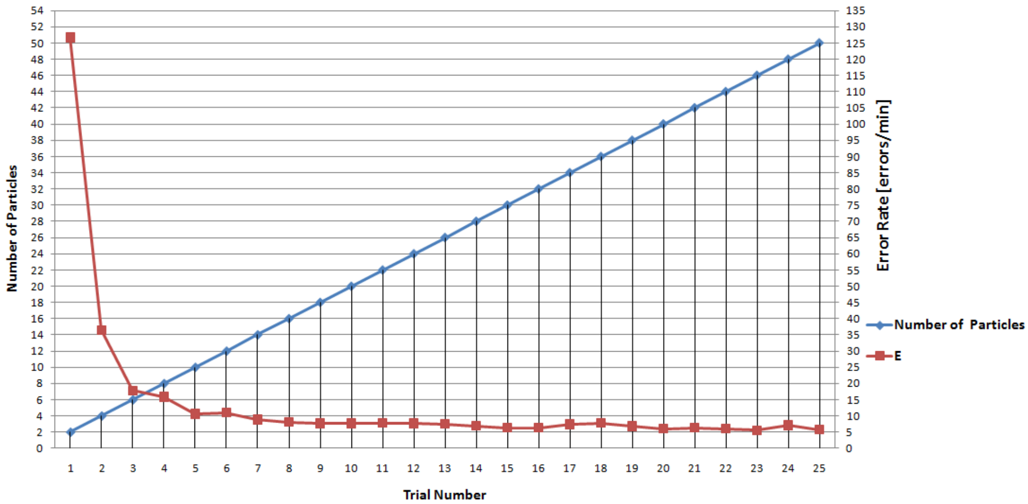


Figure 7.20: Analysis of the number of particles used in the particle filter tracking.

shows the results of the experiment. As the number of particles increases the rate of error E decreases. Starting from 20 particles the number of errors is almost constant. The minimum value of of E is 5.6 error/min comes with 46 particles. There are several minimum values of the error rate where the first minimum is at 30 particles with error rate of 6.33 error/min. So the suggested number of particles is 30.

Number of Color Bins

Another important parameter of the particle filter is the number of histogram bins for each color channel. An experiment using the benchmark data has been done in order to find the suitable number of histogram bins. Figure 7.21 shows the results of the experiment where the tracking is done on the benchmark data for different values of the number of histogram bins, from 2 to 25 with a step of 1. It should be noted that the number of bins shown in the figure means the number of color bins for each color channel. 10 bins, for example, means that for each of the three color components (RGB) there are 10 color bins which gives in total $10 \times 10 \times 10$ 3D histogram bins. It can be seen that starting from 12 bins the error rate is almost constant which means that a 3D histogram of $12 \times 12 \times 12$ is sufficiently enough for representing

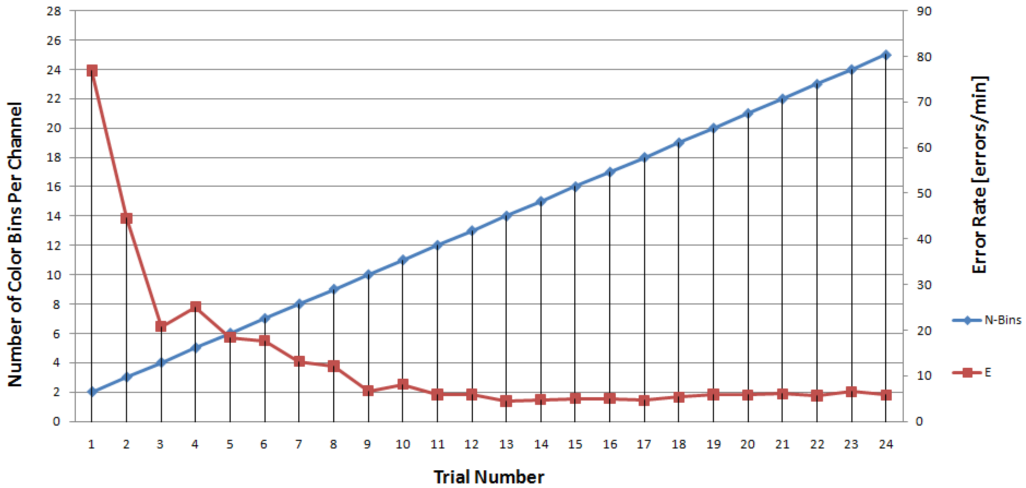


Figure 7.21: Analysis of the number of color bins for the 3D color histogram.

N_e	E	T_e
12	0.79	3.09
18	1.19	2.15
27	1.78	2.1
6	0.40	2.04
14	0.92	2.16

Table 7.4: Results of tracking a 15.16 minutes long basketball quarter using particle filter.

the color features of the player.

7.5.2 Experimental Results

Table 7.4 shows the results of tracking one team of a basketball quarter which is 15.16 minutes long. The tracking is done using Algorithm 5.2 with $10 \times 10 \times 10$ histogram bins and 30 particles for each tracker. For each of the 5 players the number of corrections is shown in the first column, the rate of error occurrence per minute is shown in the second column and the average time needed to correct one error is shown in the third column. The total number of corrections needed for the whole quarter is 77 which gives an error rate of 5.08 Errors/Minute for the whole quarter.

	Full Frame	CWA	CWA and Multicore
Pre-Processing	67	16	13
TM Tracking	45	38	38
PF Tracking	280	240	90
Total TM	112	54	51
Total PF	347	256	103

Table 7.5: Running times of template matching and particle filter tracking algorithms in millisecond.

7.6 Implementation and Running Times of Tracking Algorithms

Table 7.5 shows the running times of both template matching and particle filter tracking algorithms. The table also includes the running time of the pre-processing and the total time for the tracking algorithms including pre-processing. The presented running times are measured for the tracking of one team of basketball players which has 5 players. The table shows the differences in running times with and without optimizations. The optimizations include making use of closed world assumptions and also using the capabilities of multicore processors.

In Table 7.5 the first column shows the running time for processing of full frames. The second column shows the processing times based on CWA assumptions alone and the third shows them based on CWA and multicore optimizations. Using CWA in tracking is done through optimization of generating Voronoi masks where the masks are generated for the trackers which are closer to each other. In addition, the size of the mask is optimized based on the ROI as described in Section 7.1.3.

Using multicore optimizations has great effect on reducing the running times of particle filter tracking. The optimizations are used in computing the color histogram described in 4.2.3.

The frame rate of tracking using Template Matching and all possible optimizations is $19.6fps$ and for tracking using Particle Filters $9.7fps$.

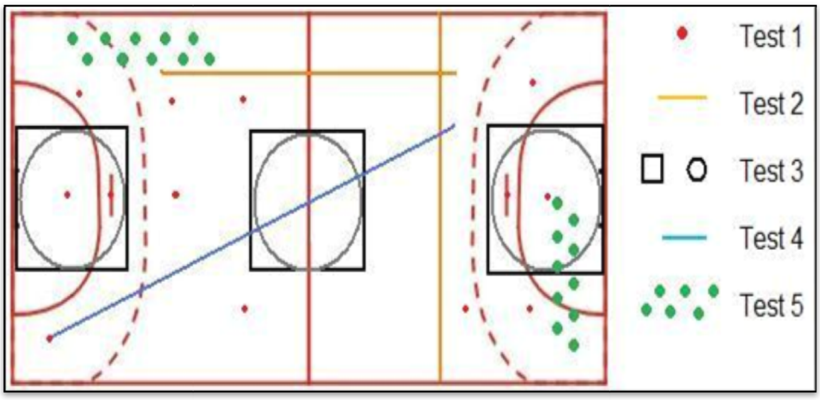


Figure 7.22: The patterns of movement and positioning of the quality criteria tests [81].

7.7 Evaluation of Quality Criteria

In order to prove the hypothesis defined in Section 7.3.3 a set of tests in which some of them are based on [84] has have been developed and run. The tests have been designed and implemented by Daniel Paier in his diploma thesis [81] which is based on the SPA system. Figure 7.22 shows the different test patterns. The tests have been run on a handball field in the sports hall of the University of Paderborn.

7.7.1 Description of Tests

Test 1: Determination of Position

This test is used to investigate the validity of the tracking which means to prove if the system produces the correct positions of the test persons (Hypotheses H.1 and H.2). It is also used to test if the movement of the upper part of the body and the extremities will affects the accuracy of the measured position. The test is divided into several tasks where the test persons are not moving from their positions so the total distance of the test persons should be 0m.

As shown in Table 7.6 there are 4 tasks in Test 1. In the first task (Task 1.a) the test persons stand still for a 60sec. In Task 1.b they should do hopping in the same position for 60sec. In Task 1.c they should stand still with movement of the arms to simulate a defensive action of handball player. In Task 1.d every two person couple is instructed to pass a handball without moving from position. The covered distance in all the tasks should be 0m.

Test 2: Shuttle Run

This test has been developed in order to report the performance of the player in the kind of motion where the players have to go and come back at almost the same line [62]. This movement pattern can also be used to simulate the performance of the basketball and handball players.

The test should run as follows, "the subjects run back and forth on a 20m course and must touch the 20m line; a sound signal is emitted from a prerecorded tape" [62].

The test has been changed in order to keep a constant speed. The investigations of Cambel [18] show that in about 80% of the distances covered by handball players the speed is about 2m/s. Thus the test is repeated in two differnent speeds, 2m/s and 3m/s. For the 2m/s test the test persons run back and forth for 10 times which gives a total distance of 400 meters. In the 3m/s they run 5 times which is equal to 200 meters.

