Master Thesis

Indoor Activity Detection and Recognition
for Automated Sport Games Analysis

Master's Thesis

at

Graz University of Technology

submitted by

Georg Waltner

Institute for Computer Graphics and Vision (ICG),
Graz University of Technology
A-8010 Graz, Austria

May 20, 2014

Supervisor: Univ.-Prof. Dipl.-Ing. Dr.techn. Horst Bischof
Advisor: Dipl.-Ing. Thomas Mauthner

TUG Graz University of Technology
Erzherzog-Johann-University

Diplomarbeit

Indoor Activity Detection and Recognition
for Automated Sport Games Analysis

Diplomarbeit

an der

Technischen Universität Graz

vorgelegt von

Georg Waltner

Institute für Maschinelles Sehen und Darstellen (ICG),
Technische Universität Graz
A-8010 Graz, Österreich

20. Mai 2014

Diese Arbeit ist in englischer Sprache verfasst.

Betreuer:	Univ.-Prof. Dipl.-Ing. Dr.techn. Horst Bischof
Mitbetreuender Assistent:	Dipl.-Ing. Thomas Mauthner

Abstract

Action and activity recognition has often been evaluated solely regarding single person behavior. Recently, successful methods proved the use of information concerning the surrounding scene to be valuable for this recognition task. One of the fields for action and activity recognition is sport. Sportsmen generally require to expend extraordinary effort when trying to succeed on a professional level. Game, player and team analysis is of great interest and such research topics within this field emerge with the objective of automating the analysis process. Each sport has very specific underlying rules, which can be used as prior knowledge for the recognition task and presents a constrained environment for evaluation of new methods.

Until now, the classification of player activities on the court is done by professionals. Such specialists produce a game statistic either after the game from recorded video or already during the game by live observation, helping the team or coaches to make successful tactical decisions.

Within this thesis, player activities in volleyball sport are investigated by methods of computer vision and pattern recognition. The automated recognition and rating of all procedures during the game for statistical report generation is the final goal. This thesis contributes a first step towards this goal by implementing an activity recognition system. The first stage of the system is a per-frame player-centered activity recognition using shape, motion and spatial information. After camera calibration, a preprocessing step is needed - removing the background through median-filtering and building color models (*Gaussian Mixture Models (GMM)*) for the two teams. After these steps the players are segmented. Using a planar homography linking court plane and image plane their positions can be estimated and a player distribution calculated. The seven different activities are modeled by descriptors for shape (*Histograms of Oriented Gradients (HOG)*), motion (*Local Histograms of Oriented Flow-Magnitudes (HOF)*), position on the court (*Real World Player Coordinates (RWPC)*) and player distribution context (*Spatial Context (SC)*). Using a learning algorithm (*one-vs-all Support Vector Machine (SVM)*) based on these descriptors, a classification model for player activity recognition is learnt. As not only single players are of importance in team sports, thereafter in a second stage all other players on the court are involved for recognition of activities. Using the before trained GMM, players are detected and their activities evaluated via the above SVM. This information about all player activities is incorporated by an activity context descriptor (*Activity Context (AC)*). The AC descriptor exploits information about all player activities over a certain timespan relative to the investigated player. Together with the features from the first stage, the AC features are used to train a new SVM for classification of player activities.

The benefit of this context information on single player activity recognition is evaluated on a new real-life dataset (50% train / 50% test split) presenting a total amount of almost 36k annotated frames containing 7 activity classes within 6 videos of professional volleyball games. Incorporation of the contextual information improves the average player-centered classification performance of 77.56% by up to 18.35% on specific classes, proving that spatio-temporal context is an important clue for activity recognition.

Keywords

Computer Vision, Action Recognition, Activity Recognition, Gaussian Mixture Models, Histograms of Oriented Gradients, Histograms of Oriented Flow, Spatial Context, Activity Context, Support Vector Machines.

Kurzfassung

Das Erkennen von Aktionen und Aktivitäten wurde bisher oft auf die Handlungen einzelner Personen beschränkt. In letzter Zeit konnte gezeigt werden, dass das Einbeziehen von Information über die Umgebung der untersuchten Person (Kontext) diese Aufgabe unterstützt. Eines der Interessensgebiete für das Erkennen von Aktivitäten ist Sport. Sportler müssen im Allgemeinen viel Aufwand betreiben, um auf einem professionellen Level erfolgreich sein zu können. Die Analyse von Spiel, Sportler und Mannschaft ist daher von großem Interesse und Forschung in diesem Bereich wird intensiviert, um eine Automatisierung zu erreichen. Jeder Sport bietet durch seine spezifischen Regeln Informationen, die für die automatische Analyse genützt werden können und stellt gleichzeitig eine gute Umgebung dar, um neue Methoden zu testen.
Bisher wird die Bewertung der Aktivitäten im Volleyball Sport durch Experten manuell durchgeführt. Diese erzeugen Spielstatistiken einerseits nach dem Spiel auf Basis einer Videoaufnahme oder bereits während des Spiels. Solche Informationen erlauben dem Team und dem Trainer gute taktische Entscheidungen zu treffen.
In dieser Diplomarbeit werden Volleyball Aktivitäten mit Methoden der automatischen Bildverarbeitung untersucht. Mit dem Endzweck, alle Aktivitäten eines Spiels fehlerfrei zu Erkennen und zu Bewerten, stellt diese Arbeit einen ersten Schritt in diese Richtung dar. Der erste Teil des implementieren Systems besteht aus einer bildweisen spielerzentrierten Erkennung mit Hilfe von Form, Bewegungsinformation sowie räumlicher Information. Nach einer Kamerakalibrierung wird eine Bildvorverarbeitung zum Filtern des Hintergrundes sowie zum Erzeugen von Farbmodellen (*Gaussian Mixture Models (GMM)*) benötigt. Danach können die Spieler segmentiert und mit Hilfe eines berechneten geometrischen Modells ihre Positionen und Verteilung am Spielfeld geschätzt werden. Sieben definierte Volleyball Aktivitäten werden durch Deskriptoren für Form ((*Histograms of Oriented Gradients (HOG)*), Bewegung (*Local Histograms of Oriented Flow (HOF)*), Spielfeldposition (*Real World Player Coordinates (RWPC)*) und Spielerverteilung (*Spatial Context (SC)*) beschrieben. Unter Verwendung von *one-vs-all Support Vector Machines (SVM)* wird auf Basis dieser Deskriptoren ein Klassifizierungsmodell gelernt. Da im Teamsport nicht nur Einzelspieler isoliert betrachtet werden sollten, werden im zweiten Schritt alle anderen automatisch erkannten Spieler bezüglich ihrer Aktivitäten klassifiziert und mit Hilfe eines Deskriptors (*Activity Context (AC)*) beschrieben. Dieser verwendet die Information über alle Aktivitäten der Spieler innerhalb eines gewissen Zeitrahmens. Eine weitere SVM wird gemeinsam mit den vorherigen Features trainiert, um die Aktivität des untersuchten Spielers besser zu bestimmen.
Zur Evaluierung der Bedeutung des Verhältnisses eines beobachteten Spielers mit seiner Umgebung wurde ein Video Datensatz mit ungefähr 36.000 Annotierungen in sechs Videos von professionellen Volleyballspielen erstellt. Die sieben Aktivitätsklassen wurden darauf (50% Trainingsdaten, 50% Testdaten) evaluiert und es konnte gezeigt werden, dass durch Verwendung von Kontextinformation die Erkennungsrate von 77.56% nach der Einzelspieleruntersuchung für Volleyball spezifische Aktivitäten um bis zu 18.35% gesteigert wurde. Somit wird gezeigt, dass räumlich-zeitliche Kontextinformation für die Erkennung von Aktivitäten eine große Rolle spielen kann.

Schlagwörter
Bildverarbeitung, Aktivitätserkennung, Gaussian Mixture Models, Histograms of Oriented Gradients, Histograms of Oriented Flow, Spatial Context, Activity Context, Support Vector Machines.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, on May 20, 2014

_____
(Georg Waltner)

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am 20. Mai 2014

_____
(Georg Waltner)

Contents

List of Figures

List of Tables

Danksagung

Mein größter Dank gilt meinen Eltern, die mich seit jeher bedingungslos unterstützt und gefördert haben. Sie haben mich zu jeder Zeit ermutigt meinen Interessen zu folgen und mir ermöglicht mich auch abseits des Studium zu verwirklichen. Neben meinen Eltern hat mich auch meine einzigartige Freundin Gaby in den letzten Monaten und Jahren unterstützt und war mir mit ihren stets positiven Worten ein großer Rückhalt in den Höhen und Tiefen dieser Zeit. Ich möchte diesen Personen die vorliegende Arbeit widmen.

Weiters möchte ich mich besonders bei Thomas Mauthner für die persönliche und intensive Betreuung bedanken, er hat mir mit seinem Wissen, seinen Ideen und seinen Anregungen in unzähligen Gesprächen sehr geholfen. Auch Horst Bischof gilt mein Dank für die Betreuung und seine Kommentare.

Georg Waltner
Graz, am 20. Mai 2014

Chapter 1

Introduction

Today, activity recognition from video is a hot topic in computer vision. More and more applications emerge from the field with the aim to analyze person behavior. Target applications can be quite different but still share the same goal: Automatic recognition of human activities by identifying reoccurring movements connected to certain situations. The focus of this thesis lies on activity recognition in sport and in volleyball in special. Sport can be a good field of study as there is more than enough data for research and the topics diverge broadly as there are so many forms of sport. There exist single player sports where the athlete performs alone and team sports with a big variety of player numbers, play fields or play styles. On the one hand there is golf, tennis, weight lifting, track and field athletics or wrestling and on the other hand football, soccer, (ice)hockey, basketball, handball or polo. The pluralism of sports is fascinating.
Especially the rules of a sport game are very important. As they are very strictly and clearly defined, they determine the structure of the play and game strategies. This sport specific game structures and strategies can provide necessary domain knowledge to build a recognition system, knowing what behaviors are allowed and what can be expected during activity analysis. Obviously the best way of incorporating this prior knowledge for building a recognition system is expert knowledge. A field of application for recognition systems is sport game analysis, where performance of players is statistically evaluated to gain information. The use of video analysis in sport has two effects: increase of sporting performance and increase of coaching performance. Both lead to improved overall results in competition. While the sporting performance can be bettered in terms of technique, motivation and feedback, the coaching performance is optimized referred to technique analysis, communication and efficiency. As professionalism in sport increases, statistical methods have become vital to detect the opponents weaknesses and to adapt own strategies. Video analysis gives clues about typical reactions of players in special circumstances. This analysis is highly complex, a special trained person needs to evaluate the players activities by assigning type and quality. The coaches (and/or players) such can use this information to develop strategies according to the opponents preferred reactions in specific situations. As result, the opponents flaws can be exploited and his strengths mitigated. Tactical strategies for game situations (attack/defense/...) in team sports involve some (football) or all players (volleyball, basketball, handball) of a team. It is very important, that every player sticks to the plan for a certain game situation as the overall goal is to make it easy for the own team and as hard as possible for the opponent to score.

1.1 Scope

Within the scope of this thesis, different occurring activities in volleyball sport (service, reception, attack, block,...) are detected by an activity recognition system on real world video data. As previously there existed no data set for volleyball, much effort was put into creating data over multiple games [1]. Having created this data set (see Section 5.4 for detail), up-to-date methods for activity recognition are used together with specially developed methods incorporating expert knowledge about the game structure.
After calibration of the videos, determination of real world geometry and modeling of foreground and background, the main task of activity recognition is tackled. At first, the system examines single frames for spatial context activity recognition, assigning activity classes to players. For this task typical descriptors like Histograms of Oriented Flow (HOG) or Histograms of Oriented Gradients (HOG) are used together with newly developed spatial descriptors. The first, Real World Player Coordinates (RWPC), is a simple player location descriptor while the Spatial Context (SC) descriptor works as a map of player probabilities estimated from blob images for modeling the on court player distribution. Subsequently, more complex activity recognition is done by temporal integration of the previous learnt activities. This is based on the Activity Context (AC) descriptor, composed of the distribution of player activities on the field observed over a certain number of frames in the past. Both tasks consist of feature extraction as first step and subsequent classification with a Support Vector Machine (SVM). Figure 1.1 displays a simplified block diagram of the implemented computer vision system.
Due to the camera setup the work is limited to one half of the court, separated from the other by the net. An extension to the whole court can be easily done later by setting up a second camera on the opposite side of the gym. In the end, a system similar to the proposed one should be able to detect and classify player activities for automatic generation of statistic reports. These reports can be used for strategic game adaptions and decisions made by the coaches before, during and after matches. An extension to other sports is imaginable, as the newly proposed methods base on real world geometry and are generalizable to other sports, especially team sports.
After this introductory Chapter 1, the following Chapter 2 gives overview of the related work done in activity recognition. Chapter 3 describes the theoretical background used from previous research. The purposely designed methods are covered in Chapter 4. Chapter 5 outlines the developed system and gives an overview on the basic rules of volleyball. The experiments carried out are shown in detail in Chapter 6. Finally, Chapter 7 gives an overview of trends and ideas for future work, while Chapter 8 summarizes the presented work.

1.2 Action/Activity Recognition

The recognition task should aim to automate detection and naming of specific activities observed from videos. This process usually consists of detection of persons, extraction of features and classification of previously learned movement patterns. Actions are mostly short and simple single person motion patterns without relationship among each others. Activities are complicated series of multi person actions often connected in some context. As this thesis focuses on specific activities that consist of simpler actions like *running*, *jumping*, *hitting*, *landing* or *moving*, the term activity is used throughout the thesis.
This works emphasis lies in full-body activity recognition, in contrast to gesture recognition or facial expression recognition. In literature the representation of the pose and movement ranges from very complex body models (where the body is split up into parts like limbs, head, torso) to simpler silhouette descriptions. During feature extraction, robustness to (partial) occlusion, background clutter,

[1] HD videos from games of UVC Graz in the Austrian Volley League (AVL)

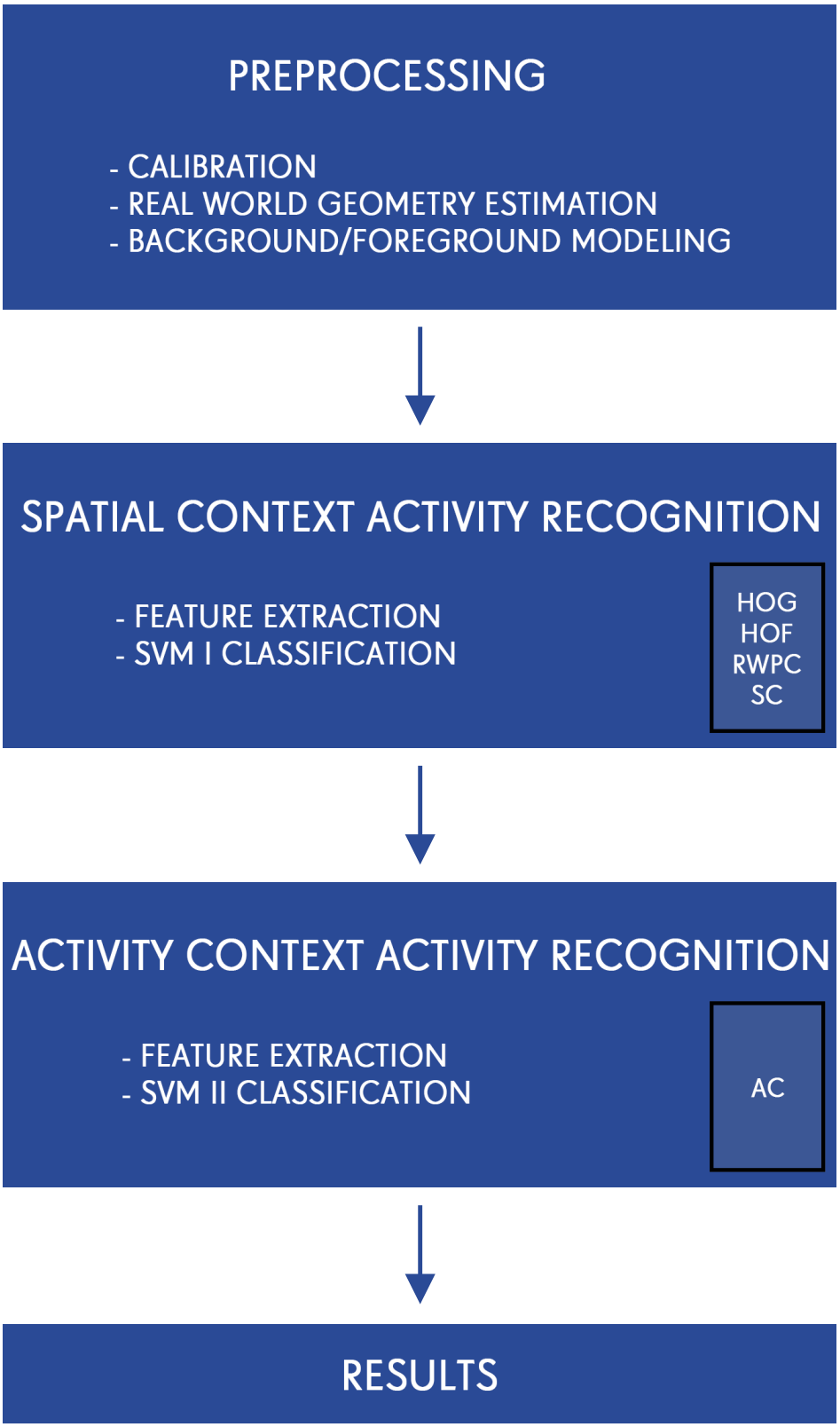
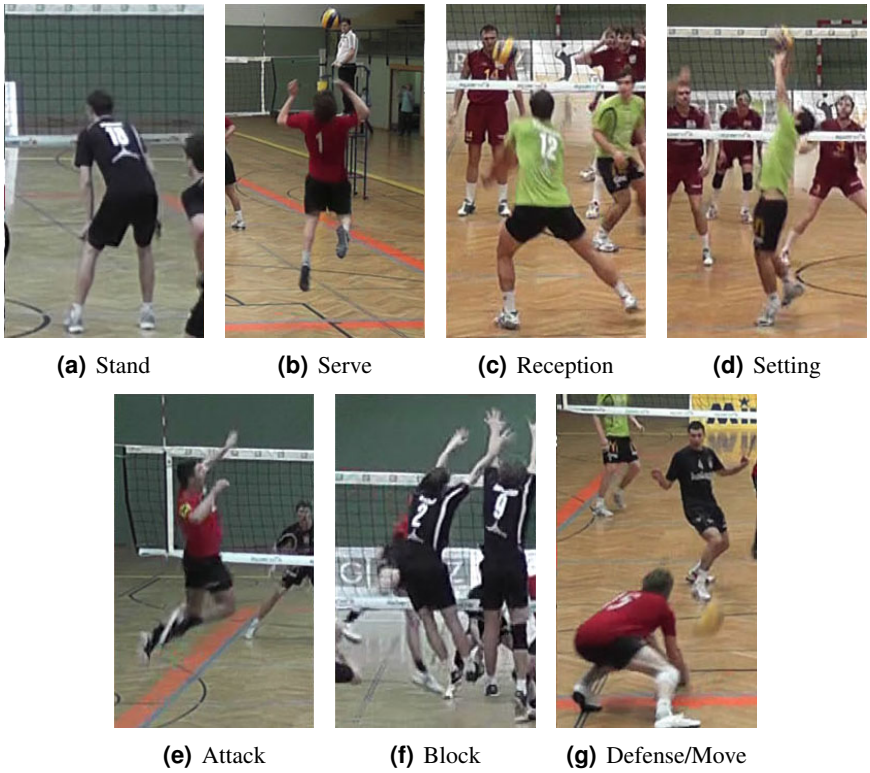


Figure 1.1: *System overview: Preprocessing is followed by player activity recognition from single frames with spatial context and player activity recognition, supported by activity context generated by other players over multiple frames.*

shadows or illumination changes need to be addressed for good performance. Also a certain intra-class variability must be allowed for correct classification of the same activity performed by different persons (speed, style, size of the person,...). Still the classification must be strong enough to differentiate between similar movements of different classes (run vs. walk, ...). It is a challenge to design activity models, which can handle this trade-off.
A survey by Weinland et al. [2011] names two domains for activity recognition in literature: temporal and spatial. In the spatial domain, activity recognition can be based on global image features (geometry of scene or camera), parametric image features (geometry of human body) or on statistical models (distribution of local image features). In the temporal domain, activity recognition can be based on global temporal signatures (stacked features, representing entire activities), grammatical models (sequential organization of activities with states and transitions) or statistical models (distributions of sparse and unstructured feature observations). Activity recognition within the context of this work denotes discrimination of various complex volleyball activities performed during a match. To keep the annotation effort limited, seven main activities were defined as shown in Figure 1.2.