The whole test is run in two different places of the playing field as seen in Figure 7.22. The first run of the test is called Test 2.a where the test persons should run in a transverse way (horizontal) and the second run is called Test 2.a where they run in a longitudal (vertical) way.

Test 3: Square and Circular Run

This test is based on the test developed by Perš and Kovačič in [84] in which the test persons have to run in square and circular paths in different parts of the field. Two identical positions on the two halves of the field have been chosen as seen on Figure 7.22. In addition a third position is in the middle so that it includes the middle line of the field. Being similar to Perš and Kovačič tests makes it easy to compare the two tracking systems. This test is aimed to test all the main quality criteria of the tracking, making it the most important one.

Test 4: 30 Meter Sprint

This is used to test how good the tracking system is with the maximum speeds of the players. The test persons are instructed to make a sprint (maximum speed run) in the path shown in Figure 7.22.

Test 5: Zick Zack Run

Because changing direction in handball and basketball occurs more often [61]. This test was developed to measure the validity of the tracking when there is a rapid change of direction in the players' movement.

7.7.2 Results of Quality Criteria Tests

In the following the results of the tests 1 to 5 will be presented. The results of these tests will be used to prove Hypothesis H1 to H4.

Test 1

In Table 7.7 the measured distances using SPA are shown. Although the subjects did not move in this test SPA has shown an average distance 0.3 ± 0.02 in the case of standing still, 7.34 ± 2.6 in the case of jumping, 5.6 ± 3.67 by arm movement and 4.8 ± 3.5 by ball pass between each two players. As described in the 7.1.2 the player height factor is an important parameter in the calculation of the position so in case of jumping longer distance is calculated.

Test 2

Test 2 is a Shuttle-Run test where the subjects should run 400m with a speed of 2m/s and again with a speed 3m/s. Table 7.8 gives a summary of the test results. The detailed numerical results are shown in Appendix A.1.

Task	**Duration**	**Description**
Test 1		
a	60 sec	still standing upright
b	60 sec	hopping on the position with one/two legs
c	60 sec	defense simulation with hand movement
d	60 sec	ball pass between two persons
Test 2		
2m/s transverse - 400m	3'30 min	Shuttle-Run
3m/s transverse - 200m	1'30 min	Shuttle-Run
2m/s longitudal - 400m	3'30 min	Shuttle-Run
2m/s longitudal - 400m	1'30 min	Shuttle-Run
Test 3		
Square		
1m/s	60 - 80 sec	Line run around the rectangle
1.5m/s	60 - 80 sec	Line run around the rectangle
2m/s	60 - 80 sec	Line run around the rectangle
Circular		
1m/s	60 - 80 sec	round run around the circle
1.5m/s	60 - 80 sec	round run around the circle
2m/s	60 - 80 sec	round run around the circle
Test 4		
30m - sprint	8 min	line run and record time
Test 5		
Zick-Zack transverse	8 min	run and record time
Zick-Zack longitudal	8 min	run and record time

Table 7.6: Description and durations of the quality criteria tests [81].

Subject	Stand[m]	Jump[m]	Arm move[m]	Pass[m]
1	0.06	5.24	4.79	6.63
2	0.31	8.39	8.65	2
3	0	6.13	13.25	2.6
4	0	13.63	10.45	7.56
5	0.86	7.85	1.5	4.87
6	0.19	8.33	2.06	12.69
7	0.03	3.84	2.86	1.51
8	0.03	9.28	5.44	9.61
9	1.62	8.88	6.27	2.75
10	0.27	7.41	4.66	5.36
11	0.51	5.3	2.52	3.72
12	0	7.08	1.89	0.97
13	0.02	4.02	8.41	2.19
AVG	0.30	7.34	5.60	4.80
STD	0.02	2.60	3.67	3.50
VAR	0.22	6.79	13.45	12.0

Table 7.7: Test1 measured distances in the case of still-stand, jump, arm move and ball passing.

Test 2	Distance[m]			Speed[m/s]		
	Avg	Std	Err[%]	Avg	Std	Err[%]
longitudinal 400m; 2m/s	405.09	4.41	1.27	2.02	0.02	0.023
longitudinal 200m; 3m/s	201.09	1.84	1.09	3.00	0.00	0
transverse 400m; 2m/s	391.35	15.51	-2.16	1.95	0.08	-2.39
transverse 200m; 3m/s	191.35	6.66	-4.32	2.88	0.10	-4.14

Table 7.8: Test 2 summary

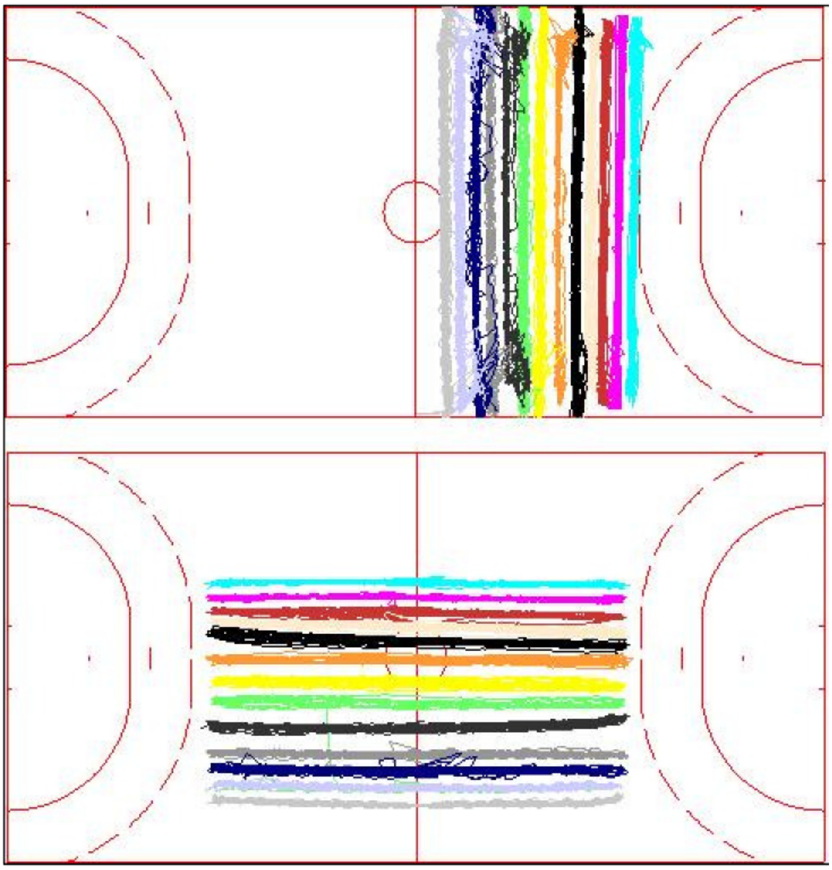


Figure 7.23: The trajectories of test 2; longitudinal (upper) and transverse (lower) [81].

Test 3

Test 3 is considered the most important one because it is used to prove the objectivity, reliability and validity of the SPA. For validity Table 7.9 shows a summary for the square and circular run in various speeds. For a speed of 1m/s the test persons should run 80m. SPA has produced an average distance of 78.86 with a standard deviation of 4.12 and average error in distance of -1.42m. The square run at 1.5m/s results in 3.75 rounds of the square which means 120m distance. SPA produced an average distance of 118.39m, a standard deviation of 9.83 and average error of -1.34m. With a speed of 2m/s the test persons could run 136m which results in average distance of 128.79m, standard deviation of 7.64 and average error of -5.3m. Figure 7.24 shows an example of the tracking paths acquired from SPA for both square and circular run with a speed of 2m/s.

In Table 7.9 the summary for the results of the circular run in three different speeds are shown. For a speed of 1m/s the test persons ran 3 rounds which means 75.6m. The result acquired from SPA shows an average distance of 74.35m and average error in distance of -1.65m and standard deviation of 3.87.

With a speed of 1.5m/s the test persons had to run 4 rounds resulting in a total distance of 100.8m. SPA produced an average distance of 96.96m, average error of -3.84m and standard deviation of 4.86. For 2m/s speed the run distance is 138.6m SPA gives an average distance of 129.78, average error of -3.84m and standard deviation of 7.87.

The detailed numerical results for each test person and for each phase of the test can be seen in Appendix A.2.

The square and circular run tests have been repeated in order to test the reliability of the SPA tracking. Table 7.10 shows a summary of the errors for the two runs with different speeds. The columns show the different speeds of 1m/s, 1.5m/s and 2m/s with sub-separation into two runs. For the second run SPA has an average error of -0.85 ± 3.08m for 1m/s speed. For 1.5m/s speed, the average error is -5.33 ± 5.79m and for 2m/s it gives average error of -8.36 ± 7.24.

Test3	Distance[m]			
	Avg	Std	Err	Err[%]
square 80m; 1m/s	78.86	4.12	-1.14	-1.42
square 120m; 1.5m/s	118.39	9.83	-1.61	-1.34
square 136m; 2m/s	128.79	7.64	-7.21	-5.30
circle 75.5m; 1m/s	74.35	3.87	-1.25	-1.65
circle 108.8m; 1.5m/s	96.96	4.86	-3.84	-3.81
circle 138.6m; 2m/s	129.78	7.87	-8.82	-6.46

Table 7.9: Summary of Test 3.

Test3 square repeated	err[m]		correlation
	avg	std	
Run 1; 1m/s	-1.14	4.12	0.94
Run 2; 1m/s	-0.85	3.08	
Run 1; 1.5m/s	-1.61	9.83	0.87
Run 2; 1.5m/s	-5.33	5.79	
Run 1; 2m/s	-7.64	7.64	0.97
Run 2; 2m/s	-8.36	7.24	

Table 7.10: Test 3 - summary of repetition of square run to test reliability.

Table 7.10 shows the correlation factor between the two runs for different speeds in the last column. The speed 1m/s run 1 and 2 result in a high correlation factor of 0.94. For a speed of 1.5m/s, the correlation is 0.87 and for 2m/s speed the correlation is 0.97 which is the highest. In total the correlation factor between the results of the two runs for the square run is 0.98.

For the circular run, Table 7.11 shows the results of the repetition of the test. For the speed 1m/s the average error was -0.85 ± 3.08 and the correlation between the two runs for the same speed was 0.89. For the speed 1.5m/s the average error was -5.33 ± 5.79 and the correlation was 0.84. For a speed of 2m/s the average error was 7.87 ± 7.24 and the correlation was 0.91. The total correlation for the two runs for all speeds was 0.99.

In order to test the objectivity of the SPA tracking, a second person has performed the tracking for the square run test. Table 7.12 shows the summary of the results for two different testers for different speeds. For a running speed

Test3 circle repeated	err[m]		correlation
	avg	std	
Run 1; 1m/s	-1.25	3.87	0.89
Run 2; 1m/s	-0.85	3.08	
Run 1; 1.5m/s	-3.84	4.86	0.84
Run 2; 1.5m/s	-5.33	5.79	
Run 1; 2m/s	-8.82	-8.36	0.91
Run 2; 2m/s	7.87	7.24	

Table 7.11: Test 3 summary of repetition of circular run to test reliability.

Test3 square repeated	err[m]		correlation
	avg	std	
Run 1; 1m/s	-1.14	4.12	0.81
Run 2; 1m/s	-2.09	4.02	
Run 1; 1.5m/s	-1.61	9.83	0.81
Run 2; 1.5m/s	-5.31	5.22	
Run 1; 2m/s	-7.64	7.64	0.93
Run 2; 2m/s	-13.30	9.86	

Table 7.12: Test 3 - summary of results of the square run test results from two different investigators.

of 1m/s the second examiner has an average error of -2.09 ± 4.02, for a speed of 1.5m/s the average error was -5.31 ± 5.22 and for the speed of 2m/s the average error was -13.30 ± 9.86. The correlation between the two examiners for the three different speeds was 0.81, 0.81 and 0.93 respectively.

Test 4

Test 4 resulted in a running diagonal distance of 30m. Figure 7.22 shows the trajectories of the 13 test persons. The black marks on the image are the position of the light gates that are used to measure the speed of the test persons. Table 7.13 shows the results from SPA and from the photo sensor. The SPA gives a distance between 27.18m and 29.11m. The average distance is 27.99 ± 0.5m. The average error is -2.01 ± 0.5m. The percentage of error is 6.96%. The average speed measured by SPA is 22.22 ± 1.76. The reference value acquired from the sensor gives an average speed of 23.44 ± 1.64 so the

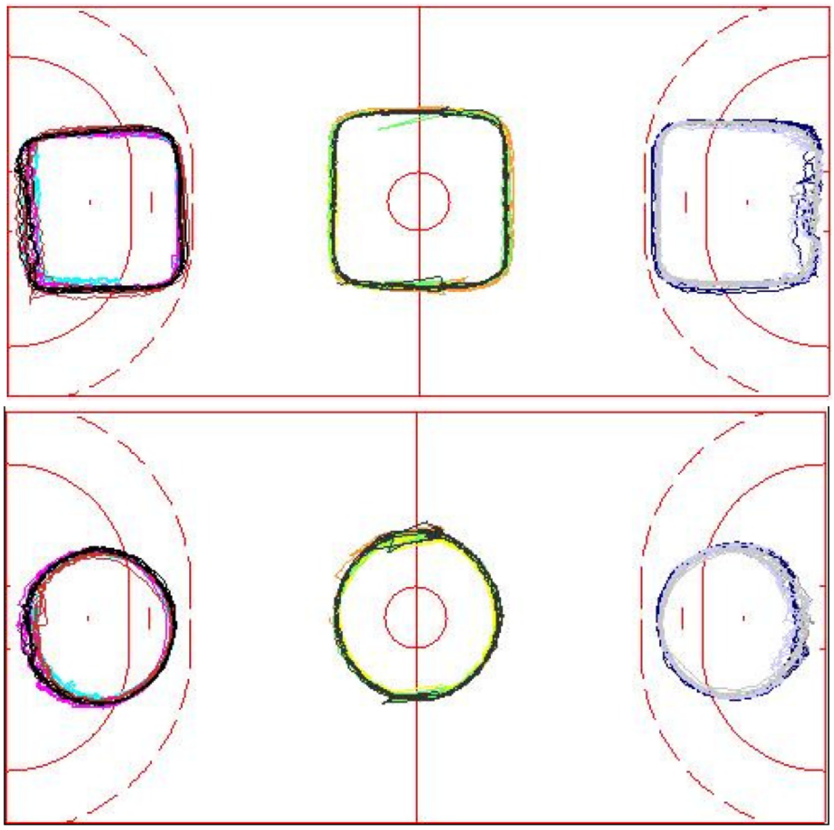


Figure 7.24: Test3 trajectories for square (upper) and circle (lower) run at speed of 2m/s [81].

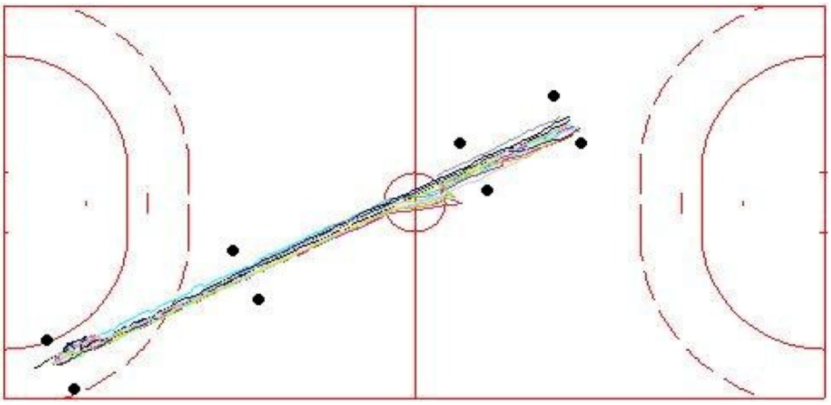


Figure 7.25: Trajectories of test 4, 30m sprint with light gates at distance of 0;10;25;30m. [81]

average error in speed is 1.22 ± 0.94 and the percentual error is 5.2%.

Test 5

The zick-zack running pattern is located in two different (longitudinal and transverse) locations of the playing field as shown in Figure 7.26 which shows also the trajectories of the 13 test persons. The total distance that should be run is 18m. In Table 7.14 the values for each test person are shown. The longitudinal trajectories are between 16.1m and 20.4m with an average distance of 17.98 ± 1.36m. The corresponding percentage error is -9%. The transversal zick-zack paths are between 19.1 and 20.8. In average the distance was 19.99 ± 0.52m and the percentage of error is 11.07%.

7.7.3 Evaluation of The Hypotheses

After showing the results of the tests, the evaluation of the hypothesis will be introduced in this section.

Hypothesis H1

Test 1 is designed to test Hypothesis H1. The hypothesis H1.1 is accepted according the following: By investigation of the x and y coordinates of the player positions acquired from SPA, a radius which determines the area where

subject	dist[m]	err[m]	avg speed SPA	avg speed sensor	err[m]
1	27.96	-2.04	22.71	24.55	1.84
2	28.19	-1.81	22.55	23.79	1.24
3	28.03	-1.97	22.1	23.48	1.38
4	27.72	-2.28	21.65	20.57	-1.08
5	29.11	-0.89	19.06	21.56	2.5
6	28.27	-1.73	23.67	24.66	0.99
7	27.18	-2.82	23.48	23.74	0.26
8	28.21	-1.79	21.76	23.13	1.37
9	28.22	-1.78	24.78	25.06	0.28
10	27.44	-2.56	20.3	22.13	1.83
11	27.34	-2.66	19.43	21.22	1.79
12	28.23	-1.77	23.63	25.12	1.49
13	27.95	-2.05	23.77	25.74	1.97
avg	27.99	-2.01	22.22	23.44	1.22
std	0.50	0.50	1.76	1.64	0.94

Table 7.13: Test 4 - distance and speed measured by SPA and photo gate sensor.