(a) Stand **(b)** Serve **(c)** Reception **(d)** Setting

(e) Attack **(f)** Block **(g)** Defense/Move

Figure 1.2: *The volleyball activities were categorized into the seven complex classes "Stand", "Service", "Reception", "Setting", "Attack", "Block", "Defense/Move". The depicted images were chosen for demonstration - shape, color and execution style are subject to broad variations in the data set.*

1.2.1 Challenges

Converting the theory into a working system in practice is always a challenge. Some of the emerging problems and their tackling will be discussed within this subsection: Player differences in execution of the activities, occlusion, tracking, color differences, background clutter.

Execution Differences

In every sport, the execution of typical movements differs from player to player. Although there might be a ”perfect” technique reference, only few players really manage to execute perfectly. Especially when mental pressure is present during competition, the execution is often worse than in training situations. In Volleyball, the players posture, size and execution speed is of great range. This leads to different sequences of particular activities and makes the distinction harder. To make differentiation possible, a rather large grid in the HOG/HOF features is used and a large class intra-variability allowed by the SVM.

Occlusion

Volleyball is a very dynamic sport, players often change positions and especially in defense situations there are many possible situations which only occur randomly and don't necessarily follow patterns. As the camera view is always from behind the court and six players per team move within an area of about 100m^2 ($9m \times 9m$ field and outer court), players are often at least partly covered by their team mates. Especially in service or reception situations players change places in court to arrange for an optimal attack or defense formation.

Tracking

Due to the camera position automated tracking is hard or even impossible to solve, working with a single camera setup. Without the use of more cameras (and even then) it is very difficult to keep track of single players over time. This is why the experimental data comes from manual annotated videos, preventing introduction of errors already in the tracking stage.

Color Modeling

Like in soccer - where the goalkeeper wears a different colored jersey - the libero as a specialized defense player in volleyball also wears a different color than his colleagues. This makes a simple color model per team difficult. Also, the color of the pants is mostly different to the color of the jerseys. Adding to that, the opponent team might have similar or same colors (often the pants of both teams are black).
To account for the problem of similar colors, Gaussian Mixture Models (GMM) are used. For each team a small subset of samples (typically 5 rectangles for background and players) is chosen by hand and a GMM is generated for each team separately. These models mostly allow good discrimination between players of the two teams. However, when jersey color of one team and pants color of the other team are similar or the same, one team often interferes with the other teams predictions. Figure 1.2 shows examples from different videos, where black is often appearing as pant or jersey color.

Background clutter

As the camera setup is fixed, the background is mostly still and can be eliminated. Except for the opponent team, coaches, referees or other persons. They are moving on the other side of the net and around the court, often with similar colored jerseys making background filtration and player segmentation a challenging task. Like above with occlusion, this problem is difficult to solve. Masking regions is not an option, as players outside the field or jumping would be ignored and removed from the recognition process. For a better estimation of players and their position, scale dependency (subject to distance from the camera) is used.

Chapter 2

Related Work

Most previous work covers the tracking of players in team sports (soccer, basketball, volleyball,....) or one-vs-one sports (tennis, badminton, squash, ...) as well as extraction of different statistical data: How often does one player get the ball? How many sprints does he do or what distance is covered during a game? Who is on court when the team scores a goal?
In many approaches a multi-camera-setup is used to get more visual information from different angles and distances. Often, this is the only applicable solution due to big courts/fields (football, soccer, basketball, hockey), where a single camera can only provide enough detail surveying a limited range. Single-camera-setups are often used, when the distance to the players is small, only a part of the court is evaluated, the camera can be mounted overlooking the whole scene of interest (squash, badminton, volleyball, basketball...) or only one player is analyzed (golf, tennis,...). Latest years show improvements in computer vision tasks, there is much interest from various fields of study for being able to recognize behavior. **Human computer interaction (HCI)** helps controlling computers and devices via gestures like in Uebersax et al. [2011], Wu and Huang [1999] or Freeman and Weissman [1995]. In **Medicine** diagnostics for orthopedic patients and gait recognition, see Lakany et al. [1999], Kohle et al. [1997] or Meyer and Denzler [1997], is based on automated recognition. **Surveillance** is also becoming more and more automated, demanding smart surveillance systems in sensitive areas such as banks, department stores, parking lots or borders, public places surveillance as presented in Collins et al. [2000], Maybank and Tan [2000] or Haritaoglu et al. [2000]. Also the **Gaming** industry is interested in operating computer games with body movements and use of augmented reality for enriching the user experience. See Tang et al. [2011] or work by Frati and Prattichizzo [2011] on the popular XBox Kinect. After the introduction of face recognition into daily live applications (photo camera, laptop unlock screens, ...) **Identification** is moving to the next level: Biometric features are used for personal identification, see Little and Boyd [1996], Shutler et al. [2000], Huang et al. [1999] and Cunado et al. [1999]. The financially and politically powerful **Warfare** is interested in surveillance and monitoring of battlefields like in Collins et al. [1999]. In **Traffic** regulation, tracking vehicles (see Remagnino et al. [1998]) and pedestrian flow (see Boghossian and Velastin [1999] and Boghossian and Velastin [2002]) are under investigation. Last, but not least personalized training systems, choreography of dance and ballet, indexing and retrieval systems and augmented reality presentation as in Yu and Farin [2005] are interested topics in **Sport**. Soccer is clearly the sport with most interest worldwide, so various recognition systems have been presented: Automatic offside detection like in Hashimoto and Ozawa [2006], automatic video analysis of goals, classification of shots and various detections (referee, penalty box, slow-motion replay...) as in Ekin et al. [2003], player and ball detection/classification (Poppe et al. [2010] or Liang et al. [2005]), players involvement in game (see Leo et al. [2009b]), unsupervised segmentation and clustering of players like in Spagnolo et al. [2007] or activities recognition by silhouette clues in soccer shown in Leo et al. [2009a].

There also exists work in other sport areas. Pitcher analysis in baseball, where four types of pitches are distinguished automatically from television broadcast like in Chen et al. [2011]. Tracking of a golf club and the swing for 3D construction (Urtasun et al. [2005]). Success rates for forehand- and backhand-strokes in tennis as shown in Zhu et al. [2006]. Tracking of beach volleyball players as in Mauthner et al. [2008]. Only few of the above mentioned works are unsupervised recognition systems like the one proposed in this thesis.

2.1 Action and Activity Recognition

A survey from Turaga et al. [2008] discusses the difference between *actions* and *activities* in depth. Actions are referred to as simple motion patterns executed by a single person and typically lasting a short period of time. The actions are in no relationship with each others and the order or occurrence is mostly arbitrary. On the other hand, activities denote a complex series of actions performed by several persons who might be interacting with each other. They typically occur in specific patterns or succession, which is used as contextual information. Within this thesis, the term *action* is not used as it would denote simple motion patterns like *jumping* or *hitting*. In contrast, the term *activity* defines a more complex motion scheme and is more suitable to the investigated sport motions like *serving* or *attacking*, executed by a player. These sport activities are mainly characterized by the interaction between the players and the ball but can also be seen in context with the court, referees and other non-player objects. Many papers use position information of persons for activity recognition. Opposite to the presented work, many of them do not classify the single player activities but only the team activities as a whole.
This thesis will follow the diction of Turaga et al. [2008] and as stated in the survey there is a significant ”gray-area” between actions and activities where the distinction is hard to make.
According to the survey an activity recognition system consists of four major steps:

1. Input video or sequences of images (preprocessing, Section 3.1).
2. Extraction of low-level features (feature extraction, Sections 3.2, 4.1 and 4.2).
3. Action description from low-level features (spacial context player activity recognition, Section 6.2).
4. High-level semantic interpretation of activities from primitive actions (activity context player activity recognition, Sections 4.3 and 6.3).

As marked in braces, this thesis follows roughly the suggested steps for building an action and activity recognition system for volleyball sport but does not first describe simple actions and more complex activities afterwards. Instead, first player activities with focus on one player at a certain moment are described, followed by enhancing this description by integration of knowledge about other player activities over a period of time.

2.1.1 Action Recognition

According to Poppe et al. [2010], action recognition can be subdivided into three main approaches for image representation. The first is global representation, where the image is analyzed in a top-down fashion that encodes the visual observation as a whole. After person localization by background subtraction or tracking, the region of interest around a person is determined and encoded by descriptors. Descriptors might be based on silhouettes, shapes or motion (flow). For the global approach to work the tracking needs to be accurate and the viewpoint can only differ slightly. Also occlusion

is a problem, as it strongly influences the results of the descriptors. Examples of such descriptors are motion-energy images (MEI) and motion-history images (MHI), developed by Davis and Bobick [1997], that use foreground motion over time to describe human activities like sitting down or waving as well as aerobic exercises. Tran and Sorokin [2008] designed a model, using information about motion (optical flow in two directions) and shape (player silhouettes) to recognize badminton actions. The approach is similar to the well-known shape context descriptor from Belongie et al. [2002] but includes motion context over time. The model was used to determine type of motion, type of shot and predict if a shot was executed or not. Thurau and Hlavac [2008] presented an approach using an HOG based descriptor. After detection of a person, histograms of pose primitives are calculated and compared to previously generated pose histogram templates. Efros et al. [2003] showed that motion includes enough information to recognize sport activities (ballet, tennis, football) from distance. They created motion channels in a spatio-temporal manner and compared these motions with action templates. By suppressing camera motion and emphasizing on the persons motion, Jain et al. [2013] showed improved recognition results. Again, a combination of local descriptors (HOF Laptev et al. [2008], HOG Dalal and Triggs [2005], MBH Dalal et al. [2006]) and a motion descriptor (flow trajectories) was used. Wang and Suter [2007] described simple human actions like bend, jump jack or run with the silhouettes of persons. Ikizler et al. [2008] combined shape (line features) and motion to recognize six action with challenging problems like different viewpoints, varying outfits or mixture of outdoor and indoor recordings.

The second group of approaches is keypoint based. Therefore keypoints are extracted in an image and a region around these points is evaluated. This is a patch-based approach using bag-of-features or bag-of-words methods to sum up information gathered from multiple patches into descriptors. As an advantage this can be used in surroundings with moving background and handle occlusion rather well. Such this approach is more robust than the previously proposed global approach. One of the first proposed descriptors was the Harris corner detector proposed by Harris and Stephens [1988], that was later extended to 3D by Laptev and Lindeberg [2003]. Popular examples of keypoint descriptors are the scale-invariant feature transform (SIFT) proposed by Lowe [2004] and the 3D SIFT descriptor proposed by Scovanner et al. [2007]. For behavior analysis of mice, Dollár et al. [2005] proposed using spatio-temporal descriptors, so called cuboids, based on keypoint detections. Bregonzio et al. [2009] proposed a method using clouds of space-time interest points to recognize human actions. In combination with random ferns, Oshin et al. [2009] used the distribution of interest points in their method.

The last group of approaches is a mixture of methods that do not typically fall into the previous two categories. They are typically very application specific in contrast to the general approaches discussed above. Examples are joint locations or joint angles, see Forsyth et al. [2005] for detailed information. A good example for this group is the work of Smith et al. [2005], where descriptors are specially designed for a group of actions. They determine, if an object is held in hand, where on the face an action (rubbing, taking of glasses,...) occurs or counts how many hands are involved in the action. A number of such features is combined and AdaBoost (Freund and Schapire [1995]) then selects the best ones for a result.

2.1.2 Activity Recognition

Many approaches base on position information of the players. Using trajectories of the players, activity recognition in a basketball game was implemented, where the coach designs a strategic code-book with different complex defense or attack activities, having several players involved. The tracking results (trajectories) are then compared to the templates in the code-book by Perše et al. [2006], Perše et al. [2008]. In this work, no player activities were recognized, but only different team offense and defense activities.

A very similar approach to that proposed in this thesis was presented by Bialkowski et al. [2013]. A hockey field was recorded by eight HD cameras and players of both teams extracted by background subtraction and color models. Team activities were expressed by position context, once with occupancy maps dividing the field into rectangles and once by calculating elliptical team centroids. As the players activities were not evaluated, only the team positions on the field were used to decide on the occurring team activity.
Atmosukarto et al. [2013] tried to recognize offensive team formations in American Football at the beginning of a play. After video registration and detection of the separation line (line of scrimmage) between the two teams, the spatial distribution of the players on the field is used to identify the offensive team. Having identified the offense players positions, the offensive play is classified as one of five formations.
Similar to Bialkowski et al. [2013], the work of Gade and Moeslund [2013] uses occupancy maps to recognize the type of sport within a sports arena. Player positions, represented as gaussian distributions, are combined over time into heatmaps that correlate to a individual sport type (badminton, basketball, handball, soccer, volleyball and miscellaneous).
Of course, there exist other needs for activity recognition than sport. The activities of groups in surveillance videos were examined by Lan et al. [2012] by describing the activity of an individual person as well as the behavior of other persons nearby. This is again similar to the presented approach where first individual players are analyzed and then the analysis is combined over all players on the field.
Zhu et al. [2013] connected the individual activities in a scene to create an activity context. With the segmented motions (continuous motions divided into action segments) in a video, they set the segments in context between themselves. Action segments that are related to each other in space and time are grouped together into activity sets. The combination of spatial and temporal context helps distinguishing activities.
Another example for activity recognition was demonstrated in the paper of Scovanner and Tappen [2009]. By calculation of a pedestrian model including movement cost, velocity, destination and avoidance probabilities a prediction about the future path of a person moving through hallways was generated.
Brendel and Todorovic [2011] built a spatio-temporal graphs from videos. The nodes are built from video segments of different scales and are connected by edges representing hierarchical, temporal and spatial relationships. In this way, the model learns what is important and significant for an activity by optimizing the graphs so that similar activities can be differentiated.
Another spatio-temporal model was examined by Ryoo and Aggarwal [2009], measuring structural similarities between sets of features. They evaluated their method on videos containing multiple interacting persons. As the similarities between two videos are based on feature points they can detect and classify non-periodic activities.

2.2 Context

Many successful approaches methods share on common method, using context to improve the recognition results. Not only the persons, animals or objects examined provide information, but also persons, animals or objects in their vicinity give clues. The context is a collection of these clues in spatial and temporal dimension.
One of the first publications about context was the work of Schyns and Oliva [1994]. They executed experiments about the human perception and displayed four images of different scenes (highway, city, living room,...) for a very short time period (125ms). The persons viewing the image sequences had to push a button if one special scene was displayed. Of course the presence of cars, trucks and road

signs is a strong indication for the scene to be of the class highway. Added blur to the images did not impair the correct interpretation of the scene images with 96% correct classifications. This shows that multiple objects in an image can give strong clues for human perception about the whole scene.
Another early example was constructed by Hollingworth and Henderson [1998] where participants were shown labels for objects like mixer or chicken. Afterwards a natural scene was displayed for a short period of time followed by a mask with embedded cue about the object location. The participants had then to decide if the object was at the proposed location or not. The detection performance was higher for semantically consistent versus inconsistent objects. However, when presenting the object after the scene, no effect was documented. This indicates, that context is only of use if humans know before what they have to look for in an image.
In the field of computer vision, it has been over a decade since Belongie et al. [2002] proposed the shape-context-descriptor for matching silhouettes and hand written digits. The approach bases on the idea that points on a silhouette are related and such angles and distances remain similar between these points of objects of a class. Based on the shape-context that models similar shapes, also other spatial context descriptors emerged.
Hoiem et al. [2006] put local objects in context with the surrounding 3D scene. They used probabilistic estimates of street scenes like ground, sky or vertical structures and estimated the scene geometry in 3D and also the camera viewpoint. With the estimates about the scene geometry and camera position the scale (and such location) of present objects can be limited and such the search for them simplified. Of course also the knowledge about some object successfully improves the guesses about geometry an camera viewpoint and help to find other objects in the scene.
Oliva and Torralba [2007] presented an overview of context for object recognition. According to them, different levels of context can be discriminated. Semantic context concerns the occurrence of similar objects in images, for example a table and a chair are likely to appear together while a bed and an elephant are not. Spatial context describes the positions of object, for example a keyboard is likely to be placed under a monitor but not on top of it. The third mentioned context is pose, for example a car will mostly be parked along the driving direction of and not across the street.
With the proposal of auto-context, Tu [2008] showed that not only the context of objects among each other, but also the context of parts of the object itself can lead to improved recognition. With an iterative learning scheme, extracted patches are weighted to segment objects in images.
The exploitation of context in natural dynamic scenes from video was researched by Marszalek et al. [2009]. They evaluated a bag-of-features framework to investigate 69 videos. The context was integrated by using text scripts belonging to the videos and visual context from the video frames. For most classes, the use of context improves results.
Burgos-Artizzu et al. [2012] showed, that using not only spatial, but also temporal context information can support the decision about rodents interactions. Videos of social mice behavior were captured by two cameras and categorized into thirteen different activities. To describe the activities they added agent trajectory features to the usual spatio-temporal features. Using a large pool of weak position features like distance between mice, movement direction, velocities and accelerations improves the recognition in comparison to only using spatio-temporal information.

Context in Sport Activity Recognition

In Kolonias et al. [2004], the authors point out, that the rules of sport provide a good basis for occurring events or activities. They presented a multilevel contextual model using Hidden Markov models to understand tennis videos and discriminate between points won by the player near to the camera and far away from the camera. They use the context provided by the events to make guesses for the following events. For example, a service can be followed by a hit of the opponent (if the service is placed within the service area) or by another service (if the ball goes into the net or touches the net

while crossing it). They evaluated their method with only few errors on one hour of tennis from the Australian Open Final in 2003, consisting of 100 played points or one and a half set of the match.
Ghanem et al. [2012] use a set of 82 visual, audio and textual features to extract highlights from American Football videos. Since two plays with very similar features might express a highlight in one video but not correspond to a highlight in another video, using context is inevitable to make good decisions about what plays are highlights for each game. By training the learning algorithm with similar videos to learn a overall highlight measure, plays in a new video can be classified to be a highlight or not.
Lucey et al. [2012] showed that tracks from players in soccer games can be used for predictions for the passing target of a player in ball possession. They define play-segments, which are spatio-temporal descriptions of ball movements and generate entropy maps for Premier League teams defining the teams behavior. In Bialkowski et al. [2013], where all players influence the teams activity - if only because of position information, a descriptor was designed that captures all player activities within a certain time period to discriminate the teams current activity. This paper is partly similar to the proposed work but lacks of the activity classification that is added to the position information.
Another example of exploiting player positions and movement directions is in McQueen et al. [2014]. They recognize on-ball screens in basketball by segmenting the video data into events (dribbles, shots, passes, fouls, ...) and using the ball and player positions to predict on-ball screen activities versus non-screen activities.

2.3 Datasets

According to Chaquet et al. [2013] there exist different levels of complexity for action and activity recognition datasets. Some of the most used datasets for evaluation of recognition methods are the KTH (Christian Schuldt and Caputo [2002]) and Weizmann (Zelnik-Manor and Irani [2001], Gorelick et al. [2005]) datasets. They provide indoor and outdoor videos with simple and static background and unrealistic human activities.
The next level of complexity contains 2 kind of realistic activity datasets, either specially recorded or collected from web videos. CAVIAR (EC Funded CAVIAR project/IST 2001 37540 [2002]), CASIA (Center for Biometrics and Security Research [2007]) and MSR (Yuan et al. [2009]) action datasets belong to the first group while Hollywood (Laptev [2008], Laptev [2009]), UCF Sports (University of Central Florida [2008]), UCF Youtube (University of Central Florida [2009]) and the HMBD51 datasets belong to the second. Datasets for interactions were presented in the BEHAVE (Fisher [2001]), TV Human Interaction (Visual Geometry Group [2010]) or UT Interaction (Ryoo and Aggarwal [2010]) datasets.
Finally, the most complex datasets are for multi-view analysis and consist of videos recorded indoor (IXMAS (INRIA [2006]), i3DPost Multi-view (University of Surrey and CERTH-ITI [2009]), MuHAVi (Kingston University [2010])) and outdoor (CASIA Action (Center for Biometrics and Security Research [2007]), VideoWeb (Video Computing Group. [2010])).

2.4 Industrial Software

While sport analysis of complex activities is still a topic of research, some commercial available software, mostly for tracking, already exists. The software applications are mainly used to augment sport broadcasts by displaying statistics or player/team movements. A industrial software system incorporating activity recognition in addition to tracking is not available at the time of this writing.