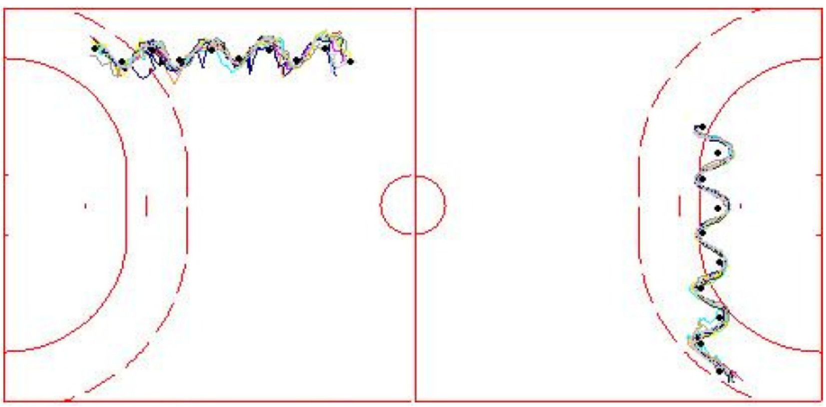


Figure 7.26: Trajectories of test 5 [81].

	longitudinal		transverse	
subject	dist[m]	err[m]	dist[m]	err[m]
1	16.1	-1.9	19.1	1.1
2	18.6	0.6	19.9	1.9
3	16.6	-1.4	20.5	2.5
4	16.5	-1.5	20	2
5	19.4	1.4	20.1	2.1
6	16.9	-1.1	20.6	2.6
7	19.6	1.6	20.1	2.1
8	18.1	0.1	20	2
9	17	-1	19.6	1.6
10	19.1	1.1	20.8	2.8
11	20.4	2.4	19.8	1.8
12	17.9	-0.1	19.1	1.1
13	17.6	-0.4	20.3	2.3
avg	17.98	-0.02	19.99	1.99
std	1.36	1.36	0.52	0.52

Table 7.14: Test 5 - longitudinal and transverse distance for zick-zack run.

the players are standing can be calculated. This radius can be calculated using the RMSE measure between the marked coordinate and the calculated coordinates from SPA. For the still-stand part of Test 1, the RMSE is 0.115m. The experiments of Perš and Kovačič [84] show a RMSE of 0.2m near to the center and 0.5m near to the playing field boundary (according to [84]) for a similar test. The 0.115m results in a diameter of 0.23m which is shorter than the shoulder length.

Hypothesis H1.2 is “The Player positions on the field are correct when the player moves on position”. This hypothesis is accepted according to the following: As in the previous sub-hypothesis, the RMSE of the error was calculated and resulted in 0.134m. Accordingly the diameter is 0.268m which is still smaller than the shoulder length. The tracking of Perš and Kovačič [84] has a RMSE of 0.3m near to the center and 0.6m near to the playing field boundary for the moving players.

Percentual Standard Deviation	Evaluation
$[\leq -15\%]$	very bad
$[-15\%; -10\%]$	bad
$[-10\%; -5\%]$	good
$[-5\%; 0\%]$	very good
$[0\%; 5\%]$	very good
$[5\%; 10\%]$	good
$[10\%; 15\%]$	bad
$[\geq 15\%]$	very bad

Table 7.15: The developed evaluation categories of the percentual measured standard deviation [20, 81].

Hypothesis H2

This hypothesis concerns the running distances and speeds for different movement patterns. In the previous tests the running distances and speeds of the players have been measured by SPA and compared to the real values that have been previously determined or measured by sensors. Carling et al. [20] uses a similar procedure where they compare the measured data, from the system under consideration, with the data that is previously measured by calibrated sensors. Also Perš and Kovačič [84] compare the values that are measured by their system with those which have been previously determined.

The evaluation of SPA tracking regarding Hypothesis H2 is done by categorization of the error percentage that has been used by Carling [20]. Calring has used this method to compare a football player tracking system measurements with the measurements of the GPS (Global Positioning System). Carling evaluates the results that bear less than a 7% standard deviation as accepted. According to Carling "these relatively small overestimations combined with an acceptable level of relative technical error of measurement both within and between trackers should not prevent the use of these technologies to monitor player movements" [20]. Since, in Carling's work, it can't be exactly differentiated between the technical errors and errors resulted from the experiment, the error categories have been adapted as shown in Table 7.15.

Hypothesis H2.1 is accepted according to the following: Table 7.7 second column shows the errors in the distance in case of still-stand. When computing the average error which is 0.22m/min over a complete game duration the error will be 13.2m for 60min, in case of handball. As stated in [72] the average distance of a player in a handball game is between 4700 and 5600. So the distance of 13.2m is marginal comparing to the whole distance in one game. The tracking in Perš and Kovačič [84] gives an average distance of 0.75m/min in stand-still case which shows that the SPA tracking is better.

Hypothesis H2.2 is accepted. Table 7.7 (third, fourth and fifth columns) shows the distances computed by SPA in case of 60 seconds jumping, arm movement and ball-pass, respectively. The average computed distance is 7.34m, 5.6m and 4.8m which is better than the result of Perš and Kovačič [84] which gives average of 7.5m for sport specific actions.

Hypotheses H2.3 and H2.4 are accepted. The results of test 2 regarding the shuttle run are shown in Table 7.8. The errors in distances are between −4.32% and +1.27. The errors in speeds are between −4.14% and 0.023%. The percentage of error in the distances and speeds are evaluated as very good accourding to Table 7.15.

Hypotheses H2.5 and H2.6 are accepted. The results of test 3 regarding the square and circular run with different speeds are shown in Table 7.9 The average errors are between −6.46% and −1.34% which are according to Table 7.15 are evaluated as very good and good.

Hypotheses H2.7 and H2.8 are accepted. The average errors in distance and speed are 6.96% and 5.2% respectively which are evaluated as good according to Table 7.15.

Hypothesis H2.9 is accepted. The percentual error in distance for the longitudinal zick-zack run is −9% which is evaluated as good according to Table 7.15.

Hypothesis H2.10 is not accepted. The error in distance for the transversal zick-zack run is 11.07% which is evaluated as bad according to Table 7.15.

Hypothesis H3

Hypothesis H3.1 is accepted. According to [79] "Correlation coefficient is an estimate of the consistency between two test occasions" and according to [38] "The closer the correlation to 1, the better the replication" so correlation coefficient is chosen as indication to the consistency of two tests. Table 7.10 gives the results of the repetition of square run. The correlation between the two runs of the tests for the different speeds is shown in the column titled "correlation". The values of the correlation between the two tests are 0.94,0.87 and 0.97 which are near to 1. This means a strong consistency between the result of the two tests.

Hypothesis H3.2 is accepted. The correlation coefficient between the two tests of the circular run are shown in Table 7.11. The values of the correlation are 0.89, 0.84 and 0.91 which are near to 1 show a strong consistency between the two tests.

Hypothesis H4

Hypothesis H4 is accepted. The correlation coefficient is used to evaluate the consistency between the evaluation of the tests by two different investigators. Table 7.12 shows the the values of the correlation coefficient between the results of the two tests. The values of the correlation are 0.81,0.81 and 0.93 which are high enough to prove the hypothesis.

7.8 Discussion

In this chapter an experimental study on the whole work of this thesis is presented. The chapter begins with the experimental setup in Section 7.1 including the installation of cameras, the pre-processing done on the image before it is delivered to the tracking algorithms and the post-processing of the position.

The pre-processing includes color space conversion as well as detection and subtraction of background. In the explanation of the post processing, the details of correcting the camera fisheye lens distortion, computing player feet position in meters and the fusion between two camera views have been presented. The hight factor parameter that has an effect on computing the feet positions in meters has been studied and determined through the benchmark data for both tracking algorithms. This analysis of hight factor is important for any tracking algorithm using the overhead camera setup. The running time of the implementation of the post-processing has been presented at the end of the experimental setup section. It has been shown that using CWA has great effect on reducing the running time of post-processing.

In Section 7.2 the different datasets that have been used in the testing of the tracking algorithms are presented. The datasets include own recorded data by the camera setup explained at the beginning of the chapter and other datasets recorded by different types of cameras and different image resolutions. The benchmark dataset that is used in the analysis of tracking algorithms' parameters has been created by SPA software using the manual tracking capability.

Section 7.3 shows the performance measurements as well as quality criteria of the tracking system. The performance measurements include the running times and rate of error occurrence in tracking. In order for the SPA tracking system to be used by sports scientists as well as trainers it should be tested for the quality criteria in the classical test theory. These criteria include the objectivity, reliability and validity of the tracking. Definitions of the quality criteria and the hypothesis about SPA have been presented.

The numerical results for the testing of both tracking algorithms and for determination of the best parameters are presented in Sections 7.4 and 7.5.

Section 7.6 presents the experimental results regarding the running times. The template matching algorithm is faster with more error rate in tracking while the particle filter is slower with less error rate.

Section 7.7 gives a description of the tests used for evaluating the hypotheses defined in Section 7.3.2. It also gives the evaluation of the hypotheses based on references from sports science and sports medicine domain. Some tests can be compared with the work of Perš and Kovačič [84].

Chapter 8

Summary, Conclusions and Future Work

This chapter presents a summary of the thesis in addition to concluding remarks and some guide lines for future work.

8.1 Summary and Conclusions

Chapter 2 presented an overview on the sport player tracking using computer vision. This review included presenting the player tracking in sports domain as a computer vision problem and the state of the art in the player tracking based on computer vision. The presented review is based on the excellent review of Kristan in his PhD [58]. The methods used in background detection and feature encoding subsections depends mainly on the quality of the images acquired from the camera. Although the work of Kristan [60, 58] and Perš [84, 47] tried to find a solution for tracking of indoor sports players, the acquisition of the video data is based on low resolution analog cameras of resolution 384×288 where the player is represented by 10×10 pixels. The low quality of the and resolution of the video is one source of errors in the tracking process as explained in [48]. The noise that comes from digitalization of analog images and the artifacts resulted from video compression [48] require sophisticated techniques for background detection and cause poor features for the tracked target. The tests done to prove the quality of the systems were not enough because the test sequences used were not long enough and do not contain all the player interactions in real games. It has been noted from the literature review that there is no benchmark dataset from real game situations that could be used to evaluate and compare the different tracking systems.

In Chapter 3, it has been shown how deterministic tracking can be used to track players using overhead cameras. As a start the simple blob tracking has been used. Tracking single player gives excellent results when the player is not interacting with others. When collision between players occurs two or more players come to one blob and after separation it is difficult to associate the paths again. The time complexity of the blob detection depends on the implementation of the edge detection. Canny edge detection [19] has been used in the implementation. After the edge detection the connected components in the “edges image” are inspected to find contours which is the classified based on the player size on the image. Speaking about the space complexity or the amount of memory required for blob detection, several temporary images and data structures are needed to store the result of the edge detection and contour detection. The second part of the Chapter 3 showed how template matching can be used for tracking sport players. Investing the high quality of the images it is possible to track the head of the player with small part of the shoulders. In order to determine the best size of the template, experiments have been done in Chapter 7. Tracking the head has the advantage that no rotation or scaling of the head is required. Although the overhead cameras reduces the chance that the head of the player is totally invisible due to occlusion, in some situations when the player falls down or other taller player comes into a strong contact with him the head disappears from the scene. The multiple player tracking framework described in Chapter 5 reduces the number of errors that occurs due to occlusion. In some cases the tracker still needs to be manually corrected. The rate of error occurrence is inspected in Chapter 7.

In Chapter 4 concerns the tracking of a single sports player using probabilistic model in order to deal with the non-linearity and the uncertainty in the sports domain. CONDENSATION algorithm which is based on particle filter techniques has been used in the CV domain for tracking. In order to use CONDENSATION for tracking in sports domain models of player (state), his motion, state transition and likelihood function should be found. The state of the player has been modeled as an ellipse and the visual features inside the ellipse are modeled as a three-dimensional color histogram. Three dimensional histograms have been chosen to preserve the spatial information of the RGB color tuples in the color space. The transition model of the player motion and the ellipse size is modeled as a normal distribution. The likelihood is modeled as the Bhattacharyya distance between the state histogram and the reference histogram.

In order for the tracking algorithm presented in this chapter to be used in practice some parameters including the number of particles and the number of histogram bins need to be determined. Chapter 7 shows the experiments done to find the best parameters as well as the experiments done to evaluate the tracking itself in terms of accuracy and running time.

In Chapter 5 the multi-player tracking in the context of closed worlds has been introduced. Kristan [60] introduced 5 closed world assumptions and used them in the tracking of multiple sports players by partitioning the search space into disjoint regions where each region contains only one tracker. Another 5 closed world assumptions have been introduced in this chapter and used to enhance the multiple player tracking as well as the image pre-processing. Clustering of position data from basketball game has been used to show that only in 16.6% of the frames partioning of space to more than three partitions is needed and in 25% only a simple partioning for two trackers is needed. In Chapter 7 experiments have been done to show the effect of using closed world assumptions in tracking accuracy and effiecieny.

In Chapter 6 a brief overview of the SPA softwares which is developed during the course of this thesis work was introduced. SPA is also a part of the sport project at the System and Circuit Technology research group. An idea about the design of the software together with screen shots from different modes has been presented. The details of the software parts that are not directly related to the domain of this thesis can be found in the appropriate references referred to in the thesis. SPA is a tool that enable users to acquire a new combination of data about the sports players. The combination of physiological and position data provides new possibilities for analysis of indoor sport games. The acquisition, tracking, visualization and analysis of these combination of data is done in one system through a user friendly graphical user interface. SPA is being used for the analysis of official basketball games played in the Paderborn city and the handball games played in Soest city. It is also used for the analysis of training sessions of handball at the University of Paderborn. The analysis results is used by the department of Sport Medicine and the department of Applied Mathematics at the University of Paderborn for further research at the levels of Bachelor, Master and PhD in these departments.

In Chapter 7 an experimental study on the whole work of this thesis is presented. The chapter begins with the experimental setup including the installation of cameras and the pre-processing done on the image before it is delivered to the tracking algorithms and the post-processing of the position data in order to acquire the final positions on the playing field in meters. The pre-processing includes color space conversion and detection and subtraction of background. In the explanation of the post processing the details of correcting the camera fisheye lens distortion, computing player feet position in meters and the fusion between two camera views have been presented. The hight factor parameter that has an effect on computing the feet positions in meters has been studied and determined through the benchmark data for both tracking algorithm. This analysis of hight factor is important for any tracking algorithm using the overhead camera setup. The running time of the implementation of the post-processing has been presented at the end of the experimental setup section. It has been shown that using CWA has great effect on reducing the running time of post-processing.

The second part of the chapter presents the different datasets that have been used in the testing of the tracking algorithms. The datasets include own recorded data by the camera setup explained at the beginning of the chapter and other datasets recorded by different types of cameras and different image resolutions. The benchmark dataset that is used in the analysis of tracking algorithms' parameters has been created by SPA software using the manual tracking feature.

The third part of this chapter shows the performance measurements as well as quality criteria of the tracking system. The performance measurements include the running times and rate of error occurrence in tracking. In order for the SPA tracking system to be used by sports scientists as well as trainers it should be tested for the quality criteria in the classical test theory. These criteria include the objectivity, reliability and validity of the tracking. Definitions of the quality criteria and the hypothesis about SPA have been presented.

The numerical results for the testing of both tracking algorithms and for determination of the best parameters are presented in Sections 7.4 and 7.5. Section 7.6 presents the numerical results regarding the running times. The template matching algorithm is faster with more error rate in tracking while the particle filter is slower with less error rate.

Section 7.7 gives a description of the tests used for evaluating the hypotheses defined in Section 7.3.2. It also gives the evaluation of the hypotheses based on references from sports science and sports medicine domain. Some tests can be compared with the work of Perš and Kovačič [84].

8.2 Future Work

Future work concerns developing new tracking algorithms as well as enhancement and acceleration of the current ones. the acceleration of the current tracking algorithms may be done by using parallel hardware platforms such as GPU's (Graphic Processing Units) and multi-core processors. The enhancements may be done by using more closed world assumptions concerning for example the playing positions of the players and using information from speed sensors to enhance the tracking. The future work can be described in the following items:

- acceleration of tracking algorithms such as particle filter using parallel hardware platforms.

- acceleration of image processing using parallel hardware architecture.
- using further closed world assumptions to enhance the tracking.
- using information from speed sensors to enhance the tracking.
- SPA System go from scientific to product.

Bibliography

[1] Forsyth David A. and Ponce Jean. *Computer Vision A Modern Approach.* Prentice Hall, August 2002.

[2] N. de Freitas A. Doucet and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice.* New York: Springer-Verlag, January 2001.

[3] Match Analysis. Internet url: `http://matchanalysis.com`, December 2010.

[4] Yiannis Demirisa Anthony Deardena and Oliver Grau. Tracking football player movement from a single moving camera using particle filters. In *in Proceedings of CVMP*, pages 29–37, 2006.

[5] David H. Salesin Steven M. Seitz Aseem Agarwala, Aaron Hertzmann. Computer vision, image warping mesh and triangulation. Technical report, University of Washington, University of Toronto, Microsoft Research, `http://grail.cs.washington.edu/projects/rotoscoping/`, CSE399b, 2007.

[6] Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, 1991.

[7] S. Avidan. Support vector tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1064–1072, Aug. 2004.

[8] S Arulampalam B. Ristic and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications.* Artech House Publishing, 2004.

[9] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, February 2004.

[10] S.T. Birchfield and Sriram Rangarajan. Spatiograms versus histograms for region-based tracking. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2:1158–1163, June 2005.

[11] Stan Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *IEEE Conf. Comp. Vision Pattern Recognition*, pages 232–237, 1998.

[12] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1):63–84, 1998.

[13] K. Bös. *Handbuch sportmotorischer Tests.* Verlag für Psychologie, Göttingen, 1987.

[14] G. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, 2(Q2), 1998.

[15] Wilhelm Burger and Mark James Burge. *Digital image processing: an algorithmic introduction using Java.* Springer, 2008.

[16] K. Chatyikoloutas C. Kotzamanidis and A. Gianakos. Optimization of the training plan of the handball game. *Handball: periodical for coaches, referees and lecturers*, 2:65–71, 1999.

[17] T. Darrell C. R. Wren, A. Azarbayejani and A. Pentland. Pfinder:real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[18] B. Cambel. An assesment of the movement requirements of elite team handball athletes. *Sports Medicine (3)*, pages 23–30, 1985.

[19] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI*, 8:679–698, November 1986.

[20] Bloomfield J. Nelsen L. Carling, C. and T. Reilly. The role of motion analysis in elite soccer. contemporary performance measurement techniques and work rate data. *Sports Medicine*, 38 (10):839–862, 2008.

[21] Lanlan Chang and Yap-Peng Tan. Hybrid color filter array demosaicking for effective artifact suppression. *Journal of Electronic Imaging*, 15(1):013003, 2006.

[22] K.M.G. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 1:I–77–I–84 vol.1, June 2003.

[23] H. Clarke. *Application of measurement to health and physical education.* New Jearsey, 1976.

[24] R. Collins. Mean-shift blob tracking through scale space. In *IEEE Conf. Comp. Vision Pattern Recognition*, volume II, pages 234–240, 2003.

[25] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):564–577, May 2003.

[26] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. pages 142–149, 2000. Particle Filter Chapter.