2.4.1 Hawkeye

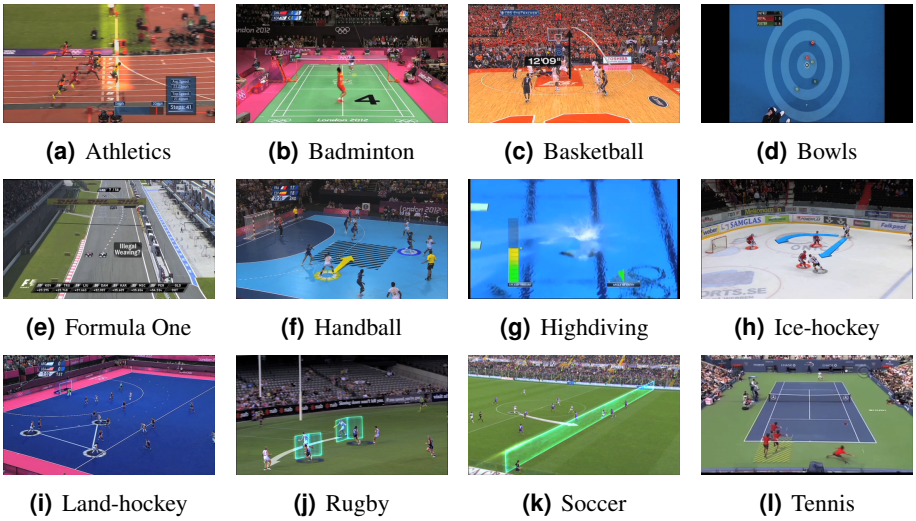
Hawkeye[1] uses multiple cameras to provide precise tracking in tennis, cricket, and other sports for refereeing and commentary. Referees are supported in their decisions, for example in tennis where players can object to the referees calls. For snooker, the system provides animations of shots or areas that are shielded from the cue ball by other balls. For cricket, different statistics are generated: From the ”wagon wheel” showing the batmans most used targets to pitch maps generated for a bowlers throws.

(a) Tennis: electronic line calling **(b)** Snooker: animated shots **(c)** Cricket: batsman targets

Figure 2.1: *Hawkeye Software examples (images taken from www.hawkeyeinnovations.co.uk)*

2.4.2 Red Bee Media

Red Bee Media[2] develops the *Piero* system for sports analysis and augmentation. The software is used for television broadcast presentation and analysis of several sports, mostly team sports. Although the software includes a tracking algorithm and supports calibration by learning the field borders and lines, the most of the marking and editing is done by a user manually.



(a) Athletics **(b)** Badminton **(c)** Basketball **(d)** Bowls

(e) Formula One **(f)** Handball **(g)** Highdiving **(h)** Ice-hockey

(i) Land-hockey **(j)** Rugby **(k)** Soccer **(l)** Tennis

Figure 2.2: *Piero software examples (all images from www.redbeemedia.com/piero)*

[1] http://www.hawkeyeinnovations.co.uk
[2] http://www.redbeemedia.com/piero

2.4.3 QuesTec

QuesTec[3] develops systems for tracking sports activities to provide enhanced broadcasts by reconstruction of 3D animations for replays and analysis. For baseball the software provides detailed information about pitching statistics, similarly for tennis service or winner statistics are generated while for golf a preview of a perfect putting stroke is displayed (see Figure 2.3).

(a) PitchTrax: statistics for baseball **(b)** TennisProView: statistics for tennis **(c)** GolfProView: put prediction for golf

Figure 2.3: *Questec software examples (all images from www.questec.com)*

2.4.4 Amisco/ProZone

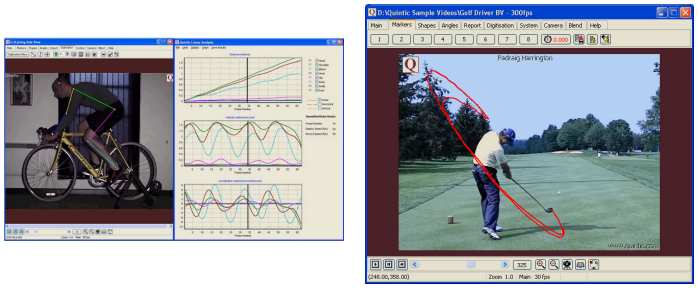
Amisco[4] builds systems for tracking sports players and the ball in real time, using some human assistance. *ProZone*[5] is a tracking system for soccer. It offers pre-match and post-match analysis, animation and statistics about fitness and performance. By using eight digital cameras, all players on the pitch can be tracked and detailed analysis achieved. The two companies have lately joined to combine their skills.

Figure 2.4: *Prozone software example (image from www.prozonesports.com)*

[3] http://www.questec.com
[4] http://www.sport-universal.com
[5] http://www.prozonesports.com/index.html

2.4.5 Quintic

Quintic[6] provides video analysis software for bio-mechanics, coaching, podiatry and education. It can be used in a broad range of sports: archery, athletics, bobsleigh, cricket, cycling, equine, football, golf, speed skating, rugby, squash, swimming. The software provides automated tracking and synchronized multi-camera setups.



(a) Cycling: analysis of velocities, rotations and accelerations **(b)** Golf: analysis of a golf drive

Figure 2.5: *Quintic software examples (all images from www.quintic.com)*

2.4.6 DataVideo

For volleyball, the most used software today is *DataVideo*[7], which is no computer vision system, but a statistics analysis system. A professional analyst has to enter every single activity of the player in form of special codes, while watching a live or recorded game. The activities occur in fast succession, so the analyst must be especially trained to distinguish the activities and assess the activity qualities. As even the best analysts cannot enter the information simultaneous with the activities, there is always a light delay which makes post game synchronization necessary. After codes and video are synchronized, the coach can easily retrieve player or team statistics with complex filtering options to find weaknesses of the opponents and strengths of his own team. With this information he can visually support his tactical strategies with presentations in team meetings.

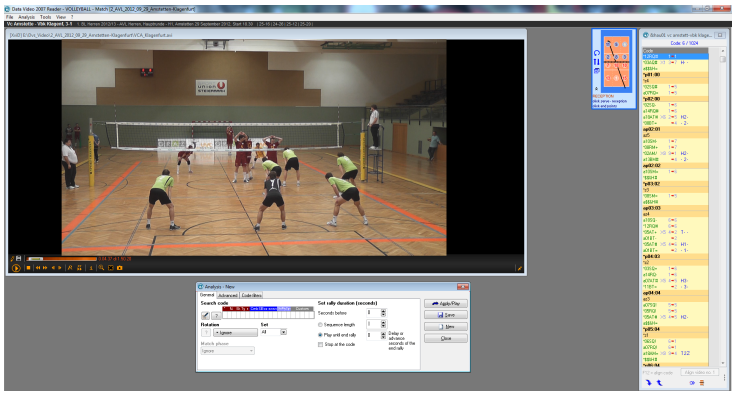


Figure 2.6: *DataVideo*

[6] http://www.quintic.com
[7] http://www.dataproject.com/Volleyball/DataVideo2007.aspx

Chapter 3

Theory

In this thesis multiple methods are used, this chapter gives an detailed overview for a understanding of the methods foundations. Influenced by many scientists and researchers, a conglomeration of theories make solving difficult computer vision tasks possible. The following chapter 4 presents methods specially designed for the subject of indoor activity recognition in volleyball, while this chapter introduces general methods that were developed by researchers and are successfully used in various activity recognition tasks.
The first part of this chapter is about image preprocessing, such preparing the video frames for further steps. Then different common descriptors are discussed, which try to extract visual information like motion or shape for discrimination of the activity classes, which is explained in the last part as classification task.

3.1 Image Preprocessing

Before the descriptors can efficiently extract information from the video frames, a preprocessing step is needed. Within this step, the videos are calibrated, color models for the background and the two teams are generated and player probabilities are generated via a bayesian-like model. The results help to find players in the frames and to remove background or other regions of no interest.

3.1.1 Transformation of Image Coordinates to Court Coordinates

By calibrating the videos a homography projection from image coordinates $\boldsymbol{x} = (x, y)$ to real-world court coordinates $\tilde{\boldsymbol{x}} = (\tilde{x}, \tilde{y})$ is obtained. The mapping of image coordinates to court coordinates via the transformation matrix $\boldsymbol{T}$ is needed for setting the proper scale for the bounding boxes around players used for feature extraction and for calculation of the context descriptors which rely on real-world positions of players on the court plane.
The correlation of the image and court plane is done by a representation called *homogenous notation*, where any 2D point $\boldsymbol{x}$ is extended to a homogenous vector (x, y, z) with $z = 1$: $(x, y, 1)$. Such, all points lie on the 'z-plane', which is required for the transformations. In other words, a plane is defined which all points (transformed and untransformed) share. In the context of the system, this plane is the volleyball court.
For a *projective transformation* the following transformation can be found:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} = \boldsymbol{T} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} \tag{3.1}$$

with

$$x = \frac{x'}{z'} = \frac{a_{11}\tilde{x} + a_{21}\tilde{x} + a_{31}}{a_{13}\tilde{x} + a_{23}\tilde{y} + a_{33}} \tag{3.2}$$

$$y = \frac{y'}{z'} = \frac{a_{12}\tilde{x} + a_{22}\tilde{y} + a_{32}}{a_{13}\tilde{x} + a_{23}\tilde{y} + a_{33}} \tag{3.3}$$

Inferring the transformation matrix $\boldsymbol{T}$
The 3x3 transformation matrix $\boldsymbol{T}$ consists of nine coefficients a_{ij}. By normalizing $\boldsymbol{T}$ such that $a_{33} = 1$, only eight coefficients are left, those can be calculated by the correspondences between (at least) four points from a reference image and processed video frames. Equation 3.2 can be rewritten as

$$x = \frac{a_{11}\tilde{x} + a_{21}\tilde{y} + a_{31}}{a_{13}\tilde{x} + a_{23}\tilde{y} + 1}$$

$$x(a_{13}\tilde{x} + a_{23}\tilde{y} + 1) = a_{11}\tilde{x} + a_{21}\tilde{y} + a_{31}$$

$$a_{13}\tilde{x}x + a_{23}\tilde{y}x + x = a_{11}\tilde{x} + a_{21}\tilde{y} + a_{31}$$

$$x = a_{11}\tilde{x} + a_{21}\tilde{y} + a_{31} - a_{13}\tilde{x}x - a_{23}\tilde{y}x \tag{3.4}$$

Rearranging Equation 3.3 for y results in

$$y = a_{12}\tilde{x} + a_{22}\tilde{y} + a_{32} - a_{13}\tilde{x}x - a_{23}\tilde{y}x \tag{3.5}$$

$[x_k, y_k]$ and $[\tilde{x}_k, \tilde{y}_k]$ with $k = 0, 1, 2, 3$ yield the following system of equations

$$\begin{bmatrix} \tilde{x}_0 & \tilde{y}_0 & 1 & 0 & 0 & 0 & -\tilde{x}_0x_0 & -\tilde{y}_0x_0 \\ \tilde{x}_1 & \tilde{y}_1 & 1 & 0 & 0 & 0 & -\tilde{x}_1x_1 & -\tilde{y}_1x_1 \\ \tilde{x}_2 & \tilde{y}_2 & 1 & 0 & 0 & 0 & -\tilde{x}_2x_2 & -\tilde{y}_2x_2 \\ \tilde{x}_3 & \tilde{y}_3 & 1 & 0 & 0 & 0 & -\tilde{x}_3x_3 & -\tilde{y}_3x_3 \\ 0 & 0 & 0 & \tilde{x}_0 & \tilde{y}_0 & 1 & -\tilde{x}_0y_0 & -\tilde{y}_0y_0 \\ 0 & 0 & 0 & \tilde{x}_1 & \tilde{y}_1 & 1 & -\tilde{x}_1y_1 & -\tilde{y}_1y_1 \\ 0 & 0 & 0 & \tilde{x}_2 & \tilde{y}_2 & 1 & -\tilde{x}_2y_2 & -\tilde{y}_2y_2 \\ 0 & 0 & 0 & \tilde{x}_3 & \tilde{y}_3 & 1 & -\tilde{x}_3y_3 & -\tilde{y}_3y_3 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{13} \\ a_{23} \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} \tag{3.6}$$

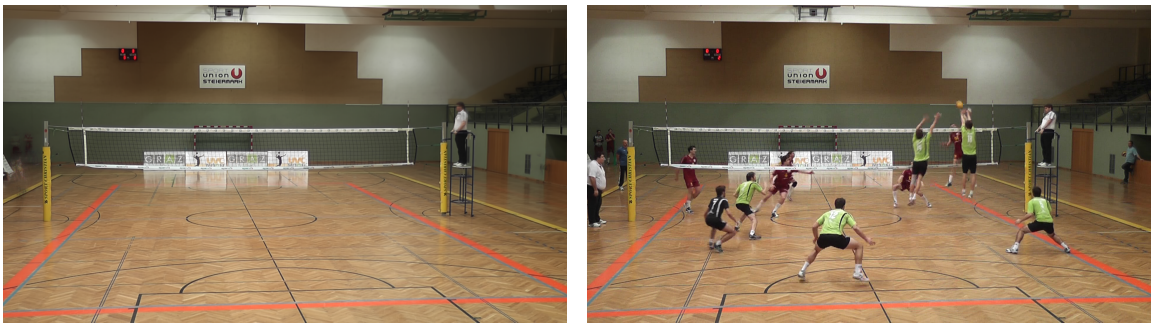
The coefficients $[a_{11}, a_{21}, a_{31}, a_{12}, a_{22}, a_{32}, a_{13}, a_{23}]$ of the transformation matrix $\boldsymbol{T}$ can be found by solving the above equations for the corresponding points $\boldsymbol{x_k}$ and $\tilde{\boldsymbol{x}}_{\boldsymbol{k}}$.

3.1.2 Modeling Foreground and Background

For the later extraction of players a model for foreground and background is needed. The background model is based on a number of video frames, that are filtered to remove foreground clutter and give a clean image of the background without players present. For the foreground, which means the players of the two teams, a gaussian mixture model (GMM) for each team is built.

Median Filtering

For automatically creating a clean background image, median filtering is used. Therefore multiple frames are chosen equally distributed over the whole video. Note, that within these video frames the appearance of players is no problem and no user interaction is needed. For each pixel the values of the selected frames are stored and the median is used as a representative value. The Median is the value that splits the sorted numbers into two halves of same size.
An example for the background extraction with 100 frames can be seen in Figure 3.1.



(a) background after median filtering over 100 frames **(b)** example frame used for filtering

Figure 3.1: *Median filtering for background modeling*

Gaussian Distribution

Foreground and background should be modeled with respect to color values, an estimation of the distribution and probability for each color pixel is needed. Gaussian distributions provide such information about the image pixels and can be calculated fast and efficient based on only few provided images (or image patches). The Gaussian Distribution g for a data vector $\mathbf{x}$ and a parameter set θ is a continuous probability distribution, defined by:

$$g(\mathbf{x}|\theta) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\mathbf{x}-\mu)^2}{2\sigma^2}} \tag{3.7}$$

The parameter set θ consists of the two variables μ (mean) and σ (standard deviation) respectively σ^2 (variance). The mean is the peak value of the gaussian distribution, the standard deviation defines the shape or width of the gaussian curve.

Gaussian Mixture Models (GMM)

Foreground models for the two teams are built by manual selection in some frames of the video. For each of the teams the foreground and background are defined via rectangles, where the background can and should also contain the other teams players for better distinction between the teams. As a preprocessing step, the images are filtered to result in better separations.
From the previously defined rectangles all pixels are extracted and a Gaussian Mixture Model (GMM) for each team is fitted separately to this data. The GMM is a joint function f of k Gaussian distributions:

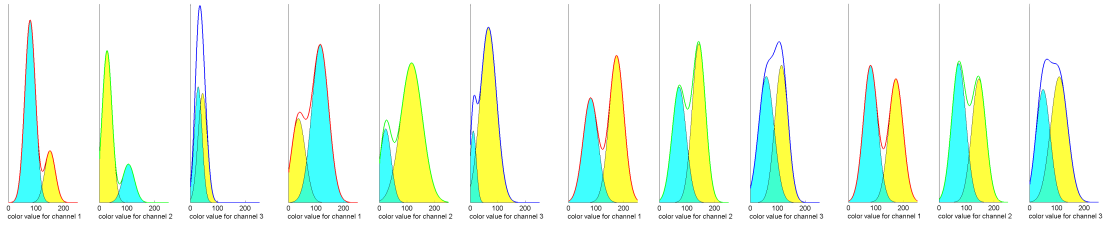
$$f(\mathbf{x}|\Theta) = \sum_{k=1}^{K} p_k \, g(\mathbf{x}|\theta_k) = \sum_{k=1}^{K} p_k \, g(\mathbf{x}|\mu_k, \sigma_k) \tag{3.8}$$

where p_k are *mixing probabilities* fulfilling two constraints, they

1. are non-negative $p_k \geq 0$
2. sum up to one $\sum_{k=1}^{N} p_k = 1$

$\mathbf{x}$ is a D-dimensional, continuous data vector. $\boldsymbol{\Theta}$ contains the parameters of the Gaussian distributions. Finding the model parameters of the distribution is done by using the expectation maximization algorithm.

Gaussian distributions for background and foreground models of one of the used videos are displayed in Figure 3.2. There are two background models, as for each team the other team is also considered background. When modeling the foreground for one team, the user chooses image patches with and without the team. Optimally the chosen background patches also include players of the opponent team, making discrimination between players easier.



(a) foreground GMM built from team 1 patches **(b)** foreground GMM built from team 2 patches **(c)** background GMM built from team 1 patches **(d)** background GMM built from team 2 patches

Figure 3.2: *GMMs for foreground and background, each color channel separately displayed (red/green/blue hulls). Note the difference between the foreground models of the two teams, while the background models are quite similar as one would expect.*

Expectation Maximization (EM) Algorithm

The EM algorithm is an iterative method to fit a statistical model to the given data, assuming the data is generated by a certain probability distribution. The aim is to find the model parameters which maximize the likelihood of the GMM for the data provided. In the *expectation step*, the association of the data points with the model are improved, whereas in the *maximization step* the model is adapted to fit the data better. As an initialization, the unknown parameters sets are guessed. In case of the GMM, these k parameter sets $\hat{\theta} = (\hat{\theta_1}, \cdots, \hat{\theta_k})$ are the mixing probability $\hat{p_k}$ and the variables of the n Gaussian distributions: $\hat{\theta_k} = \hat{\mu_k}, \hat{\sigma_k}$. With $\mathbf{x} = (x_1, \cdots, x_i)$ as the available i data vectors, the likelihood function Λ can be defined as follows:

$$\Lambda(\mathbf{x}|\hat{\theta}) = \prod_{n=1}^{N} f(x_n|\hat{\theta}) = \prod_{n=1}^{N} \sum_{k=1}^{K} p_k \, g(x_n|\hat{\mu_k}, \hat{\sigma_k}) \tag{3.9}$$

To simplify solving the equation, often the log-likelihood estimation is used. The maxima stay the same after applying the logarithm:

$$\lambda(\mathbf{x}|\hat{\theta}) = \sum_{n=1}^{N} log \sum_{k=1}^{K} p_k \, g(x_n|\hat{\mu_k}, \hat{\sigma_k}) \tag{3.10}$$

The maximum likelihood estimate for the parameters is found with:

$$\theta = \arg\max_{\theta} \Lambda(\mathbf{x}|\hat{\theta}) \;\; and \;\; \theta = \arg\max_{\theta} \lambda(\mathbf{x}|\hat{\theta}) \tag{3.11}$$

The basic idea of the EM algorithm is, beginning with an initial model $\hat{\theta}$, to estimate a new model θ, such that $\lambda(\mathbf{x}|\theta) \geq \lambda(\mathbf{x}|\hat{\theta})$. The new model then becomes the initial model for the next iteration and the process is repeated until some convergence threshold ϵ is reached: $\left|\lambda(\mathbf{x}|\theta) - \lambda(\mathbf{x}|\hat{\theta})\right| \leq \epsilon$. To find the local maxima of λ (or Λ) the derivatives of with respect to μ, σ, p_k are computed (Bishop [1995]).

Expectation step: The mixing probabilities p_k from Equation 3.7 are used to calculate the *a posteriori probabilities* Pr, which are the conditional probabilities of a data vector $\mathbf{x}$ yielding the component k in the model:

$$Pr(k|\mathbf{x},\theta) = \frac{p_i g(x_i|\mu_i,\sigma_i)}{\sum_{j=1}^{M} p_j g(x_j|\mu_j,\sigma_j)} \tag{3.12}$$

Maximization step: With the a posteriori probabilities from the E-step, the new weighted means and standard deviations (D being the dimensionality of the data points) can be computed (Tomasi [2004]):

$$p_k = \frac{1}{N}\sum_{t=1}^{T} Pr(k|\mathbf{x},\theta) \tag{3.13}$$

$$\mu_k = \frac{\sum_{t=1}^{T} Pr(k|\mathbf{x},\theta)\, x_t}{\sum_{t=1}^{T} Pr(k|\mathbf{x},\theta)} \tag{3.14}$$

$$\sigma_k = \sqrt{\frac{1}{D}\frac{\sum_{t=1}^{T} Pr(k|\mathbf{x},\theta)\ |x_t-\mu_k|^2}{\sum_{t=1}^{T} Pr(k|\mathbf{x},\theta)}} \tag{3.15}$$

3.1.3 Bayesian Model for Player Blob Extraction

Under the assumption that the probabilities for foreground, background and players can be expressed in terms of probabilities, a simple Bayesian model can be composed.