[27] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 1:886–893 vol. 1, June 2005.

[28] Jonathan Deutscher and Ian Reid. Articulated body motion capture by stochastic search. *Int. J. Comput. Vision*, 61(2):185–205, February 2005.

[29] A. Doucet. On sequential simulation-based methods for bayesian filtering. Technical Report Technical Report TR310, Cambridge University, 1998.

[30] Ariel dynamic worldwide. Internet URL: `http://www.arielnet.com`, November 2010.

[31] A. Elgammal, R. Duraiswami, and L.S. Davis. Probabilistic tracking in joint feature-spatial spaces. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 1:I–781–I–788 vol.1, June 2003.

[32] Patrick Thomas Emad Monier. Spa software documentation, heinz nixdorf institute, university of paderborn, August 2010.

[33] W. S. Erdmann. Gathering of kinematic data of sport event by televising the whole pitch and track. In *International Socity of Biomechanics. Sports Symposium*, pages 159–162, 1992.

[34] P. Figueroa, N. Leite, R.M.L. Barros, I. Cohen, and G. Medioni. Tracking soccer players using the graph representation. *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 4:787–790 Vol.4, Aug. 2004.

[35] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, Apr 1993.

[36] Y. Seo H. Ok and K. Hong. Multiple soccer players tracking by condensation with occlusion alarm probability. In *Int. Workshop on Statistically Motivated Vision Processing*, 2002.

[37] G.D. Hager and P.N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. volume 20, pages 1025–1039, Oct 1998.

[38] G. Hopkins. Measures of reliability in sports medicine and science. *Sports Med*, pages 1–15, July 2000.

[39] SoftSport inc. Internet url. `http://www.softsport.com`, December 2010.

[40] Stephen S. Intille and Aaron F. Bobick. Visual tracking using closed-worlds. In *International Conference on Computer Vision*, 1994.

[41] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *Int. J. Comput. Vision*, 29(1):5–28, 1998.

[42] M. Isard and J. MacCormick. Bramble: a bayesian multiple-blob tracker. *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, 2:34–41 vol.2, 2001.

[43] Michael Isard and Andrew Blake. Contour tracking by stochastic propagation of conditional density. In *Europ. Conf. on Computer Vision (ECCV)*, volume 1, pages 343–356, 1996.

[44] S. Iwase and H. Saito. Parallel tracking of all soccer players by integrating detected positions in multiple view images. *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 4:751–754 Vol.4, Aug. 2004.

[45] M. Isard J. MacCormick. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proceedings of the 6th European Conference on Computer Vision*, pages 3–19, 2000.

[46] C. De Vleeschouwer J. Olszewska and B. Macq. Speeded-up gradient vector flow b-spline active contour for robust and real-time tracking. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, Hawai, April 2007.

[47] S. Kovačič B. Dežman J. Perš, G. Vučkovič. A low-cost real-time tracker of live sport events. In *the 2nd international symposium on image and signal processing and analysis in conjunction with 23nd int'l conference on information technology interfaces,*, pages 362–365, Pula, Croatia, University Computing Center, University of Zagreb, June 19-21 2001.

[48] Marta Bon Janez Perš and Stanislav Kovacic. Errors and mistakes in automated player. In *Sixth Computer Vision Winter Workshop*, pages 25–36. Bostjan Likar (Ed.), February 7-9 2001.

[49] Y.D. Seo K. Choi and S.W. Lee. Probabilistic tracking of soccer players and ball. In *Asian Conf. Computer Vision*, pages 27–30, January 2004.

[50] E. Koller-Meier K. Nummiaro and L. Van Gool. An adaptive color-based particle filter. In *Image and Vision Computing*, pages 99–110, 2003.

[51] N. De Freitas J. J. Little K. Okuma, A. Taleghani and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *European Conf. Computer Vision*, volume 1, pages 28–39, 2004.

[52] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, (82 (Series D)):35–45, 1960.

[53] Jinman Kang, Isaac Cohen, and Gerard Medioni. Soccer player tracking across uncalibrated camera streams. In *In: Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS) In Conjunction with ICCV*, pages 172–179, 2003.

[54] Zia Khan. Mcmc-based particle filtering for tracking a variable number of interacting targets. volume 27, pages 1805–1918, Washington, DC, USA, 2005. IEEE Computer Society. Member-Tucker Balch and Member-Frank Dellaert.

[55] Seyed Hossein Khatoonabadia and Mohammad Rahmati. Automatic soccer players tracking in goal scenes by camera motion elimination. *Image and Vision Computing*, 27:469–479, March 2009.

[56] O. King and D. Forsyth. How does condensation behave with a finite number of samples? In *6th ECCV*, volume volum 1, Ireland, 2000.

[57] J. Perse M. Kovacic S. Bon M. Kristan, M. Pers. Multiple interacting targets tracking with application to team sports. In *The 4th International Symposium on Image and Signal Processing and Analysis, ISPA 2005*, pages 322–327, 2005.

[58] M. Kristan. *Tracking people in video data using probabilistic models.* PhD thesis, Faculty of Electrical Engineering, University of Ljubljana, May 2008.

[59] M. Kristan, J. Pers, A. Leonardis, and S. Kovacic. A hierarchical dynamic model for tracking in sports. In *Proceedings of the sixteen Electrotechnical and Computer Science Conference, ERK07*, pages 187–190, September 2007.

[60] M. Kristan, J. Perš, M. Perše, and S. Kovačič. Closed-world tracking of multiple interacting targets for indoor-sports applications. *Computer Vision and Image Understanding*, 113:598–611, May 2009.

[61] Mohr M. Amstrup T. Rysgaard T. Johansen J. Steensberg Krustrup, P. The yo-yo intermittent recovery test: Physiological response, reliability. *MEDICINE & SCIENCE IN SPORTS & EXERCISE*, pages 697–705, April 2003.

[62] Mercier D. Gadoury C.and Lambert J. Leacuteger, L. The multistage 20 meter shuttle run test for aerobic fitness. *Journal of Sports Science*, 6 (2):93–101, 1988.

[63] J. P. Lewis. Fast normalized cross-correlation. In *Vision Interface*, pages 120–123. Canadian Image Processing and Pattern Recognition Society, 1995.

[64] G. Lienert and U. Raatz. Testaufbau und testanalyse. *Psychologie Verlags Union*, 1994.

[65] Dave Litwiller. Ccd vs. cmos: Facts and fiction. *Photonic Spectra*, 2001.

[66] W. L. Lu and J. J. Little. Tracking and recognizing actions at a distance. In *Workshop on Compter Vision Based Analysis in Sport Environments In conjunction with ECCV06*, pages 49–60, May 2006.

[67] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[68] A. Blake. M. Isard. Icondensation: Unifying low-level and high-level tracking in a stochastic. In *Computer Vision ECCV98*, volume volume 1, pages 893–908, Freiburg, Germany, 1998.

[69] A. Witkin M. Kass and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, vol. 1(no. 4):321–331, January 1988.

[70] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. 1:572–578 vol.1, 1999.

[71] D. Mackay. *Introduction to Monte Carlo Methods.* MIT Press, 1998.

[72] D. Martin. *Trainingslehre: Kursbuch für die Sporttheorie in der Schule.* Wiesbaden: Limpert, 1990.

[73] S. Maskell and N. Gordon. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Target Tracking: Algorithms and Applications (Ref. No. 2001/174), IEE*, Workshop:2/1–2/15 vol.2, Oct. 2001.

[74] J. Ming Xu, Orwell and G. Jones. Tracking football players with multiple cameras. *International Conference on Image Processing, 2004. ICIP '04. 2004*, 5:2909–2912, Oct. 2004.

[75] A. Gee M. Niranjan N. de Freitas and A. Doucet. Sequential monte carlo methods for optimisation of neural network models. Technical Report TEchnical Report TR 328, Cambridge University, 1998.

[76] C. J. Needham. *Tracking and Modelling of Team Game Interactions.* PhD thesis, School of Computing, University of Leeds, UK, October 2003.

[77] Katja Nummiaro, Esther Koller-meier, and Luc Van Gool. Color features for tracking non-rigid objects. *Special Issue on Visual Surveillance, Chinese Journal of Automation, May 2003*, 29:345–355, 2003.

[78] A. Blake O. Williams and R. Cipolla. A sparse probabilistic learning algorithm for real-time tracking. In *International conference on Computer Vision*, pages 353–360, Nice, France, 2003.

[79] L. Odom and J. Morrow. What's this r? a correlational approach to explaining validitiy, reliability and objectivity coefficients. *Measurement in physical Education and Exercise Science*, 2(10):137–145, 2006.

[80] J. Vermaak M. Gangnet P. Perez, C. Hue. Color-based probabilistic tracking. In *ECCV 02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 661–675, London, UK, 2002. Springer-Verlag.

[81] Daniel Paier. Sports performance analyzer: Eine bewertung der teilkomponente videotracking hinsichtlich validität, reliabilität und objektivität. Master's thesis, Universität Paderborn, Fakulität für Naturwissenschaften, 2009.

[82] Emad Monier Robert Timmermann Michael Dellnitz Felix Werner Per Wilhelm, Patrick Thomas and Ulrich Rückert. An integrated monitoring and analysis system for performance data of indoor sport activities. In *The 10th Australasian Conference on Mathematics and Computers in Sport*, July 2010.