Bayes' theorem

The Bayes' theorem describes the relation between probabilities and results in a statistical model for the posterior probability. Given two observations A and B, the *a-priori* probabilities or *priors* for the observations are $P(A)$ and $P(B)$. By linking the two observations, one can describe so called *conditional probabilities* or *likelihood* of A and B ($P(A|B)$, $P(B|A)$). For example, $P(B|A)$ is the probability for observing B given observation A.
The *joint* probability of observing both, A and B is

$$P(A,B) = P(A|B)P(B) = P(B|A)P(A)$$

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{3.16}$$

In the second equation, $P(A)$ and $P(B)$ are *a-priori probabilities* and $P(B|A)$ the *likelihood*. $P(B)$ is used as a normalization term. The result in form of $P(A|B)$ is the *a-posteriori probability*.
$P(B)$ can also be expressed as a sum of possible outcomes giving

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_j P(B|A_j)P(A_j)} \tag{3.17}$$

Duda et al. [2001] put the problem in simple terms:

$$posterior = \frac{likelihood \times prior}{evidence} \tag{3.18}$$

Player Model

The *posterior* probability P_{player} is the desired result of the player model (Figure 3.3(f)). It describes the probability of a pixel being part of a player of the front team. For the foreground and background probabilities, Gaussian Mixture Models (GMM) are trained from front team patches one the one and back team and background patches on the other hand. The *prior* probabilities for foreground (P_{fg}) and background (P_{bg}) are taken from the probability density function of these GMMs (Figures 3.3(d), 3.3(e)). Summing up the two *prior* probabilities yields the *evidence* term or *normalization* term. For any pixel $\mathbf{x}$ a color similarity measurement M_{dyn} describing non-static objects (players, referees, ball, moving net, ...) is calculated from absolute differences between a median filtered background model BG (Figure 3.3(a)) and the current frame F at time i (Figure 3.3(b)).

$$M_{dyn}(\boldsymbol{x}) = \max_{c \in \{r,g,b\}} |BG^c(\boldsymbol{x}) - F_i^c(\boldsymbol{x})| \tag{3.19}$$

The result is an image with blobs of the players (see Figure 3.3(f)).

$$P_{player}(\boldsymbol{x}) = \frac{P_{fg}(\boldsymbol{x})\, M_{dyn}(\boldsymbol{x})}{P_{fg}(\boldsymbol{x}) + P_{bg}(\boldsymbol{x})} \tag{3.20}$$

This resulting player probability image is later used for the descriptors. The SC descriptor uses if for a non player specific probability estimate for the player positions and on court distribution, while for the AC descriptor the player positions required for player activity classification are found via blob extraction from P_{player}.

3.1.4 Morphological Operations

After applying the Bayes' theorem some morphological operations are needed to get the desired result: clean player blobs. To remove noise and connect areas the image is morphologically closed, which means first dilated and then eroded.

Dilation

The dilation of image A by a structuring element B is the set of all displacements.

$$A \oplus B = \bigcup_{b \in B} A_b \tag{3.21}$$

As result, the original image A is enlarged around the edges, the image becomes bigger and gaps smaller than the structuring element B are closed. B can be one of different shapes, typical forms are diamond, disk, rectangle or square. Also arbitrary shapes are possible.

Erosion

The erosion of image A by a structuring element B is the set of all displacements.

$$A \ominus B = \bigcap_{b \in B} A_{-b} \tag{3.22}$$

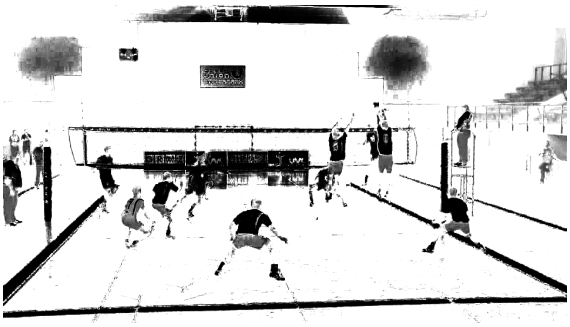
As result, the original image A is shrunk around the edges, the image becomes smaller and gaps are widened. As with dilation, B can have arbitrary shape.

(a) median filtered background

(b) processed frame (F)

(c) dynamic difference measurement (M_{dyn})

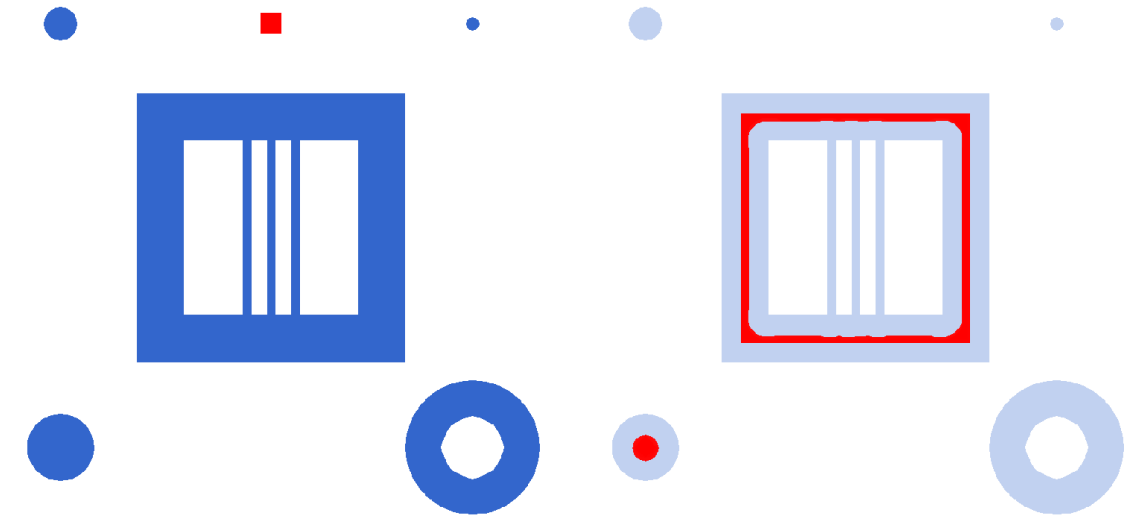
(d) foreground probability for the team with the green jerseys (P_{fg})



(e) background probability (P_{bg}) including opponents

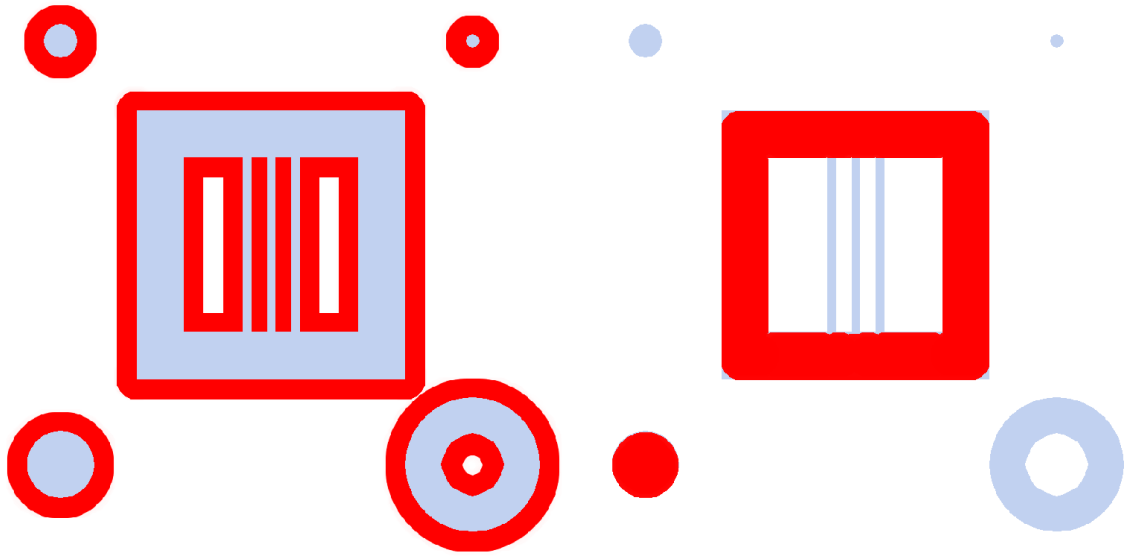
(f) result (P_{player}) for the team with the green jerseys (after morphological closing)

Figure 3.3: *Application of Bayes' theorem for extraction of player blobs. The processed frame is shown in 3.3(b). Using only the dynamic difference measurement 3.3(c) introduces noise from shadows, net, noninvolved persons (referee, coach) and advertising boards. On the other hand, the foreground probability for the players also introduces noise, as the wall in the back has a similar color (green) as the jerseys of the players. Combining the two and adding the background probability 3.3(e), the result 3.3(f) is apparently better.*



(a) example shapes, the red rectangle on top ist the structuring element

(b) dilation of (a) with original shapes in blue and resulting shapes in red



(c) erosion of (a) with original shapes in blue and resulting shapes in red

(d) closing of (a) with original shapes in blue and resulting shapes in red

Figure 3.4: *Example of morphological operations. (a) shows some example shapes in light blue and the structuring element in red. (b) shows the result in red after dilation. (c) shows the result in red after erosion. (d) shows the result in red after closing. After closing, lines and circles smaller than twice the size of the structuring element have disappeared while larger shapes are preserved.*

Closing

Closing is an important operator from the field of mathematical morphology. It consists of the two previously discussed operations of erosion and dilation. First the image is dilated and then eroded. This preserves regions with similar shape as the structuring element (or also larger areas) while removing small artifacts and noise. Thanks to the erosion, the initial shapes are kept in similar form. Closing of an image is defined by

$$A \bullet B = (A \oplus B) \ominus B \tag{3.23}$$

As showed in Figure 3.3(f) in the final player probability image much noise is removed. Morphological closing helps removing net meshes or lines and connects homogenous player regions.

3.2 Descriptors

Descriptors are designed to find the best or at least a good description of objects. Therefore, as the description depends on the domain of the objects, knowledge about the domain is vital for finding a good descriptor. Clearly descriptors are often very specific and only of use under certain circumstances or in certain domains. Still, some descriptors in computer vision research have proven to be successful over multiple domains and problems. For action and activity recognition, the *Histograms of Oriented Gradients (HOG)* and the *Histogram of Flow Orientation (HOF)* are state-of-the-art methods. The following sections describe these general descriptors, while Chapter 4 will provide information about special designed descriptors for this subject, incorporating context.

3.2.1 Histograms of Oriented Gradients

Gradients have been widely used to build descriptors for the shape of objects. The Histograms Of Gradients (HOG) descriptor was introduced by Dalal and Triggs [2005] for human detection from images. Prior to the work of Dalal and Triggs [2005], the equally popular Scale Invariant Feature Transform (SIFT) (Lowe [2004]) emphasized the use of gradients for object description. Both methods are still present in today's object detection and recognition tasks.
By exploiting the distribution of edge directions and strengths, the descriptor describes local object appearance without the need to know the exact position of the object itself. Therefore the investigated image or image part is subdivided into smaller regions (cells). The edge information within these cells is binned into histograms of n bins from $0°$ to $180°$. The resulting cell histograms are combined into blocks, each consisting of multiple adjacent cells. For building the descriptor, one cell is multiply used in different blocks accounting for illumination or contrast changes by normalizing the blocks.
For the calculation of the HOG descriptor vector, first all edges are extracted by computation of the gradients ∇x and ∇y with gradient filter masks.

$$\nabla x : \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \qquad \nabla y : \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \tag{3.24}$$

Then, magnitude m and orientation φ are computed.

$$m(\mathbf{x}) = \sqrt{\nabla_x \mathbf{x}^2 + \nabla_y \mathbf{x}^2} \tag{3.25}$$

$$\varphi(\mathbf{x}) = atan\left(\frac{\nabla_y \mathbf{x}}{\nabla_x \mathbf{x}}\right) \tag{3.26}$$

Next, the orientations are quantized into bins within 0° to 180°, then collected within one histogram per cell, weighted by their magnitudes. Finally, the block histograms are normalized over multiple cells. The resulting normalized block histograms are concatenated resulting in the feature vector. See Figure 3.5.

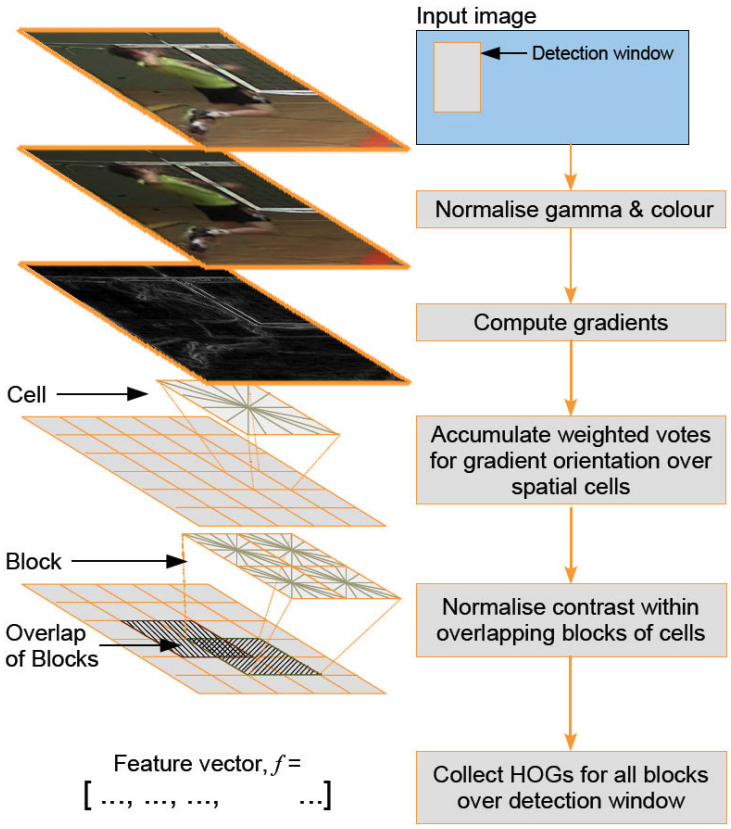


Figure 3.5: *HOG descriptor computation. Image taken from Dalal [2006] and altered.*

The HOG descriptor has already been used for activity recognition in sport. For example, Lu and Little [2006] used a PCA-HOG descriptor for tracking and classification players in ice-hockey and soccer.

3.2.2 Optical Flow

Optical flow is a measure for movement between two images I_t and I_{t+1}, used to determine where, in which direction and with which strength movements occur over time.

$$I_t(x, y) = I_{t+1}(x + u_x, y + u_y) \tag{3.27}$$

Approximation by a Taylor series yields

$$I_t(x, y) \approx I_{t+1}(x, y) + u_x \frac{dI_{t+1}}{dx} + u_y \frac{dI_{t+1}}{dy} \tag{3.28}$$

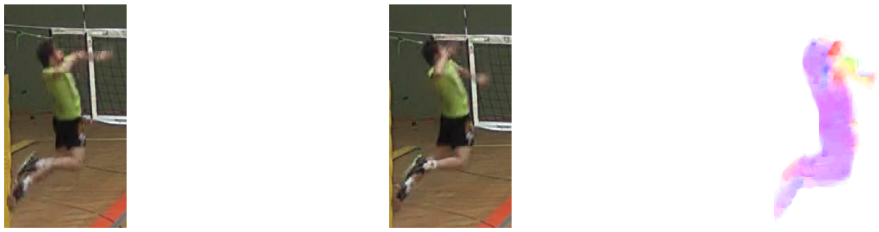
$$u_x \frac{dI_{t+1}}{dx} + u_y \frac{dI_{t+1}}{dy} \approx 0 \tag{3.29}$$

In these equations I_t and I_{t+1} are the images, u_x and u_y the displacement fields. This last Equation 3.29 is referred to as the *optical flow constraint*, as it assumes the image intensities stay the same

during motion (displacement by $\mathbf{u} = (u_x, u_y)$). This leads to the aperture problem, describing the problem of assignment of pixels that can have moved in different ways yielding in the same image result. For example, the movement parallel to an edge can not be determined. Using only intensity as constraints makes the problem ill-posed, as it leads to an under-determined set of equations. To help with this problem a regularization term is introduced. Horn and Schunck Horn and Schunck [1981] presented a variational model for optical flow:

$$\min_u \{ \int_\Omega \lambda \left(|\nabla u_x|^2 + |\nabla u_y|^2 \right) d\Omega + \int_\Omega (I_{t+1}(\mathbf{x} + \mathbf{u}) - I_t(\mathbf{x}))^2 d\Omega \} \tag{3.30}$$

The first part is for regulation of the smoothness of the displacement fields (*regularization term*). Partial derivatives of each flow component are enforced to be small. The second part of Equation 3.30 is called the *data term*, as introduced before. λ is a free cost parameter, a larger value results in a smoother flow. A disadvantage of the Horn-Schunck-model is the sensitiveness to noise and problems when discontinuities in the displacement field occur. This is due to the quadratic penalization. To overcome these limitations, various methods have been proposed: Regularization can be image-driven or flow-driven, homogenous or in-homogenous, or isotropic or non-isotropic. The common target is to preserve edges while smoothing homogenous regions. For details refer to Beauchemin and Barron [1995], Weickert and Schnörr [2001], Bruhn et al. [2005].



(a) image at time t **(b)** image at time $t + 1$ **(c)** flow result

Figure 3.6: *Illustration of flow as the difference in motion between an image at time t and an image at time $t + 1$. Note that the image is the color coded result of the flow motions in two directions.*

3.2.3 Histograms of Oriented Flow

The Histograms of Oriented Flow descriptor (HOF) is based on the idea of the Histograms of Oriented Gradient (HOG) descriptor. The flow image is subdivided into cells. But instead of gradient directions, the flow directions are collected into normalized histograms (Laptev et al. [2008]). The combination of HOG and HOF features has become a standard approach for human activity recognition (e.g. Ikizler et al. [2008], Mauthner et al. [2010], Raptis and Soatto [2010]).

3.3 Classification

3.3.1 Support Vector Machines

A widely used method in supervised learning are Support Vector Machines (SVM). SVM is an example for inductive learning, which means that samples (training data) are used to find a general prediction rule that should classify new samples (test data) in an optimal way.

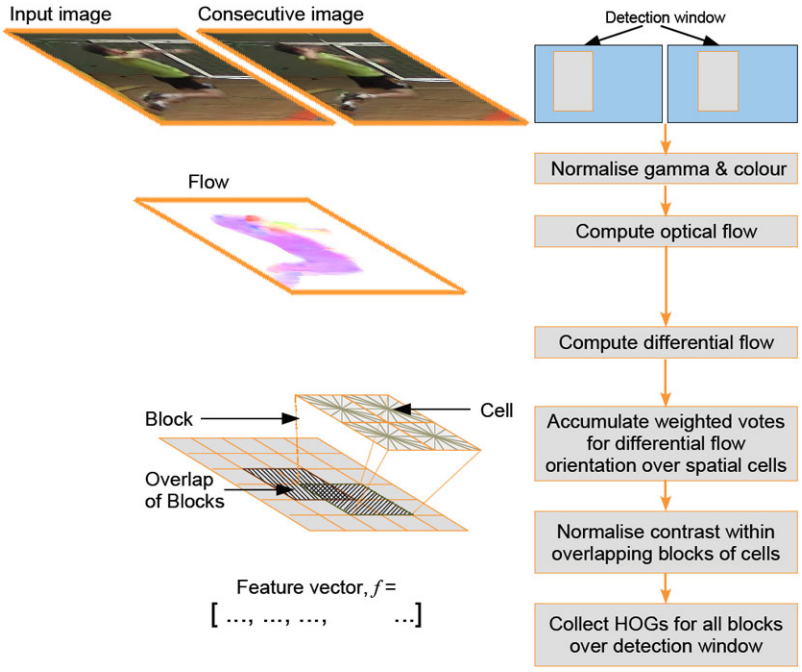


Figure 3.7: *HOF descriptor computation. Image taken from Dalal [2006] and altered.*

A mapping $\mathbf{X} \rightarrow \mathbf{Y}$ should be learned, where $\mathbf{x} \in \mathbf{X}$ is some object (respectively a n dimensional data vector) and $y \in \mathbf{Y}$ is a class label. A good generalization is needed, meaning to map the corresponding classes correctly for 'new' - meaning not previously used - examples. For good results often many examples are needed to make the prediction usable. Basically: The more training examples, the better the result (up to a certain degree). A classifier $y = f(\mathbf{x}, \alpha)$ is sought, where α is the collection of parameters of this classifier. If the classifier is chosen from the set of hyperplanes in $\mathbf{R}^n$, this results in:

$$f(\mathbf{x}, \{\mathbf{w}, b\}) = sign(\mathbf{w} \cdot \mathbf{x} + b) \tag{3.31}$$

Where $\mathbf{w}$ is the normal vector to the hyperplane from a given point, while b (or more precisely $\frac{b}{\|\mathbf{w}\|}$) defines its distance to the hyperplane. Given examples as pairs consisting of a vector (filled with pixels, observations, features,...) $\mathbf{x}_i \in \mathbf{R}, (i = 1 \ldots l)$ and a associated class $y_i \in \{-1, 1\}$ the following equations for all points can be found:

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq 1 \; for \; y_i = 1 \tag{3.32}$$

$$\mathbf{x}_i \cdot \mathbf{w} + b \leq 1 \; for \; y_i = -1 \tag{3.33}$$

These equations can be visualized as shown in Figure 3.8. It is easily to see, that only points satisfying the Equations 3.32 and 3.33 lie on and influence the two parallel hyperplanes (dotted lines in Figure 3.8) - they are called *support vectors*. Removing points that are not on these hyperplanes do not alter the solution found. The two equations from above can be combined into one set of inequalities:

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0 \; \forall i \tag{3.34}$$

The optimal separating hyperplane is the one with the largest margin. It separates the data without error and the distance between the closest data points is maximal. The distance is given by

$$\underset{\{\mathbf{x}_i | y_i = 1\}}{max} \frac{\mathbf{x}_i \cdot \mathbf{w} + b}{\|\mathbf{w}\|} - \underset{\{\mathbf{x}_i | y_i = -1\}}{max} \frac{\mathbf{x}_i \cdot \mathbf{w} + b}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|} \tag{3.35}$$

Thus the maximization of the margin $\frac{2}{\|\mathbf{w}\|}$ is found by minimizing $\|\mathbf{w}\|$.