[83] Patrick Thomas Ulrich Rückert Per Wilhelm, Emad Monier. Spa - a system for analysis of indoor team sports using video tracking and wireless sensor network. In *6th International Symposium on Image and Signal Processing and Analysis (ISPA 2009)*, Salzburg, Austria, September 2009.

[84] J. Perš and S. Kovačič. Tracking people in sport: Making use of partially controlled environment. In *Int. Conf. Computer Analysis of Images and Patterns*, pages 374–382, 2001.

[85] F. Pitie, S.-A. Berrani, A. Kokaram, and R. Dahyot. Off-line multiple object tracking using candidate selection and the viterbi algorithm. *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, 3:III–109–12, Sept. 2005.

[86] Petra Platen. Video recording of the handball game between germany and slovenia, handball world cup germany 2007. Technical report, Ruhr-Universität Bochum, Fakultät für Sportwissenschaft, Deutschland, 2007.

[87] ProZone. Internet url. `http://www.prozonesports.com`, December 2010.

[88] A. Doucet R. van der Merwe, N. de Freitas and E. Wan. The unscented particle filter. Technical Report Technical Report TR380, Cambridge University, 2000.

[89] X. Rong Li and V.P. Jilkov. Survey of maneuvering target tracking. part i. dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, Oct. 2003.

[90] Daniel Rowe. *Towards Robust Multiple-Target Tracking in Uncostrained Human-Populated Environments.* PhD thesis, Universitat Autonoma de Barcelona, 2007.

[91] R. Russell and P. Norvig. *Artificial Intelligence, a Modern Approach, chapter 13-18.* Prentice Hall, 2003.

[92] Mohammed Abdel-Megeed Mohammed Salem. *Multiresolution Image Segmentation.* PhD thesis, Department of Computer Science, Humboldt-Universitaet zu Berlin, Germany, November 2008.

[93] Friedhelm Schwenker, Hans A. Kestler, and Günther Palm. Algorithms for the visualization of large and multivariate data sets. pages 165–183, 2002.

[94] C. Sminchisescu and B. Triggs. Kinematic jump processes for monocular 3d human tracking. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 1:I–69–I–76 vol.1, June 2003.

[95] Grimson W. Stauffer, C. Adaptive background mixture models for real-time tracking. In *Conference on Computer Vision and Pattern Recognition*, 1999.

[96] Josephine Sullivan and Stefan Carlsson. Tracking and labling of interacting multiple targets. In *9th European Conference on Computer Vision*, Graz, Austria, May 7-13 2006.

[97] Hyunwoo Kim Sunghoon Choi, Yongduek Seo and Ki sang Hong. Where are the ball and players? soccer game analysis with color-based tracking and image mosaick. In *Proc. ICIAP*, pages 196–203, 1997.

[98] M.J. Swain and D.H Ballard. Colour indexing. *International Journal of Computer Vision*, 7(1):11–32, November 1991.

[99] Andry Tanoto, Jia Lei Du, Tim Kaulmann, and Ulf Witkowski. Mpeg-4-based interactive visualization as an analysis tool for experiments in robotics. In *Int. Conf. on Modeling, Simulation & Visualization Methods*, pages 186–192, 2006.

[100] Hai Tao, H.S. Sawhney, and R. Kumar. Object tracking with bayesian estimation of dynamic layer representations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(1):75–89, Jan 2002.

[101] Patrick Thomas. Projektmanagement in der entwicklung hardware-naher softwaresysteme am beispiel einer anwendung zur sportanalyse. Diplomarbeit, Universität Paderborn, Fachgebiet Schaltungstechnik, September 2009.

[102] TrackSYS. Internet url: `www.tracksys.co.uk`, December 2010.

[103] N. Vandenbroucke, L. Macaire, C. Vieren, and J.-G. Postaire. Contribution of a color classification to soccer players tracking with snakes. In *Systems, Man, and Cybernetics, 1997. 'Computational Cybernetics and Simulation'., 1997 IEEE International Conference on*, volume 4, pages 3660–3665 vol.4, Oct 1997.

[104] X. Varona, J. Gonzalez, F. Xavier Roca, and J.J. Villanueva. itrack: Image-based probabilistic tracking of people. 3:1110–1113 vol.3, 2000.

[105] J. Vermaak, A. Doucet, and P. Perez. Maintaining multimodality through mixture tracking. *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1110–1116 vol.2, Oct. 2003.

[106] Felix Wener. *Vision-based Topological Mapping and Localisation*. PhD thesis, Faculty of Science and Technology at Qeensland University, Brisbane, Australia, October 2009.

[107] F. Werner, U. Rueckert, Tanoto, and J. A.Welzel. The teleworkbench: A platform for performing and comparing experiments in robot navigation. In *Proceedings of the Workshop on The Role of Experiments in Robotics Research at ICRA 2010*, Anchorage, Alaska, USA, May 2010.

[108] Wikipedia. White balance. internet url: `http://en.wikipedia.org/wiki/Color_balance`.

[109] Wikipedia. Bayer filter. internet url: `http://en.wikipedia.org/wiki/Bayer_filter`, December 2010.

[110] Wikipedia. A bayer pattern on a sensor in isometric perspective projection, internet url: `http://en.wikipedia.org/wiki/File:Bayer_pattern_on_sensor.svg`, December 2010.

[111] Wikipedia. Brownian motion. internet url: `http://en.wikipedia.org/wiki/Brownian_motion`, December 2010.

[112] Wikipedia. Cross-section of a bayer filter. internet url: `http://en.wikipedia.org/wiki/File:Bayer_pattern_on_sensor_profile.svg`, December 2010.

[113] Wikipedia. Demosaicing. internet url: `http://en.wikipedia.org/wiki/Demosaicing`, December 2010.

[114] Wikipedia. Voronoi diagrams. internet url: `http://en.wikipedia.org/wiki/Voronoi_diagram`, December 2010.

[115] Emad Monier; Per Wilhelm and Ulrich Rückert. Template matching based tracking of players in indoor team sports. In *Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2009)*, Como, Italy, September 2009.

[116] Chenyang Xu and Jerry L. Prince. Snakes, shapes, and gradient vector flow. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 7(3):359–369, 1998.

[117] N. de Feritas Y. Cai and J. J. Little. Robust visual tracking for multiple targets. In *European Conf. Computer Vision*, volume volume IV, pages 107–118, 2006.

[118] Changjiang Yang, Ramani Duraiswami, and Larry Davis. Fast multiple object tracking via a hierarchical particle filter. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 212–219, Washington, DC, USA, 2005. IEEE Computer Society.

[119] J. Yang and A.Waibel. A real-time face tracker. In *WACV*, pages 142–147, Sarasota, FL, 1996.

[120] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4):13, 2006.

Appendix A

Numerical Results

A.1 Test 2

The detailed numerical results of test 2 for each test person.

Subject	Dist[m]	Err[m]	v[m/s]	Err[m/s]
1	403.1	3.1	2.02	0.02
2	408.1	8.1	2.04	0.04
3	404.3	4.3	2.02	0.02
4	403.5	3.5	2.02	0.02
5	407.1	7.1	2.04	0.04
6	411.7	11.7	2.01	0.01
7	403.7	3.7	2.02	0.02
8	402.9	2.9	2.02	0.02
9	409.1	9.1	2.05	0.05
10	410.6	10.6	2.06	0.06
11	406.8	6.8	2.04	0.04
12	396.6	-3.4	1.98	-0.02
13	398.7	-1.3	1.99	-0.01
avg	405.09	5.09	2.02	0.02
std	4.41	4.41	0.02	0.02
percent		1.27		1.14

Table A.1: Test 2 longitudinal, $v = 2$ m/s, measured covered distance and speed.

Subject	Dist[m]	Err[m]	v[m/s]	Err[m/s]
1	199.7	-0.3	2.98	0.02
2	199.9	-0.1	2.98	0.02
3	201.4	1.4	3.00	0.00
4	201.5	1.5	3.00	0.00
5	203	3	3.02	0.02
6	202	2	3.01	0.01
7	199.4	-0.6	2.97	-0.03
8	201.2	1.2	3.00	0.00
9	205.6	5.6	3.06	0.06
10	201.4	1.4	3.00	0.00
11	199.5	-0.5	2.97	-0.03
12	198.5	-1.5	2.96	-0.04
13	201.1	1.1	3.00	0.00
avg	201.09	1.09	3.00	0.00
std	1.84	1.84	0.03	0.03
percent		0.546		-0.15

Table A.2: Test 2 longitudinal; $v = 3$m/s measured distance and speed.

Subject	Dist[m]	Err[m]	v[m/s]	Err[m/s]
1	374.1	-25.9	1.87	-0.13
2	388.4	-11.6	1.94	-0.06
3	380.6	-19.4	1.90	-0.10
4	362.5	-37.5	1.81	-0.19
5	402.2	2.2	2.01	0.01
6	376.8	-23.2	1.88	-0.12
7	403.4	3.4	2.01	0.01
8	401.5	1.5	2.00	0.00
9	376.3	-23.7	1.88	-0.12
10	405	5	2.02	0.02
11	409.8	9.8	2.04	0.04
12	405.4	5.4	2.02	0.02
13	401.5	1.5	2.00	0.00
avg	391.35	-8.65	1.95	-0.05
std	15.51	15.51	0.08	0.08
percent		-2.16		-2.45

Table A.3: Test 2 transverse; $v = 2m/s$ - measured distance and speed.