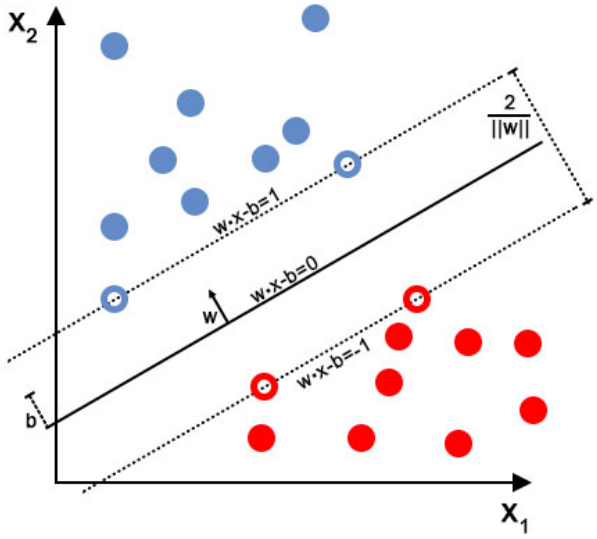


Figure 3.8: *Separation of two classes (red and blue dots) by two parallel hyperplanes. In the middle lies the separating hyperplane which has the margin* $\frac{2}{\|\mathbf{w}\|}$ *to the support vectors (circles).*

The Kernel Trick

In general, data is almost never linearly separable. This makes finding a good decision boundary with small error impossible. To overcome the limitation of linear separation planes, the data can be transferred into a feature space of higher dimensionality and a new linear decision function as a mapping between this transfer function and the target classes is learned.

$$\mathbf{x} \mapsto \Phi(\mathbf{x}) \quad \Rightarrow \quad y = \mathbf{w} \cdot \Phi(\mathbf{x}) + b \tag{3.36}$$

The problem with this solution is, a high dimensionality of the transfer function $\Phi(\mathbf{x})$ makes holding $\mathbf{w}$ in memory and solving the problem impossible.
With the *Representer Theorem* from Kimeldorf and Wahba [1971], for SVMs the following is valid:

$$\mathbf{w} = \sum_{i}^{n} \alpha_i y_i \Phi(\mathbf{x}_i) \tag{3.37}$$

Instead of optimizing $\mathbf{w}$, optimizing α gives the same result. The decision rule becomes:

$$f(\mathbf{x}) = \sum_{i}^{n} \alpha_i y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}) + b \tag{3.38}$$

$\mathbf{x_i}$ are the n support vectors, $\mathbf{x}$ are the data vectors. By introducing a "kernel function" K, the expensive high dimensional computation of the dot products of $\Phi(\mathbf{x_i}) \cdot \Phi(\mathbf{x})$ can be avoided:

$$f(\mathbf{x}) = \sum_{i}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{3.39}$$

Some popular kernel functions are listed below.
Linear Kernel
The linear kernel function is the simplest kernel function and is denoted by the inner product of the data.

$$K(\mathbf{a}, \mathbf{b}) = < \mathbf{a}, \mathbf{b} > = \mathbf{a} \cdot \mathbf{b}$$

Polynomial Kernel
The polynomial kernel is a non-stationary kernel, best suited for normalized data.

$$K(\mathbf{a}, \mathbf{b}) = (\gamma < \mathbf{a}, \mathbf{b} >)^d$$

Radial Basis Function (RBF) kernel
The RBF kernel is based on a gaussian distribution and adds a elevation around each data point with γ determining the width of the Gaussian bell. It is widely used, as it operates like a low bass filter and such delivers smooth results.

$$K(\mathbf{a}, \mathbf{b}) = e^{-\gamma(\frac{\|\mathbf{a}-\mathbf{b}\|^2}{2\sigma^2})}$$

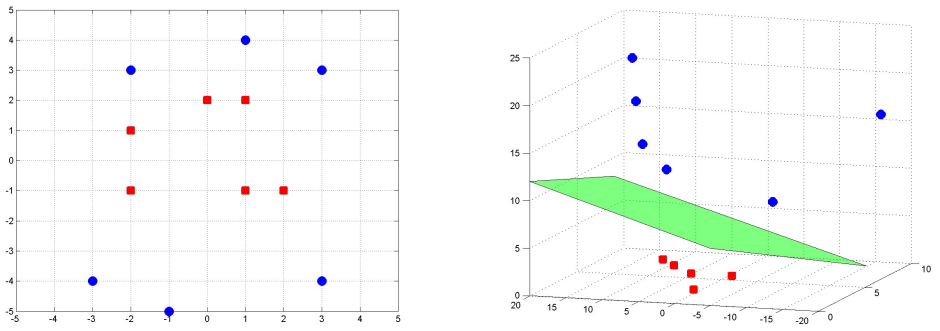
Sigmoid Kernel
The sigmoid kernel, also known as hyperbolic Tangent kernel or multilayer perceptron kernel, comes from the field of neural networks where the bipolar sigmoid function is used to describe the activation of neurons.

$$K(\mathbf{a}, \mathbf{b}) = tanh(\alpha x^T b + c)$$

χ^2-Kernel
As can be guessed by the name, the chi-squared kernel is derived from the χ^2 distribution.

$$K(\mathbf{a}, \mathbf{b}) = 1 - 2\sum_{i=1}^{n} \frac{(a_i - b_i)^2}{(a_i - b_i)}$$

Many more kernels exist, for further information see Shawe-Taylor and Cristianini [2004]. As an



(a) data points in 2D space, not linearly separable **(b)** feature points (data points transferred to 3D space), linearly separable by a hyperplane (green)

Figure 3.9: *SVM kernel trick: The red and blue points are not linearly separable in 2D space, but a simple transformation into a higher feature space (3D) solves the separation problem.*

example some points in 2D might not be separable, but a transformation into 3D space makes the data separable (see Figure 3.9). Consider the mapping $\Phi(\mathbf{x}) : \mathbf{x} = (x_1, x_2) \mapsto (x_1^2, \sqrt{2}x_1x_2, x_2^2)$ and a kernel function $K(\mathbf{a}, \mathbf{b}) =< \mathbf{a}, \mathbf{b} >^2$, the result is:

$$\begin{aligned} K(\mathbf{a}, \mathbf{b}) &=< \mathbf{a}, \mathbf{b} >^2 = (a_1b_1 + a_2b_2)^2 = a_1^2b_1^2 + 2a_1b_2 + a_2^2b_2^2 \\ &= (a_1^2, \sqrt{2}a_1a_2, a_2^2)(b_1^2, \sqrt{2}b_1b_2, b_2^2)^T =< \Phi(\mathbf{a}), \Phi(\mathbf{b}) > \end{aligned}$$

Obviously the dot product of $\Phi(\mathbf{a})$ and $\Phi(\mathbf{b})$ can be caluclated without explicitly using Φ. The square of $\mathbf{a}$ and $\mathbf{b}$ (=kernel function) is sufficient. In other words: A nonlinear separation of data using a kernel function in a (mostly low dimensional) data space is equally to a linear separation using the dot product in a (higher dimensional) feature space.

Chapter 4

Purpose-Built Methods

Every computer vision system demands for adapted methods. Besides the features described in the previous Section 3.2, in this chapter three additional descriptors are proposed to model position and activity relations amongst players. These player position and activity relations equate to context descriptions of game situations where the team as whole tries to succeed and every player accomplishes specific tasks at a corresponding position. The three descriptors are motivated by the inherent game structure in Volleyball that has repetitive activity patterns. Every player has assigned tasks depending on the situation of the play and these situations often reoccur in a similar manner (reception formations, block/defense formations, ...). Defense or offense activities in volleyball are always performed by the whole team, not only one player. For example, if one player attacks the others try to spread within the court such that a potentially blocked ball can be recovered. Similarly in defense situations some players may block while others try to defend the rest of the court not covered by the blockers. The first descriptor uses the position of a single player as clue for classification of his specific activity. The second descriptor calculates the player distribution of all players on and around the court to help recognizing one players activity. Finally, the third descriptor uses information about the locations and activities of all players as summarized activity description of all players over a period of time.
After termination of a rally and before the next rally begins, players need to take defined positions within the court. As soon as the ball is in play, the players move to their designated positions and fulfill their assigned roles (attacker, blocker, receiver, setter, libero). While every player takes every of the six positions in the court during the game process (rotating in the court, see Figure 5.1), when the ball is brought into play every player moves repeatedly to the very same position. These positions are connected to execution of specific activities. To make use of this structure for recognition, the **Real World Player Coordinates (RWPC)** descriptor is introduced, consisting of two dimensional court coordinates estimated from the image plane. This descriptor is a simple exploitation of the players positions in and around the court. The real world player positions give hints about the performed activities. The **Spatial Context (SC)** descriptor makes use of all players positions in the field while a certain activity is performed by one player. By doing a dense grid search for foreground occupancy throughout the court, the relation of the performing player and the other players is evaluated resulting in a team distribution description. This is motivated by the fact, that not only the player in contact with the ball is executing activities, but also the other team members need to fulfill their role by supporting this player. While the SC descriptor only evaluates player distributions as foreground probabilities without investigating the occurring player activities, the **Activity Context (AC)** descriptor uses the results of the activity classification to determine the activities of all players. Additionally, the evaluation is done over a number of frames to incorporate the temporal dimension into the descriptor. The result is a collective description of the type and location of all player activities over time. Therefore all players are first searched for via blob extraction and then the executed player activities are classified.

This is repeated for a certain time span preceding the tested frame. Combining the information about where and when all players on court perform which activities within this time span yields to the AC descriptor.

4.1 Real World Player Coordinates (RWPC)

The player positions $\boldsymbol{x}$ in the image plane, inferred from projections of player blobs (resulting from the Bayesian model, see Section 3.1.3), are transformed to real world positions $\tilde{\boldsymbol{x}} = (\tilde{x}, \tilde{y})$ on the ground plane (see Section 3.1.1) and then normalized.

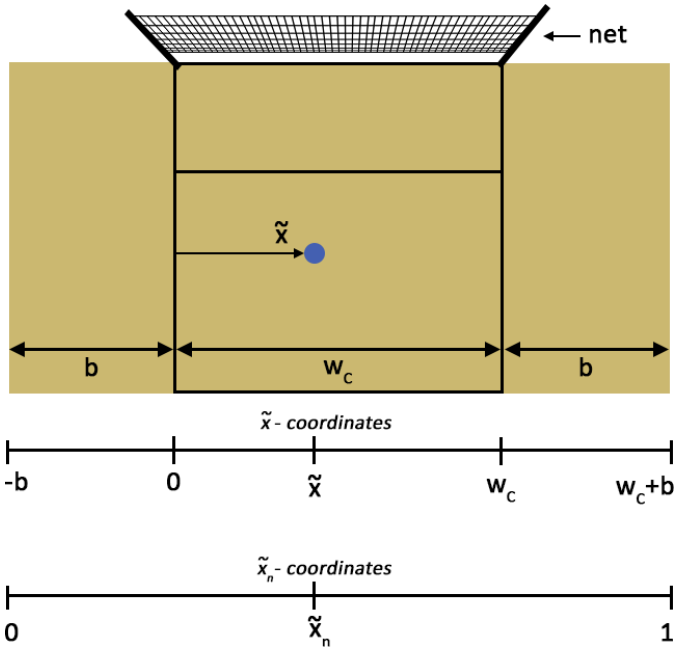


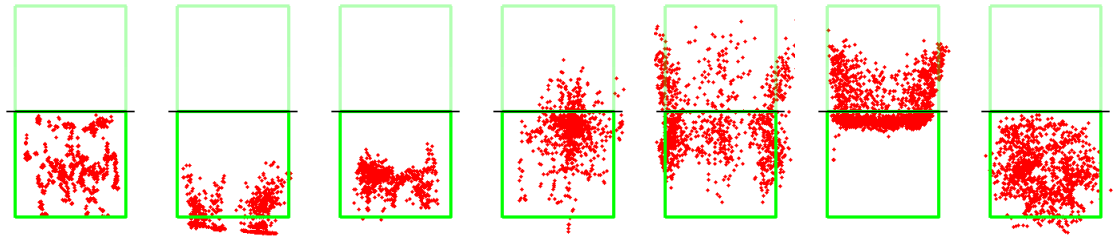
Figure 4.1: *Extension of the court for normalization of player positions (example for $\tilde{x}$ coordinates). The blue point marks a player on the court, positioned at $\tilde{x}$ from the left border of the court. After normalization, $\tilde{x}_n$ is between 0 and 1, around 0.45 for this example. The first scale under the illustration indicates the $\tilde{x}$-coordinate scale while the second scale denotes the $\tilde{x}_n$-coordinates. For reasons of clarity, the $\tilde{y}$ coordinate is not visualized.*

To account for players outside the court, the court of width w_c is extended by adding a specific area to the left, right, top and back of the court which will be referred to as border b. The $\tilde{x}$ coordinates are then normalized by adding b to any position $\tilde{x}$ in court and subsequently dividing by the total extended width $(b + w_c + b)$. As a result the values range between 0 and 1, where 0 denotes the $\tilde{x}$ coordinate on the left edge of the border and 1 denotes the $\tilde{x}$ coordinate on the right outside border of the extended court. Normalization is similar for the $\tilde{y}$ coordinates, using the court height (same as court width, as the volleyball court is a square) and a vertical border extension.

$$\tilde{x}_n = \frac{\tilde{x} + b}{b + w_c + b} = \frac{\tilde{x} + b}{2 \times b + w_c} \qquad \tilde{x}_n \in \{0, 1\} \tag{4.1}$$

Figure 4.2 shows the distribution of the different activities listed in detail in Table 5.2. Note, that due to the planar homography jumping players appear to be on the opponents court. This is not resolvable without added 3D information. A jump is therefore equally to a player moving very fast into and

back out of the opponents court, as the player position is mapped to the location of his feet. To collect all player activities, it is necessary to make the border as wide as needed, especially in direction of the opponent. The figure shows the motive of this descriptor, as due to the game immanent structure the activities take place on specific areas in the court. For example, while players might jump into the court during service execution, all service activities start from behind the court. Also, blocking activities always occur at the net, although jumping players seem to move into the other court due to the transformation to the court plane. The classes stand, defense and move are spread on the court. (That circumstance and the similarity of the two classes move and defense were part of the motivation behind merging them together into one class.)



(a) stand **(b)** service **(c)** reception **(d)** setting **(e)** attack **(f)** block **(g)** defense/move

Figure 4.2: *Position of players during specific activities: Courts are marked in green, positions are shown as red dots. The black line indicates the net.*

While for the RWPC descriptor the transformed coordinates are directly corresponding to the features, for the Spatial Context (SC) and Activity Context (AC) descriptors these coordinates are binned into groups for dimensionality reduction of the features. Therefore a grid of b_x bins in x direction and b_y bins in y direction is distributed over the analyzed court area. Each of these subareas Λ_{i_x,i_y} contains multiple sampled points, the number of points within the subareas depends on the number of bins b_x and b_y and on the sampling density. With the normalized player position $\tilde{\boldsymbol{x}}_{\boldsymbol{n}} = (\tilde{x}_n, \tilde{y}_n)$ and the bin parameters b_x and b_y, the corresponding bin indices for the descriptors can be calculated.

$$i_x = \lfloor \tilde{x}_n * b_x \rfloor, \quad i_y = \lfloor \tilde{y}_n * b_y \rfloor \tag{4.2}$$

4.2 Spatial Context Descriptor (SC)

The proposed SC descriptor is basically describing the on court distribution of all players during execution of a player activity. Player positions are estimated by using the foreground probabilities at dense sampled points. For dimensionality reduction, these probabilities are averaged within subareas of the court. First the player probability image P_{player} (see Equation 3.20) is used as input image. A dense grid of equally distributed positions is laid on the court plane. To normalize for different scaling artifacts of the perspective view, the grid is warped according to the underlying homography. As a result the sampling density in the image plane becomes denser with increasing distance from the camera (see green grid lines in Figures 4.4(a) and 4.4(b)).
The coordinates of the grid points $\tilde{\boldsymbol{x}}$ on the court are transformed to coordinates $\boldsymbol{x}$ the image plane and used as basis for the rectangular scaled areas $\Omega_{\boldsymbol{x}}$, which vary in size dependent on the distance from the camera (see red rectangles in Figures 4.4(a) and 4.4(b)). Within each area $\Omega_{\boldsymbol{x}}$ spanned by upper left and lower right corner points $\boldsymbol{x_{ul}} = (x_u, y_u)$ and $\boldsymbol{x_{lr}} = (x_l, y_l)$ the filled area percentage

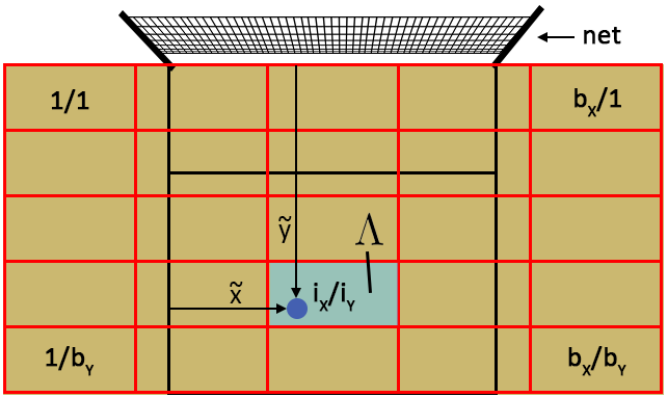


Figure 4.3: *Binning: The investigated area containing the court and surrounding borders is horizontally subdivided into b_x bins and vertically into b_y bins. The light blue area Λ_{i_x,i_y} is a bin with bin index (i_x, i_y), containing the blue example point $\tilde{\boldsymbol{x}} = (\tilde{x}, \tilde{y})$.*

$\Pi(\Omega_{\boldsymbol{x}})$ is computed by summing up all foreground pixels from the probability image P_{player} and normalizing the result by their area. As a result, for any point $\tilde{\boldsymbol{x}} = (\tilde{x}, \tilde{y})$ on the court a probability for the appearance A_p of a player at that position is obtained. This probability is equivalent to the filling percentage within the scaled area $\Omega_{\boldsymbol{x}}$ in the image plane and can be denoted as $\Pi(\tilde{\boldsymbol{x}})$ for the corresponding point $\tilde{\boldsymbol{x}}$ on the court plane.

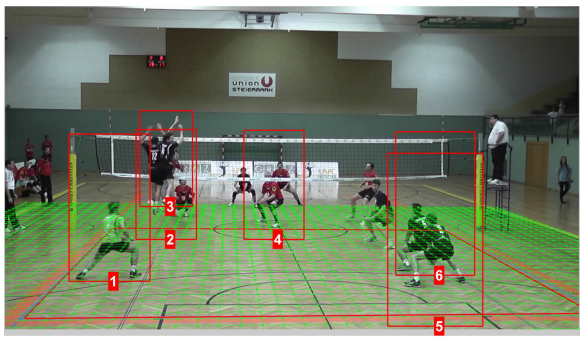
$$P(A_p|\tilde{\boldsymbol{x}}) = \Pi(\tilde{\boldsymbol{x}}) = \Pi(\Omega_{\boldsymbol{x}}) = \frac{\sum\limits_{\boldsymbol{x}\in\Omega_{\boldsymbol{x}}} P_{player}(\boldsymbol{x})}{|\Omega_{\boldsymbol{x}}|} \tag{4.3}$$

This filling percentage or probability $P(A_p|\tilde{\boldsymbol{x}})$ gives clues about presence or absence of a player at a certain point. As dense sampling of points $\tilde{\boldsymbol{x}}$ results in up to hundreds of thousands of values, the resulting probabilities are binned in x and y direction into areas Λ_{i_x,i_y} (see Equation 4.2 and Figure 4.3). These court subareas contain probabilities for all contained sampled points. Depending on the number of subdivisions such areas typically cover 0.5 to 1 square meters. The SC descriptor is the composition of all subareas and holds averaged player appearance probabilities within each bin, defined by vertical and horizontal indices i_x and i_y.

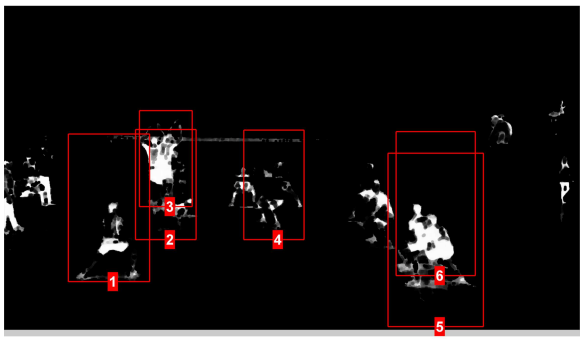
$$SC(i_x, i_y) = \frac{\sum_{\tilde{x}\in\Lambda_{i_x,i_y}} P(A_p|\tilde{\boldsymbol{x}})}{|\Lambda_{i_x,i_y}|} \tag{4.4}$$

An method overview and an example result are shown in Figure 4.4, where the dense sampled points are binned into 15 times 20 areas (see Figure 4.4(d)). Due to the camera position behind the court and resulting scale dependency, ambiguities cannot be prevented. If a player is relatively close to the camera, he fills out rectangles designed for player detections near the net. Such the probability for players near the net are high, although the player is in the back of the court.
As stated before with the RWPC descriptor (and shown in Figure 4.2), players move in and out of the court during the game depending on the actual situation. This needs to be addressed to collect all players involved in the game. Therefore not only the court itself but also some space behind the court (service), on both sides of the court (attack) and into the opponents court (attack/block) is examined.

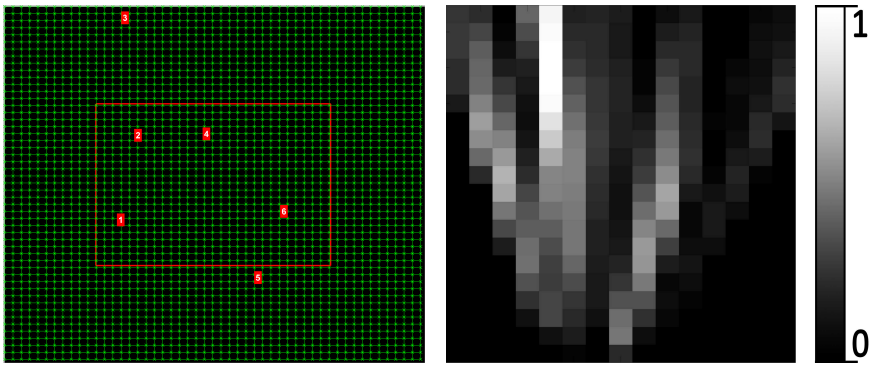
Speedup 1: Look-up-table
As thousands of rectangles must be computed for this dense descriptor, a look-up-table is precalculated for each video. This table contains the rectangle dimensions for every point on the dense grid.



(a) Frame with grid, exemplary points $\boldsymbol{x}$ (1 to 6) and scaled rectangles $\Omega_{\boldsymbol{x}}$. The court is outlined in red.



(b) Player probability image $P_{player}(\boldsymbol{x})$



(c) Top view of the grid and transformed points $\tilde{\boldsymbol{x}}$. Again the court is outlined in red.

(d) Final SC descriptor, binned into 15x20 bins

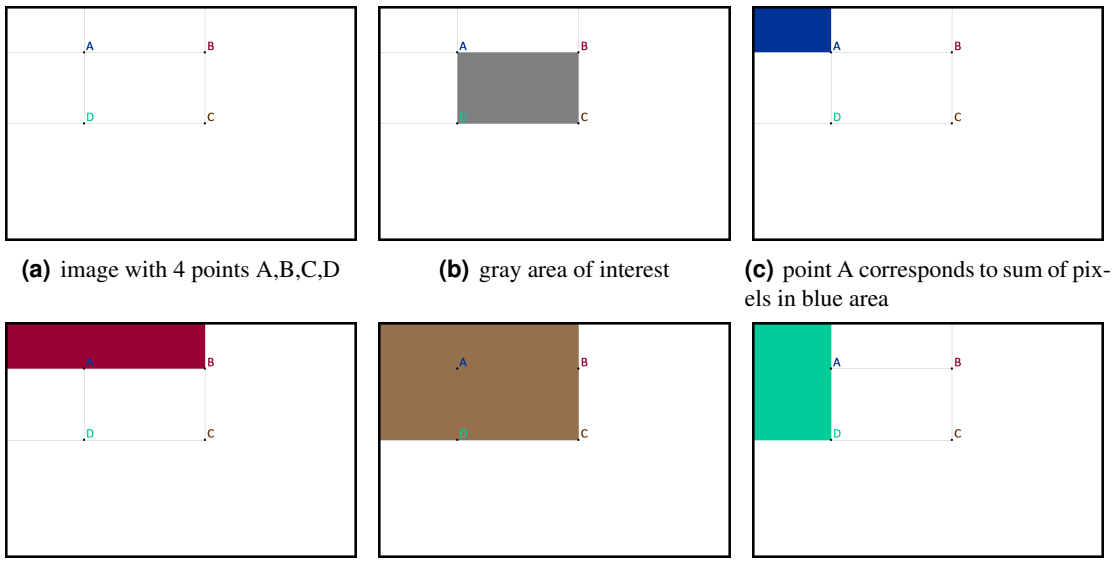
Figure 4.4: *SC descriptor calculation: After calculating the player probabilities P_{player} for the evaluated frame, for every point on the grid a corresponding rectangle $\Omega_{\boldsymbol{x}}$ and the filled area percentage Π is computed. Then the results are binned for dimension reduction. Note: Players closer to the camera (rectangle 5) also fill out rectangles further away (6), such blurring the descriptor. Also, players at the net (partly) fill multiple rectangles around them, as the step size is getting smaller with the distance from the camera (2,3).*

This way the transformed top-view points of the rectangle corresponding to a certain position in the court need not be calculated for each of the processed video frames.

Speedup 2: Integral image For better performance, the values in the rectangles are calculated with an integral image. Typically an integral image I_{int} is constructed by starting from the top left corner of an image I and summing up all the pixel values (greyscale or binary) row and column wise:

$$I_{int}(x,y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i,j) \tag{4.5}$$

The integral image can be calculated in one pass, as every point in the integral image is a sum of its top and left neighbor plus the value of the pixel at the same location of the input image.
After the image is constructed one does not need to access all points within an area, but only those on the border of the area. A certain point in the integral image corresponds to the sum of all set pixels in the rectangle spanned from this point as the bottom right corner to the top left corner of the image. For an arbitrary rectangle within the integral image, these points are the four corners of the rectangle. Thus, only three operations are needed to compute the sum within a rectangle - independent on the size of the rectangle or the number of pixels within this area. See Figure 4.5 for an illustration. Using the integral image, the calculation of the filling percentage can be simplified and accelerated



(a) image with 4 points A,B,C,D

(b) gray area of interest

(c) point A corresponds to sum of pixels in blue area

(d) point B corresponds to sum of pixels in red area

(e) point C corresponds to sum of pixels in brown area

(f) point D corresponds to sum of pixels in cyan area

Figure 4.5: *Illustration of calculation with an integral image. The gray area can easily be computed with only three operations: A-B+C-D.*

by changing the numerator in Equation 4.3 from

$$\sum_{\boldsymbol{x} \in \Omega_{\boldsymbol{x}}} P_{player}(\boldsymbol{x}) = \sum_{x=x_u}^{x_l} \sum_{y=y_u}^{y_l} P_{player}(x,y) \tag{4.6}$$

to

$$\sum_{\boldsymbol{x} \in \Omega_{\boldsymbol{x}}} P_{player}(\boldsymbol{x}) = I_p(x_u, y_u) - I_p(x_l, y_u) + I_p(x_l, y_l) - I_p(xu, y_l) \tag{4.7}$$

after calculating the integral probability image I_p from the foreground probability image P_{player} (Equation 3.20).

$$I_p(x,y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} P_{player}(i,j) \tag{4.8}$$

4.3 Activity Context Descriptor (AC)

The previously proposed SC descriptor makes use of other players *positions* in the same video frame and supports the classifiers decision. In comparison, the activity context descriptor intends to improve prediction results by gathering information about the other players *activities and connected positions over time*. There are some main activities in volleyball, although always only one player is in direct relation to the ball, the other players behave according to the specific activity. These defined team activities are deduced from game situations. The dataset presented and used in this thesis consists of seven classes. Five of them are volleyball specific activities and two are general activities. A short introduction about the relationship between the activities follows, motivating the exploration of temporal and spatial activity context between players on the court. **Service**: While one player brings the ball into play, the other players wait in the court and move to their designated positions. **Reception**: When the opponent serves, the 3 receivers position to play the ball to the setter. During the ball flight from server to receivers and on to the setter the other players position themselves such that they are ready for attack. **Setting**: The setter passes the ball to one of the attackers on court, simultaneously the attackers begin with their approach. **Attack**: The attacker who receives the ball for attack executes the attack. In the meantime all other attackers abort their attack movements and try to cover the attacker in case of successful opponent block. **Block**: One to three blockers try to build a block against the opponents attacker. The other the players spreads in the court trying to defend the space not covered by the blockers. While the previous five activities are volleyball specific and occur within attack or defense patterns, the remaining two activities are very general and do mostly not occur in specific moments. **Stand:** A player standing on the court. Usually happening between points while waiting for a service or for moving to ones designated position. **Defense/Move**: Activities within this category are collected unspecific motions occurring throughout the game. Any movement that does not fall into the above five specific categories is contained within this class.

In this thesis it is believed that studying the other players while investigating one player executing a specific activity might support recognition of this activity. Some of the activities are exclusive (service, attack, setting), while others can be performed by multiple players simultaneously (stand, move/defense, reception, block). Often a temporal overlap of two activities is needed for successful play. For example, while the setter is playing the ball, the attacker already approaches the net and jumps as the ball leaves the setters hands. This timing allows the attacker to hit the ball at his highest position during the jump. As the interaction between players is dependent on the current game situation and the transitions between activities are rather smooth, the length of the temporal investigation prior to the actual frame is important and modeled by the parameter τ.

The proposed activity descriptor works as follows: To obtain the unknown player positions, the upper body blobs are extracted via player foreground probabilities P_{player} and the positions are estimated. With these estimates the corresponding scale dependent bounding boxes can be calculated for feature extraction. The shape (Section 3.2.1), motion (Section 3.2.3), position (Section 4.1) and spatial context (Section 4.2) features are then put into the previously trained SVM classifiers for evaluation of all activities A. Such, for every blob in the image, a classification result containing probabilities for all activities is available. As the number of blobs (players) and such the number of evaluated positions on the court is small - typically between 5 and 10 classifications per frame, dependent on the number of extracted blobs - these positions are binned resulting in a compact descriptor size. Like before,

the spatial binning is done in b_x horizontally and b_y vertically partitioned bins by translating the normalized position $\tilde{\boldsymbol{x}}_{\boldsymbol{n}} = (\tilde{x}_n, \tilde{y}_n)$ (Equation 4.1) of every player on the court to the corresponding bin indices i_x and i_y (Equation 4.2). While for the SC descriptor only the player position probabilities are binned, the AC descriptor additionally contains information about the player activities. The AC descriptor matrix is of size $b_x \times b_y \times a$, where b_x and b_y denote the number of partitions in vertical/horizontal direction and can be seen as subdivisions of the court while a is the number of activity classes in A. Each matrix cell is describing the average probability of all occurrences Θ for each activity class c in the bin (i_x, i_y) within the chosen time span of τ previous frames.

$$0 \leq i_x \leq b_x, \qquad 0 \leq i_y \leq b_y, \qquad c \in A$$

$$AC(i_x, i_y, c) = \sum_{n=1}^{|\Theta|} \frac{P_n(c|i_x, i_y)}{\tau} \tag{4.9}$$

To control the possible introduction of noise by weak classes, an additional parameter p is introduced. This parameter controls the number of class responses for a evaluated player, that should be integrated into the descriptor while the other weaker responses are omitted. When p is set to the number of classes in A, no restrictions are imposed on the classification result.

$$p = |A| \qquad c \in A \tag{4.10}$$

As p is set to 1, only the strongest class response from the classification is used. This is equivalent to non-maxima suppression.

$$p = 1 \qquad c = \arg\max_{c \in A} P(c|i_x, i_y) \tag{4.11}$$

For soft-pooling, the strongest p class responses of A are collected within a subset B and the weaker responses are omitted.

$$p = |B| \qquad c \in B \qquad B \subseteq A \qquad |B| < |A| \tag{4.12}$$

Figure 4.6 illustrates the method for a sample frame where the front team is in receiving activity.

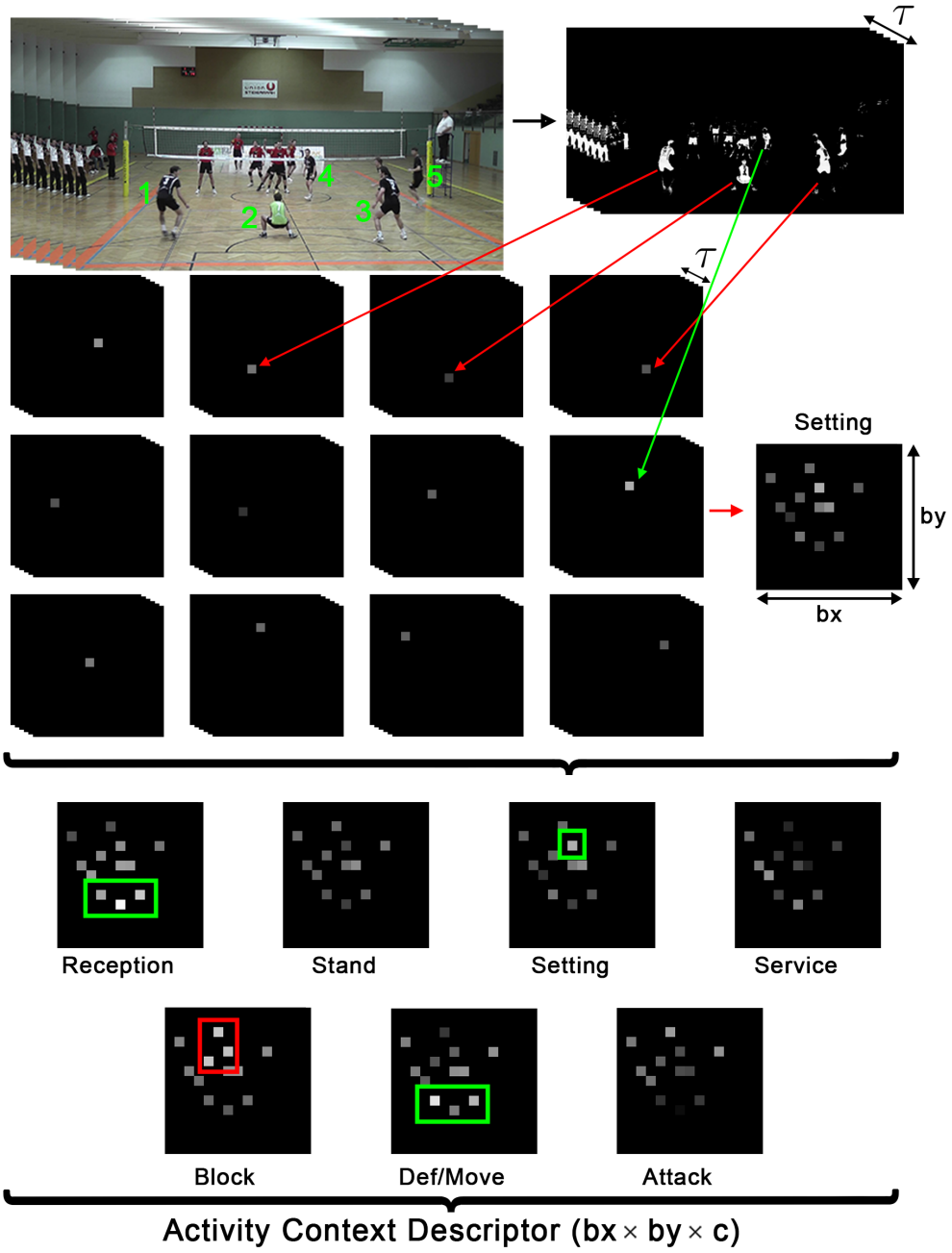


Figure 4.6: *Illustration of the AC descriptor: Blobs from a series of k frames are extracted, each blob is classified and the results saved in a $bx \times by$ map (averaged by k). For the receiving players numbered 1,2 and 3 (red arrows), the probability for the class "Setting" is expectedly low, whereas for the setter at the net (4, green arrow) the probability is high. The AC descriptor is the combination of all c class probability maps. The response for the three receiving players is evident by the high values in the "Reception" map, but also the related "Defense/Move" class shows strong responses (green rectangles). Due to the proximity to the net, the opposite players influence the "Block" map. This can be considered noise (red rectangle). The player marked with number 5, although standing causes a strong response in the "Attack" class, as this is a typical position for attack and the classification framework is biased by spatial information. The sixth player on the court is not marked as he is behind player 3, and is a good example for occlusions in the video data.*

Chapter 5

Activity Recognition in Volleyball

5.1 A Brief Explanation of the International Volleyball Rules

With an estimated number of almost 1 billion participants, Volleyball is on of the most popular team sports worldwide. This includes beach volleyball, played by 2 players per Team, and indoor volleyball, where 6 players team up. As this thesis focuses on indoor volleyball, only indoor rules will be explained as simple as possible.
Two teams are separated by the net, defending a 9x9m area (court). The goal is to keep the ball from touching the court on your side of the net, while trying to ground it on the opponents floor. Each team is allowed 3 short contacts with the ball. A team loses a point by either having the ball touch inside the own court or making another illegal move like touching the net. Also penalties can be given by the referee. The ball is brought into play by a player from behind the court who has to hit the ball over the net into the opponents court (service). On the other side the players try to control the served ball by receiving and placing it - if possible - within a certain area at the net (reception). There, another player takes control of the ball and passes it on (setting), acting as the mastermind of the game. He decides which player will get the chance to score the point and tries to present the ball in a best possible way. The player with the last contact then hits the ball over the net and a) directly into the field or b) outside the field having an opponent touch the ball (attack). The second option is often used when the defending team forms a block at the net, then the attacker tries to hit on the top or side of the block. The resulting deviation often makes it impossible for the defending team to recover the ball before it touches ground. At the begin of each set, the coach must choose 6 players from the team - a total of twelve players are allowed to be listed on the score sheet. These 6 players take up positions in the field. At the moment the ball is hit by the server, each team must be positioned within its own court in the rotational order (except the server). Figure 5.1 shows the positions in court.
As in other team sports, the coaches can substitute players and take timeouts to influence the game. Furthermore, there is the special position of the so called Libero player - he is a defense specialist and can substitute for the back-row players (positions 1,5,6) at any time between two rallies.
Referees
A rally always begins and ends with the whistle of a referee. There are two referees at each game and up to four line judges supporting the referees at their decisions. The first referee is elevated on a chair on one side of the net and makes all decisions. The second referee is standing on the other side, handling time-outs, substitutes, line crossings, rotation and position errors and supports the first referee in his decisions. There are 25 hand signals (referees) and 5 flag signals (line judges) in volleyball [1].

[1] A poster with all the signals can be found at http://www.fivb.org/EN/Refereeing-Rules/Documents/FIVB_Volleyball_Hand_Signal_Poster_2013.pdf

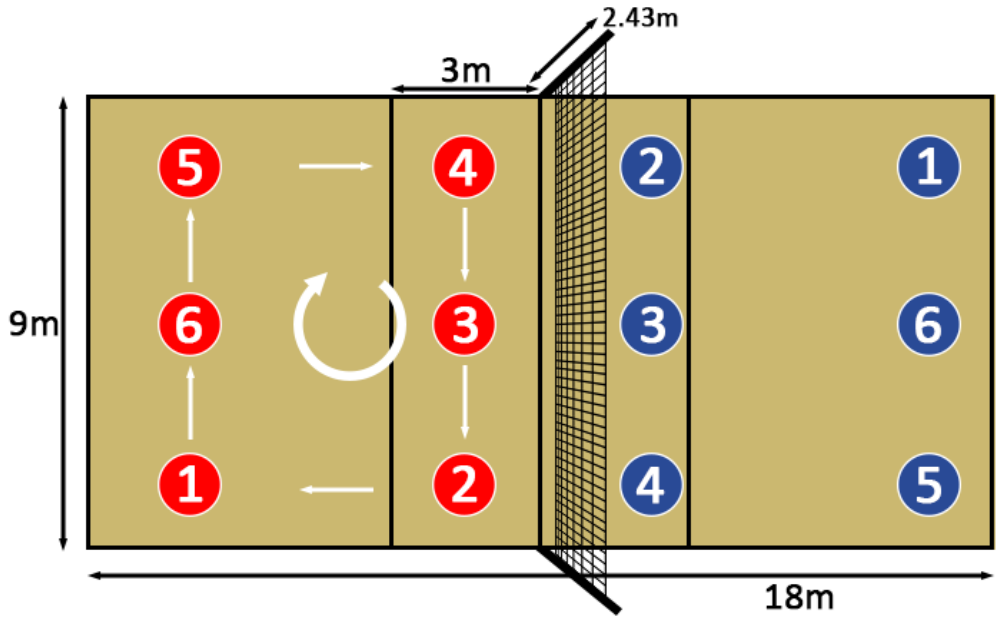


Figure 5.1: *Volleyball court measures and player positions (1-6). The rotation order is indicated by arrows.*

Illegal moves
Illegal moves that directly lead to a point loss:

- Touching the net while playing the ball.
- Playing the ball over the net outside of the antennas.
- Step on the back line while serving (foot fault).
- Step over the middle line into the opponents court.
- Catch, hold or lift the ball.
- More than three contacts. Special case: a block touch does not count as a contact.
- More than 1 contact from one player in a row. Special case: block as first touch is allowed.
- Positioning faults: Every player must start a rally in a certain position.
- Rotational fault: Every player must follow the serving order determined at the start of the set by the initial lineup.
- Receiving a penalty by the referee.