Subject	Dist[m]	Err[m]	v[m/s]	Err[m/s]
1	182.7	-17.3	2.75	-0.25
2	192.1	-7.9	2.89	-0.11
3	189	-11	2.84	-0.16
4	178.6	-21.4	2.68	-0.32
5	195.8	-4.2	2.94	0.04
6	181.6	-18.4	2.73	-0.27
7	195.4	-4.6	2.94	-0.06
8	199.2	-0.8	2.99	-0.01
9	188.9	-11.1	2.84	-0.16
10	196.3	-3.7	2.95	-0.05
11	196.8	-3.2	2.95	-0.05
12	195.6	-4.4	2.94	-0.06
13	195.6	-4.4	2.94	-0.06
avg	191.35	-8.65	2.88	-0.12
std	6.66	6.66	0.10	0.10
percent		-4.32		-4.32

Table A.4: Test 2 transverse ; v=3m/s - measured distance and speed

A.2 Test 3

The detailed numerical results of test 3 for each test person.

squar	1m/s		1.5m/s		2m/s	
Subject	dist[m]	err[m]	dist[m]	err[m]	dist[m]	err[m]
1	71.98	-8.02	105.6	-14.4	118.36	-17.64
2	74.68	-5.32	107.4	-12.6	119.58	-16.42
3	76.12	-3.88	111.1	-8.9	123.37	-12.63
4						
5	75.34	-4.66	113.06	-6.94	124.54	-11.46
6	83.66	3.66	130.35	10.35	142.58	6.58
7	84.29	4.29	124.33	4.33	135.38	-0.62
8	81.01	1.01	123.62	3.62	138.37	2.37
9	81.74	1.74	138.01	18.01	134.97	-1.03
10	77.46	-2.54	123.85	3.85	127.28	-8.72
11	78.75	-1.25	114.86	-5.14	129.51	-6.49
12	84.43	4.43	118.14	-1.86	128.67	-7.33
13	76.87	-3.13	110.34	-9.66	122.84	-13.16
avg	78.86	-1.14	118.39	-1.61	128.79	-7.21
std	4.12	4.12	9.83	9.83	7.64	7.64
percent		-1.42		-1.34		-5.30

Table A.5: Test 3 squar run - measured distances.

circle	1m/s		1.5m/s		2m/s	
Subject	dist[m]	err[m]	dist[m]	err[m]	dist[m]	err[m]
1	71.71	-3.89	92.35	-8.45	123.43	-15.17
2	73.25	-2.35	95.88	-4.92	128.8	-9.8
3	69.53	-6.07	89.39	-11.41	120.31	-18.29
4						
5	72.78	-2.82	94.73	-6.07	127.79	-10.81
6	79.7	4.10	102.89	2.09	140.77	2.17
7	78.53	2.93	102.13	1.33	134.58	-4.02
8	77.65	2.05	102.97	2.17	140.65	2.05
9	79.23	3.63	102.84	2.04	142.40	3.8
10	72.5	-3.1	94.44	-6.36	125.98	-12.62
11	72.63	-2.97	95.51	-5.29	126.46	-12.14
12	76.58	0.98	98.98	-1.82	125.31	-13.29
13	68.16	-7.44	91.46	-9.34	120.90	-17.70
avg	74.35	-1.25	96.96	-3.84	129.78	-8.82
std	3.87	3.87	4.86	4.86	7.87	7.87
percent		-1.64		-3.80		-6.46

Table A.6: Test 3 circle run - measured distances.

Square	err 1m/s		err 1.5m/s		err 2m/s	
	run 1	run 2	run 1	run 2	run 1	run 2
1	-8.02	-6.1	-14.4	-10.94	-17.64	-16.77
2	-5.32	-4.01	-12.6	-12.55	-16.42	-16.81
3	-3.88	-2.23	-8.9	-11.06	-12.63	-14.89
4						
5	-4.66	-2.15	-6.94	-7.47	-11.46	-11.58
6	3.66	3.98	10.35	4.47	6.58	3.05
7	4.29	1.33	4.33	1.89	-0.62	0.13
8	1.01	2.16	3.62	-0.96	2.37	-1.26
9	1.74	1.36	18.01	1.47	-1.03	0.79
10	-2.54	-1.43	3.85	-8.18	-8.72	-11.24
11	-1.25	-2.04	-5.14	-6.89	-6.49	-8.17
12	4.43	2.59	-1.86	-3.83	-7.33	-9.26
13	-3.13	-3.64	-9.66	-9.95	-13.16	-14.36
avg	-1.14	-0.85	-1.61	-5.33	-7.21	-8.36
std	4.12	3.08	9.83	5.79	7.64	7.24

Table A.7: Test 3 square run - result of repetition of the tests.

circle	err 1m/s		err 1.5m/s		err 2m/s	
	run 1	run 2	run 1	run 2	run 1	run 2
1	-3.89	-6.1	-8.45	-10.94	-15.17	-16.77
2	-2.35	-4.01	-4.92	-12.55	-9.8	-16.81
3	-6.07	-2.23	-11.41	-11.06	-18.29	-14.89
4						
5	-2.82	-2.15	-6.07	-7.47	-10.81	-11.58
6	4.1	3.98	2.07	4.47	2.17	3.05
7	2.93	1.33	1.33	1.89	-4.02	0.13
8	2.05	2.16	2.17	-0.96	2.05	-1.26
9	3.63	1.36	2.04	1.47	3.8	0.79
10	-3.1	-1.43	-6.36	-8.18	-12.62	-11.24
11	-2.97	-2.04	-5.29	-6.89	-12.14	-8.17
12	0.98	2.59	-1.82	-3.83	-13.29	-9.26
13	-7.44	-3.64	-9.34	-9.95	-17.7	-14.36
avg	-1.25	-0.85	-3.84	-5.33	-8.82	-8.36
std	3.87	3.08	4.86	5.79	7.87	7.24

Table A.8: Test 3: Circle Run- Result of repeatition of the tests

Square	err 1m/s		err 1.5m/s		err 2m/s	
	user 1	user 2	user 1	user 2	user 1	user 2
1	-8.02	-7.45	-14.4	-11.84	-17.64	-0.18
2	-5.32	-5.04	-12.6	-11.03	-16.42	4.46
3	-3.88	-4.29	-8.9	-9.66	-12.63	11.79
4						
5	-4.66	-5.26	-6.94	-4.99	-11.46	8.32
6	3.66	4.36	10.35	3.16	6.58	28.38
7	4.29	3.73	4.33	1.92	-0.62	23.79
8	1.01	3.04	3.62	-0.61	2.37	28.27
9	1.74	1.27	18.01	-1.05	-1.03	22.6
10	-2.54	-3.74	3.85	-7.15	-8.72	12.19
11	-1.25	-4.08	-5.14	-4.25	-6.49	7.19
12	4.43	-3.51	-1.86	-7.17	-7.33	7.19
13	-3.13	-4.05	-9.66	-11.04	-13.16	5.64
avg	-1.14	-2.09	-1.61	-5.31	-7.21	13.30
std	4.12	4.02	9.83	5.22	7.64	9.86

Table A.9: Test 3: Square - results of the tests from two different investigators.

Appendix B

Publications

1. Wilhelm, Per; Monier, Emad; Xu, Feng; Witkowski, Ulf, *Analysis of Indoor Team Sports Using Video Tracking and Wireless Sensor Network.* In World Congress of Performance Analysis of Sport VIII, S. 165-169, Magdeburg, Germany, 3. - 6, September, 2008

2. Monier, Emad; Wilhelm, Per; Rückert, Ulrich, *A Computer Vision Based Tracking System for Indoor Team Sports.* In The fourth International Conference on Intelligent Computing and Information Systems, Cairo, Egypt, 19. - 22, March 2009

3. U. Witkowski, Emad Monier and U. Rückert, *An Automated Platform for Minirobots Experiments*, Intl. Conf. on Control, Automation, Robotics and Vision Hanoi, Vietnam, 17. - 20, December, 2008

4. Monier, Emad; Wilhelm, Per; Rückert, *Ulrich Template Matching Based Tracking of Players in Indoor Team Sports.* In: Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2009), Como, Italy, 30. August - 2. September, 2009

5. Wilhelm, Per; Monier, Emad; Thomas, Patrick; Rückert, Ulrich, *SPA - A System for Analysis of Indoor Team Sports Using Video Tracking and Wireless Sensor Network.* In: 6th International Symposium on Image and Signal Processing and Analysis (ISPA 2009), Salzburg, Austria, 16. - 18. September, 2009.

6. Per Wilhelm; Patrick Thomas, Emad Monier, Robert Timmermann, Michael Dellnitz, Felix Werner and Ulrich Rückert, *An Integrated Monitoring and Analysis System For Performance Data of Indoor Sport Activities.* In: The 10th Australasian Conference on Mathematics and Computers in Sport, 5. - 7 July, 2010.

Abbreviations

Abbreviation	Meaning
CCD	Charge Coupled Device
CMOS	Complementary Metal Oxide Semiconductor
RGB	Red Green Blue
SPA	Sport Performance Analyzer
CONDENSATION	CONditional DENSity PropagATION
GPS	Global Positioning System
ROI	Region of Interest
RW	Random Walk
NCV	Nearly Constant Velocity
CV	Computer Vision
GUI	Graphical User Interface
CWA	Closed World Assumption
HSV	Hue Saturation Value
LUT	Look Up Table
GPS	Global Positioning System
RMSE	Root Mean Square Error
CFA	Color Filter Arrays
JPEG	Joint Photographic Experts Group
TIFF	Tagged Image File Format