Scoring
Volleyball is played in a running score manner. Each point counts, no matter who serves or made the last point. To win the game one team must win 3 sets (best-of-five system), a set is won if a team can score at least 25 points with a 2 points difference (or more). If one team reaches 25 and the other is only one point behind, the set continues until one team can reach the 2 point gap.

A detailed collection of up-to-date official international rules can be found on the web page of the International Federation of Volleyball (Fédération Internationale de Volleyball, FIVB) [2].

5.2 Annotation Framework

As this thesis is based on indoor volleyball - a sport previously not on the radar of computer vision research - there existed no data sets for evaluation of methods. The need for annotated data was satisfied by the use of a special designed Matlab annotation framework. The target of activity extraction from video files was achieved by allowing the user to navigate within the video and annotate the different activities with simple mouse clicks. An example can be seen in 5.2. An annotation depicts a

Figure 5.2: *Matlab annotation framework: The user has detailed options to annotate volleyball videos. A list of annotated activities is maintained and all annotations of the actual frame are displayed. Navigating through a video is eased with keyboard shortcuts.*

bounding box around a player. Together with such a bounding box the user chooses player number, executed activity, if the marked player belongs to the home or opponent team and if he is occluded by other players or not. Although it is also possible to assign simple actions (jump, hit, land,...) to the players, only complex activities consisting of a sequence of such actions were annotated. This makes annotation simpler but presumably complicates the recognition task as some actions occur in multiple classes introducing intra class variances.

Activities

The following activities are used in the annotation (and such in the proposed method), characterizing the volleyball sport: *"Stand", "Service", "Reception", "Setting", "Attack", "Block", "Defense/Move"*. A detailed description of the activities is given in Section 5.4.

[2]For official international rules see http://www.fivb.org/EN/Refereeing-Rules/RulesOfTheGame_VB.asp

5.2.1 Calibration

The video camera is mounted behind the field on different positions. Therefore first a calibration of the view is needed, which is done by selecting points on the field (see Figure 5.3). This calibration is later also used for the context descriptors (see Sections 4.2 and 4.3 for details).
For the forward spatial transformation at least four point pairs are required (see Figure 5.3). The points

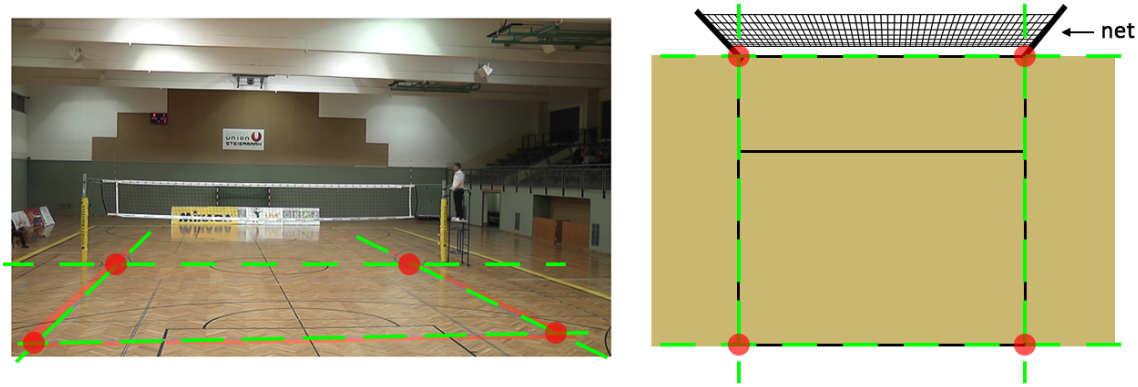


Figure 5.3: *Video calibration: Four corner points of the court need to be marked for a proper calibration of the ground plane, needed for calculation of the planar homography. After choosing the 4 points on the middle and base line the video is ready for processing.*

$\boldsymbol{x} = (x, y)$ selected by the user then are projected via the transformation function to corresponding points $\tilde{\boldsymbol{x}} = (\tilde{x}, \tilde{y})$ on a rectangular grid, resulting in a top view transformation:

$$\tilde{\boldsymbol{x}} = T(\boldsymbol{x}) \tag{5.1}$$

For details about inference of the transformation matrix see Section 3.1.1.

5.3 System Overview

The proposed system consists of three main parts: Preprocessing, spatial context player activity recognition and activity context player activity recognition. First the videos are prepared and annotated. After the manual annotation process (Section 5.2), geometry estimation (Section 3.1.1) and creation of foreground and background models (Section 3.1.2), the videos are ready to be examined under the proposed methods (Sections 3.2.1, 3.2.3, 4.1-4.3). The classification models (Section 3.3) are tested with previously ”unseen” data to give performance results of the proposed methods. For the spatial context player activity recognition a SVM is trained with manually annotated data only. For the activity context player activity recognition a new SVM is trained, using classification results from the previously trained SVM that are evaluated on manual annotated data as well as on automatically segmented players. In each case a 50%/50% split between test and training data is used. This section provides an overview of these tasks.

5.3.1 Preprocessing

As automatic tracking of the players is beyond the scope of this thesis, the videos are annotated within a built Matlab annotation framework. Therefore the videos are calibrated on a ground plane, to ensure the bounding boxes, which are marking the players during their executed activities, are

adapted to different scales. Since the video camera is mounted behind the court, players at the net appear smaller on the video than players in the middle or behind the court. Figure 5.4 displays the process.

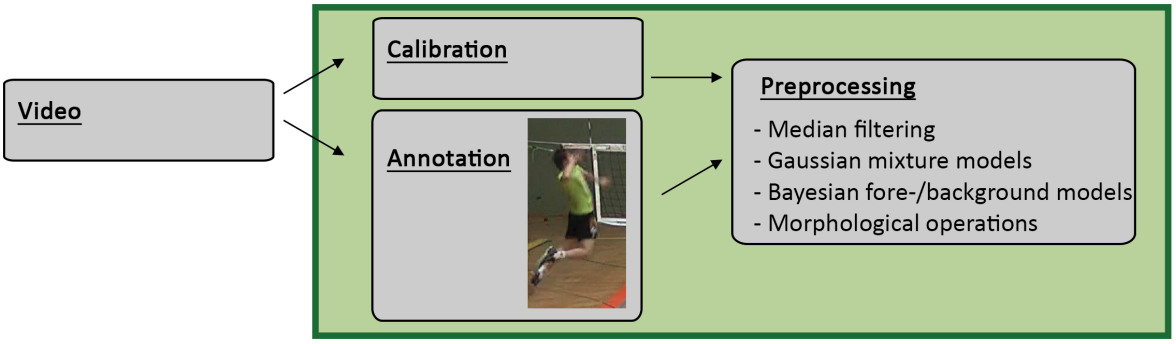


Figure 5.4: *Preprocessing: The input videos are calibrated (Section 5.2.1) and the players annotated (Section 5.2) as tracking substitution. Then the color models for the background and foreground is learnt (Section 3.1.2) and together with a median filtered background image a bayesian player probability image is calculated (Section 3.1.3).*

5.3.2 Feature Extraction Pipeline

The feature extraction process starts with interpolation of the manually annotated frames. Position and size of each bounding box around the player are interpolated between the marked key frames (Figure 5.5). Typically, every fifth to tenth frame is manually annotated. These intervals stand for a good offset between accuracy and annotation effort. As second step, the interpolated frames are cropped to

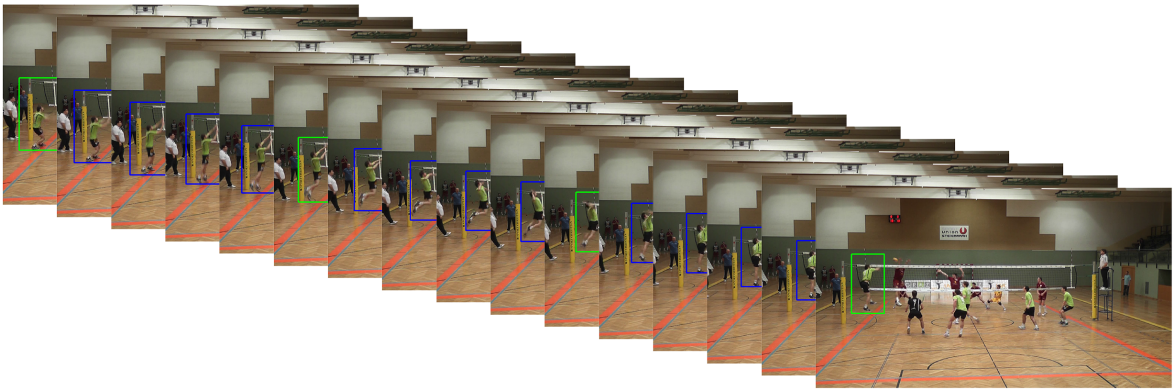


Figure 5.5: *Interpolation of annotations: Key frames are marked green, the interpolations are shown as blue rectangles.*

bounding box size, scaled depending on the player position using the reckoned homography. This is important for limiting the calculation area to the region of interest as the videos are available in HD resolution. The bounding box clippings are of small size compared to the full HD frames such the features from the HOG descriptor are calculated more efficiently. The RWPC and SC descriptors are calculated from the whole frame, as well as the optical flow that was pre-calculated for each frame

separately. The HOF descriptor is then calculated from the cropped region in the flow image, similar to the HOF descriptor.

5.3.3 Spatial Context Player Activity Recognition

For the first main task, multiple features are extracted from each annotated player. Due to the manual data preparation a good annotation accuracy is assumed that motivates the use of HOF features for motion and HOG features for shape description. Besides these two descriptors this thesis on the one hand proposes player coordinates (RWPC) calculated via real world geometry and on the other hand the use of a spatial context descriptor (SC) that models the on court player distribution. These features are concatenated into a vector for each annotation and classified via a SVM, trained on one half of the manually annotated data. See Figure 5.6.

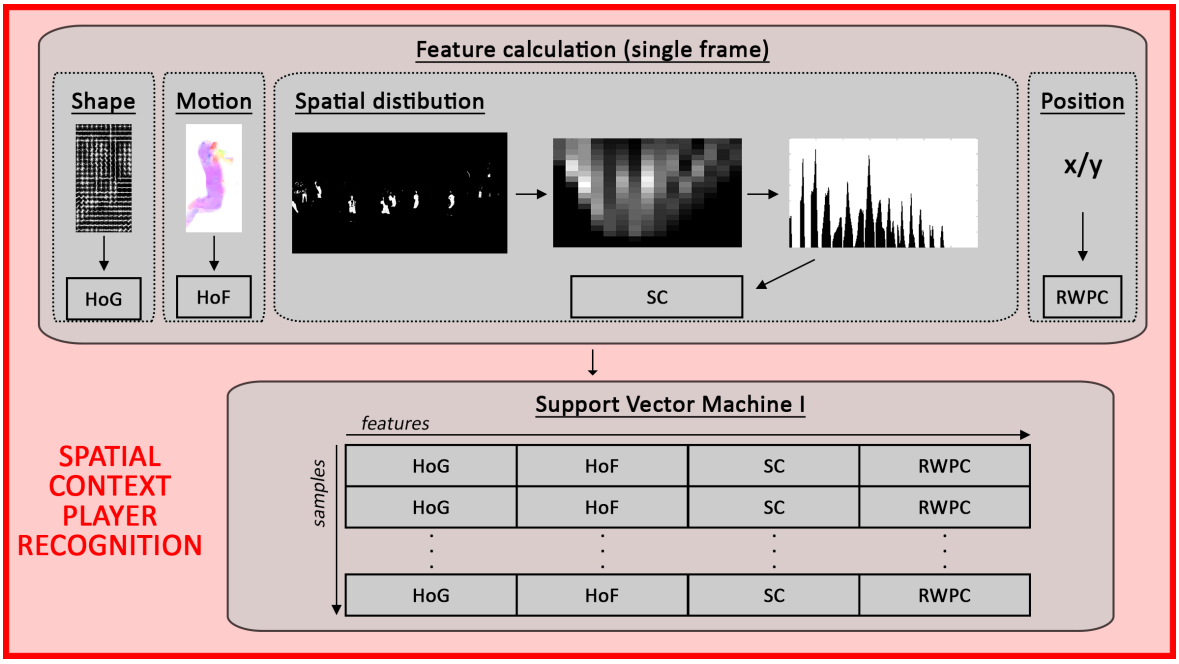


Figure 5.6: *Spatial context player activity recognition: Features for an annotation in a frame are calculated (HOG, HOG, RWPC and SC). Then the SVM is trained and the results are verified.*

5.3.4 Activity Context Player Activity Recognition

As second main task this thesis tackles activity context recognition. Therefore activities over time of all players on the court are examined and this information is used via the AC descriptor to improve the results of the examined annotated players. The classification process is similar to the one in the previous described task, only AC features are added to the single frame features to incorporate activity information of all players over time. A graphical visualization is shown in Figure 5.7.

5.4 Data

The videos used in this thesis were recorded from matches in the 1st volleyball league of Austria. The videos are in HD resolution (1920x1080) at 25fps, compressed with the DivX codec(www.divx.com).

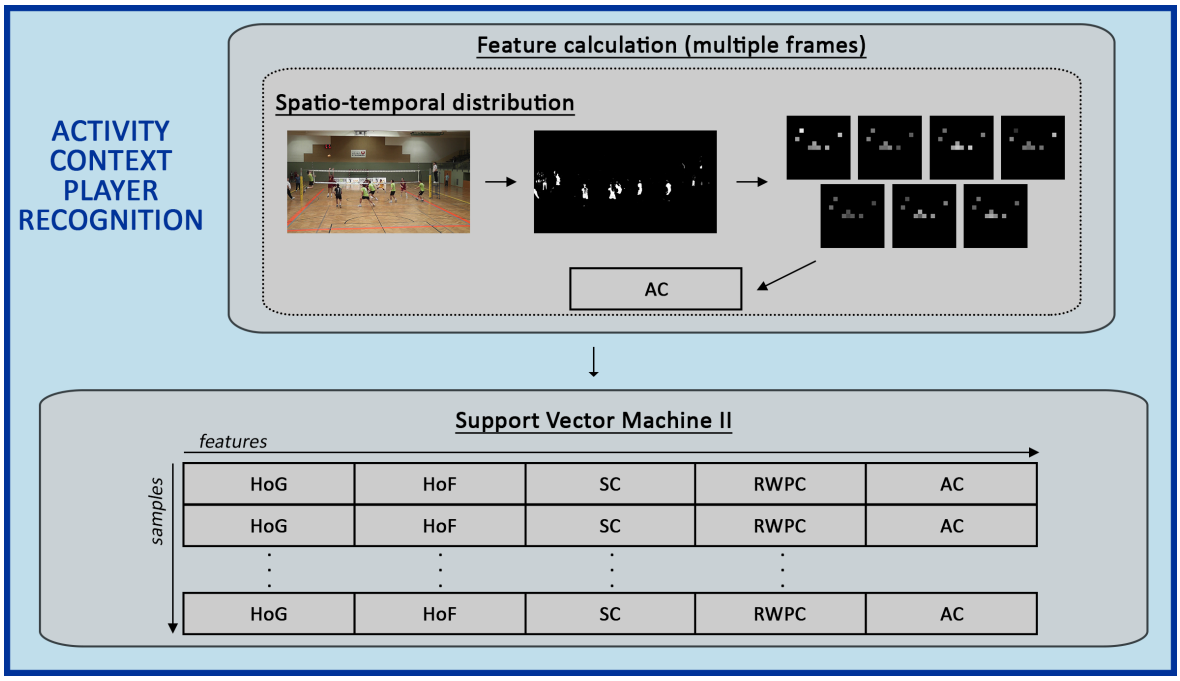


Figure 5.7: *Activity context player activity recognition: For activity recognition additional features are calculated that describe what the other players on the field do while the annotated player executes some activity. These features are added to the previously calculated feature vector and improve the results when using a special trained SVM.*

Six video clips are used from three different games. The typical procedure of play was followed to determine which activities were chosen and where the temporal divisions between activities needed to be set: First a ball is served from one side of the court to the opponents side, then the receivers try to control the ball, the setter passes it on to the attackers which hit the ball trying to score a point. The serving team tries to avoid a point by blocking the ball directly a the net or prevent the ball from touching the floor within the court.

1. **Stand**
   A player standing on the court. Not much movement and upright stand. This activity occurs mostly before a service activity (opponent or own team) when the players wait for the ball to be brought into play, so they can leave their position and move to specific positions in the field.

2. **Service**
   A player must bring the ball into play from behind the court baseline. After a team wins the point, it has to serve the ball over the net to continue the game. As soon as the server touches the ball, all players are allowed to move into their specific positions on the court. For example, a setter runs to the net awaiting the ball coming from the receivers.

3. **Reception**
   A player receives a ball brought into play by a server from the opponent team and passes it on to the setter (1^{st} touch). The receivers try to place the ball to the setters position as accurate as possible.

4. **Setting**
   The setter brings the attackers into play by distribution of balls to specific positions, mostly at

the net (2nd touch). While the ball travels from server over the net and to the receivers, the setter moves to a previously arranged position at the net. The setter passes the balls to the attackers, choosing from different previously communicated attacking scenarios.

5. **Attack**
   The attacker hits the ball and tries to win the point for the team by hitting it down into the opponent court (3rd touch). He can also try hit the ball out via the blockers hands. If an opponent last touches the ball before it hits the ground (inside or outside the field), it is a point for the attacker.

6. **Block**
   The blocker tries to protect area of the own field while the opponent attacker tries to hit the ball. If the block can send the ball back into the opponents court, touching the floor (inside the court) it is a direct point. If the ball goes out the court from the blockers hands, the point is lost to the opponent.

7. **Defense/Move**
   All other movements on the court with and without ball interaction are collected into this class: A defense is an attempt to avoid the hit ball touching the floor. As this activity is very variable, also the different moving activities without ball are out together into this category. For example, players not involved in the attack try to distribute on the field such that balls coming back from the opponents blockers can be saved.

After hours of annotation, a total of approximately 2,5 hours (2:23:53, see Table 5.1)) of video material was accurately annotated yielding 7999 activity annotations. A subsequent set of annotations

video name	**length (mm:ss)**
graz-arbesbach_2.avi	21:41
graz-arbesbach_3.avi	21:38
graz-arbesbach_4.avi	21:35
graz-arbesbach_5.avi	21:44
graz-gleisdorf_1.avi	35:33
graz-klagenfurt1_2.avi	21:42
total	2:23:53

Table 5.1: *Video data*

of the same player and activity are collected in a short video clip, called a tracklet. A tracklet such contains one player performing a complete activity over approximately 40 frames on average. As the annotations were made every 5 to 10 frames, a larger set was generated by interpolating the players positions. With a frame rate of 25fps the interpolated activities are precise enough. As a result, starting with 8000 manual annotations, the data set generated contains over 36000 annotations or 1000 tracklets. The number of annotated frames (4399) is far smaller than the number of annotated activities (7999), as often only one key player per frame is annotated but for the activities ”Block”, ”Stand” or ”Defense/Move” often up to three players are annotated per frame. This is also the reason for the dominance of ”Block” annotations within the data, as a block is more often executed by two or three than by one player alone. The alert reader might notice, that the classes ”Stand” and ”Defense/Move” depict very general classes of activities. They are not volleyball specific and can be found in various other sports or even every day live situations. This is due to the fact, that neutral activities should be added to the specific Volleyball activities. This should show the generalization possibilities of the proposed methods.

Figure 5.8: *Examples for all activities, randomly chosen from the dataset. Each row contains activities from one video. The columns present the seven activity classes from left to right: "Attack", "Block", "Defense/Move", "Reception", "Service", "Setting", "Stand".*

Due to the game immanent structure, the activity occurrences differ. Also some activities like ”Block” or ”Stand” can be executed by multiple players simultaneously. Still, the number of activities is quite balanced with around 1000 samples per activity class. Table 5.2 shows a list of all activities and their quantities.
The annotations work was done by one expert (the author) and two persons without prior knowledge about the volleyball game structure. Compared to many papers this dataset can be considered rather large. This supplies a good basis for evaluation of the proposed methods as the videos differ more or less in terms of camera position, jersey colors, involved players/teams or illumination, all being real world problems for a computer vision application.

activity name	**number of tracklets**	**number of annotated frames**	**number of annotations**	**number of interpolated annotations**
Stand	146	694	1339	6181
Service	126	811	868	3911
Reception	94	761	767	3482
Setting	151	850	891	3903
Attack	148	1108	1157	5233
Block	241	1039	1847	8332
Defense	174	825	1130	5062
total	1080	4399	7999	36104

Table 5.2: *Activity quantities: This table displays information about the dataset. The left column shows the activity labels as previously defined. Tracklets are player activity clips, the number of frames depicts the total count of frames containing at least one annotation, the number of annotations shows the quantity of manual annotations per class and the interpolated value denotes the total number of annotations per class available within the dataset.*

Chapter 6

Experiments

This chapter contains the experiments conducted on the data set. First, various parameters of the descriptors and the classifiers had to be determined and empirically evaluated to find good values. Then, the performance was evaluated on a 50/50 data split. One half of the data was used for training, the other half for testing the system with unknown data. To avoid the appearance of similar data in training and testing the data was split after the tracklet in the middle of the dataset, resulting in slight shift of the 50% border to 50.32% training and 49.68% testing data.

6.1 Parameter Configuration

For testing the methods, many parameters need to be evaluated and set. For the spatial context player activity recognition (Section 6.2) the HOF, HOG, SC and RWPC descriptors and the SVM classifier can be configured in a number of ways for optimally matching the task. For activity context player activity recognition (Section 6.3) the AC descriptor can be parametrized in various possibilities. This parametrization was tested excessively and will be introduced in this section.

6.1.1 RWPC

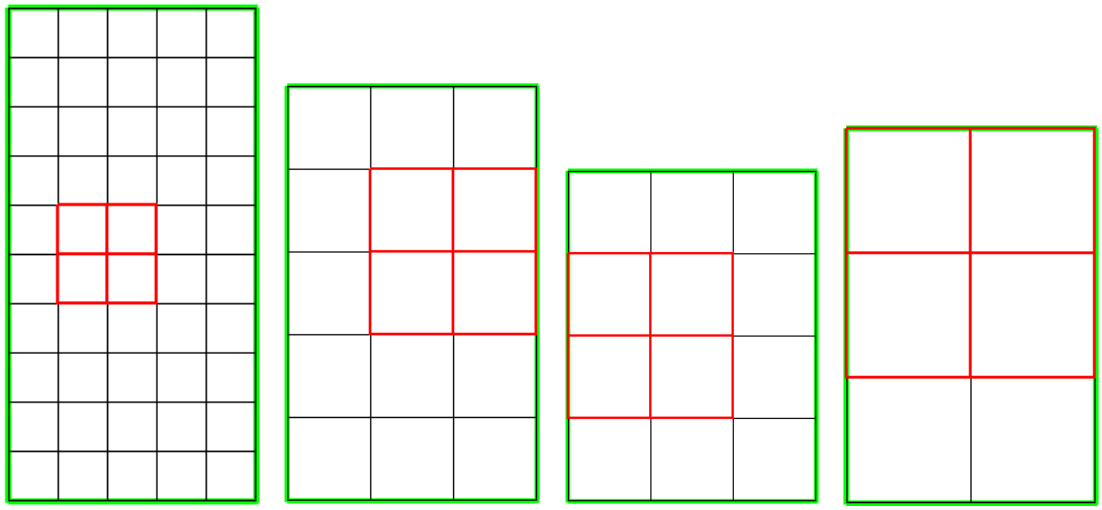
The position coordinates from the RWPC descriptor (Section 4.1) were normalized to values between 0 and 1. As they contain only two dimensional x and y coordinates, these features did not need any special parameter set. Only the area of the included border outside the field had to be declared, for simplicity and to avoid exclusion of any point it was set equally to the field width.

6.1.2 SC

To include the field and the outside areas in the SC descriptor, a grid of 300x250 evenly distributed points was used. After calculating the fill percentages for these 75000 rectangles, they were binned into 15 times 20 bins resulting in 300 features. Other binnings were tested but did either not improve the results or only by a small amount while greatly increasing the number of features.
The calculation of the rectangles is very time consuming, down-scaling of the images would improve the speed while deteriorating the results. As this thesis emphasis lies more on accuracy than on speed, during the experiments the scale was set to 1.0.

6.1.3 HOG/HOF

Setting the parameters for the HOG and HOF descriptors is vital, as very small cells create much noise in the descriptor while too big cell sizes result in only vague estimation of the shape and motion. Within the works context and due to adaption on the dataset, where the inner-class variability is very large because of a rather coarse annotation, better results are achieved with larger cell sizes.
Different sets were tested, as can be seen in Table 6.1 and are illustrated in Figure 6.1. Four parameters



(a) cell size: 32, window size: 160×320, cells per block: 4, orientation bins: 9 **(b)** 64, 192×320, 4, 9 **(c)** 70, 210×280, 4, 9 **(d)** 88, 176×264, 4, 9

Figure 6.1: *Illustration of different parameter sets for HOG/HOF descriptor. One block per example is marked red, consisting of multiple cells (in this case four cells). The window size (bounding box containing the player) was slightly adapted to be a even multiple of the cell size.*

were to be set, the patch size (width and height) to which each bounding box including a player in activity should be resized for further processing. The cell size within the bounding box, the number of cells combined to a block and the number of bins the orientations should be sorted. The number of cells per block was set to 4 and the number of bins to 9 (40°steps), as the number of features should not become to large and a smaller number did not seem to suffice.

cell size	patch width	patch height	cells per block	bins	blocks	dimensionality of features
32	160	320	4	9	36	1296
64	192	320	4	9	8	288
70	210	280	4	9	6	216
88	176	264	4	9	2	72

Table 6.1: *Parameter sets for HOG and HOF descriptors*

6.1.4 AC

The activity context has four parameters to be set. The first and most influencing parameter τ is the number of frames examined (and included) by the descriptor. The more frames are used, the more information but also more noise is included. One second corresponds to 25 frames. The second

parameter determines how many votes are included in the descriptor for every player. For every blob, the classification returns seven class probabilities. With the parameter, one can sort out the weak votes and only keep the p best votes. The last two parameters influences the number of features, they control the binning of the activity context in two directions (field width and field height). An overview of the tested parameters can be seen in Table 6.2.

p	k	bins
1,3,7	10-250	10x10, 15x10

Table 6.2: *Parameters for AC descriptor.*

6.1.5 SVM

Within the SVM framework overall three parameters were evaluated: c, γ and the kernel type. The type of the kernel used is a very important parameter. Using the freely available library LIBSVM[1] five kernels are available: linear, polynomial, radial basis function (rbf), sigmoid and precomputed. Out of these, four were tested: linear, polynomial, rbf and sigmoid. The χ^2-kernel, broadly used for computer vision classification tasks, could not be evaluated as it is not included in the used Matlab SVM library. Details about motivation of kernel choice can be found in Section 3.3.1. While being slower, the polynomial and rbf kernels outperformed the two others in most cases. This is due to the great class intra-variability, making a linear decision hard.
The cost parameter c controls the degree of allowed misclassification. Allowing no errors (large value) on the training set creates a so called hard-margin SVM but decreases generalization and leads to overfitting. As a result the performance on the test set is mostly bad. On the opposite, allowing some errors (smaller value)leads to a soft-margin SVM. The value of c is the cost a misclassified point adds in dependance to its distance from the margin - the distance is multiplied with c so that wrong classifications further away are more costly than near ones. A low c makes the decision surface smooth, while a high c aims at classifying all training examples correctly and such increases the complexity of the separating hyperplane.
The third parameter γ also controls the shape of the hyperplane by setting the influence of a single training example on the hyper plane. Increasing the value causes a higher number of support vectors to closely adapt to the examples, possibly causing overfitting. A low values means that the influence of single training examples is small and the hyperplane is not fitted exactly to the data. This might cause underfitting.
Such, increasing both parameters leads to overfitting. This is why when increasing one parameter, one should decrease the other. Setting both parameters to a high value will work well on the training data but fail on test data. Of course, the terms *high* or *low* depend on the underlying data and need to be evaluated for every application separately.
For the results a adaptive grid search was executed on (portions of) the data. As the features, kernels and descriptor parameters change, different configurations were used. Three value pairs were tested for the parameters c / γ: 5.66 / 1.05, 32 / 0.18558 and 181.02 / 0.03. With the above mentioned four kernel types, the number of tested SVM configurations multiplies to 12. For evaluation a one-vs-all multi-class SVM was used, learning a distinct model for every class and assigning the label with the highest response from all class models.
The SVM for the activity context player activity recognition is trained with similar parameters as the SVM used for the spatial context player activity recognition, where the most successful parameters are $c = 181.02$, $\gamma = 0.03$ and a polynomial kernel. The rbf kernel could have been used as well as the performance of these two kernels is comparable good.

[1] Version 3.12, http://www.csie.ntu.edu.tw/ cjlin/libsvm

6.1.6 Descriptor Combinations

To test which descriptor and which combination of descriptors has the best impact, all descriptors were cross-evaluated. Choosing from four descriptors (RWPC, HOF, HOG and SC) yields 15 combinations to test. Table 6.3 lists all combinations.

descriptor combinations				
SC	RWPC	HOF	HOG	SC-RWPC
HOF-SC	HOG-SC	HOF-RWPC	HOG-RWPC	HOF-HOG
HOF-SC-RWPC	HOG-SC-RWPC	HOF-HOG-SC	HOF-HOG-RWPC	HOF-HOG-SC-RWPC

Table 6.3: *Possible combinations from all four descriptors.*

6.1.7 Tracklet Cuts

As the annotations start about one second before the player makes contact with the ball and end about one second after the contact, they include acceleration and deceleration motions. For example for the activity ”Attack” or ”Service”, the annotation includes the approach, jump, hit and landing. For ”Block”, the annotations include the movements of the players parallel to the net to the position where they jump, reach over the net and land back on the floor. The ”Setting” annotation follows the setter running in position, jumping, setting the ball and landing.
To evaluate the influence of this activity noise to the overall activity recognition rate, the tracklets were cropped to the core of the activity. The core describes the short period of time, when the player makes contact with the volleyball and is located mostly in the second half of the tracklet. Like before, different parameter values were tested: 0%, 20%, 33%, 50%. Here, 0% means no cropping is applied, and for example 30% means that the first 30% of the tracklets frames are removed. Of course, cropping the tracklets reduces the number of examples.

6.2 Results for Spatial Context Player Activity Recognition

The frame by frame recognition task is the basis for the following extended recognition (time- and context-wise). With widely used descriptors and a purposely designed spatial descriptor, every activity frame was classified. For the single player activity recognition a total of 1415 experiments were conducted. This part shows the results of the spatial context player activity recognition in detail. Many parameter sets were tested to have a broad foundation of the discussion of the results. First, a general overview is presented. Then the influence of SVM parameters is described and finally each descriptor is discussed separately.

6.2.1 Results for Differing Descriptor Combinations

Figure 6.2 displays all evaluations within one image. The horizontal axis lists different descriptor combinations, while the vertical axis shows the performance of correct classified activities. From left to right, descriptors are added for evaluation. The HOG descriptor performs best, while the HOF descriptor is of inferior performance. This might be because many activities are executed mirror-inverted, once from the right side of the field, the other time from the left side of the field. ”Attack” is a typical activity that can be executed from various positions on the court and such the movement in relationship to the camera is varying. The SC and RWCP descriptors, both focusing on player position information only, are performing just around 50% for most cases and considering that no activity

description is involved this is a good result. Maybe this exclusion of activity specific information is also the strength of these descriptors, as many classes have strong intra-class differences while having similar player positions on court. Within the groups of descriptor combinations, using one, two, three and four descriptors, the best results are achieved by combinations containing SC and HOG descriptors. As depicted in the figures, adding descriptors steadily improves performance.
The results are strongly influenced by the choice of the svm kernel. The sigmoidal kernel is almost in any case worse than the other three kernels, where rbf and polynomial kernels are equally good and the linear kernel is not as competitive. The ups and downs of the mean values are due to the sigmoidal kernel results, which performs very different depending on the involved descriptors.

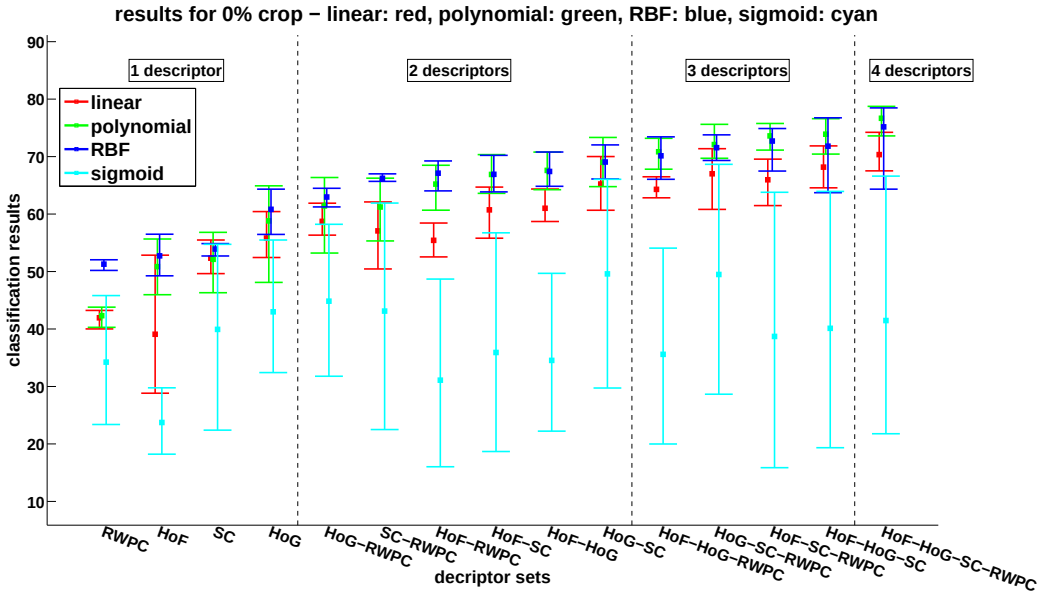


Figure 6.2: *Results overview: adding descriptors increases performance. Polynomial (green) and rbf (blue) kernels perform better than the linear (red) kernel and the sigmoidal (cyan) kernel performs worst. The deviation for the sigmoidal kernel under different parameter sets is large, while for the other kernels the deviation is smaller.*

6.2.2 Influence of Tracklet Cuts

For the tracklet cuts, various parameterizations were tested. Although the performance is slightly better when cropping the first frames of a players activity, a significant performance of more than a few percents could not be observed. The maximum improvement for best results between 0% and 50% is only 2.41% (78.74% versus 81.15%), the average difference is even smaller. This might be due to the strong influence of context features (SC) and location cues (RWCP) which can make up for movement variabilities. This consideration is further supported, as only when using the HOF features alone the results improve significantly due to the reduced variability in activity execution and when combining features the difference vanishes. See Figures 6.3 (20% crop), 6.4 (33%) and 6.5 (50%) for comparison. The proposed method seems to handle movement noise well through information encoding by four very different descriptors. Figure 6.6 contains all information about the different tracklet cuts within one single figure and makes obvious that the frames at the beginning or before an activity are well modeled by the proposed method (and such the descriptors) so that a wide performance variance could not be observed.

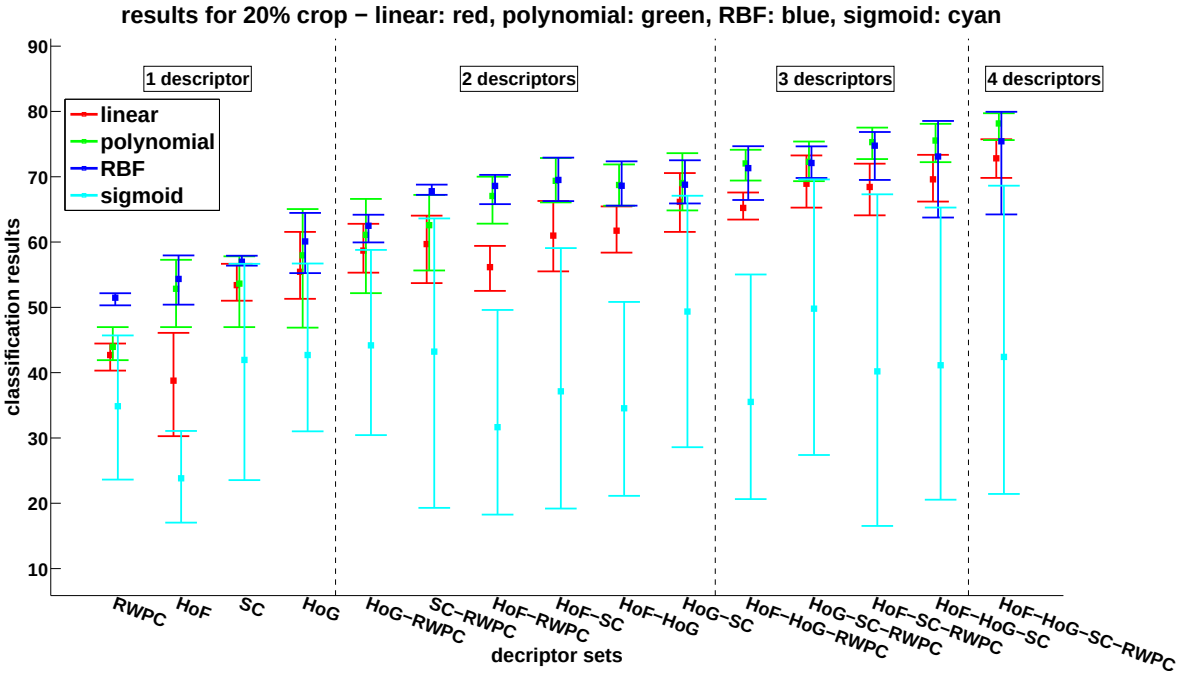


Figure 6.3: *Crop results for 20% removal*

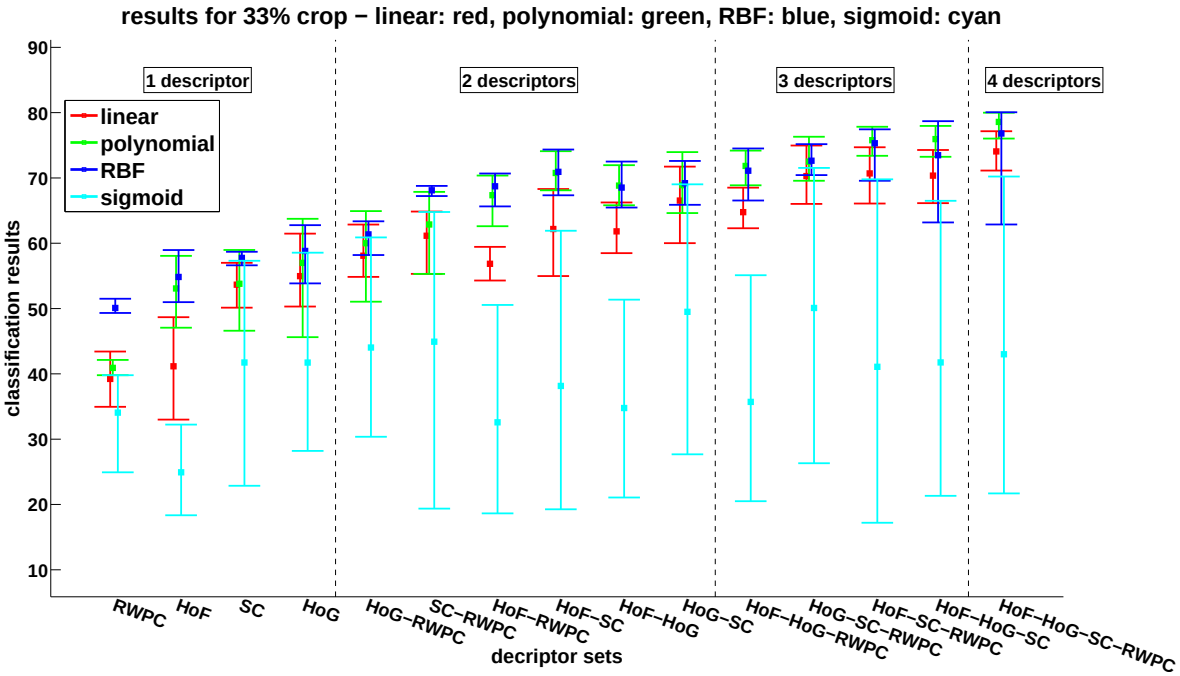


Figure 6.4: *Crop results for 33% removal*

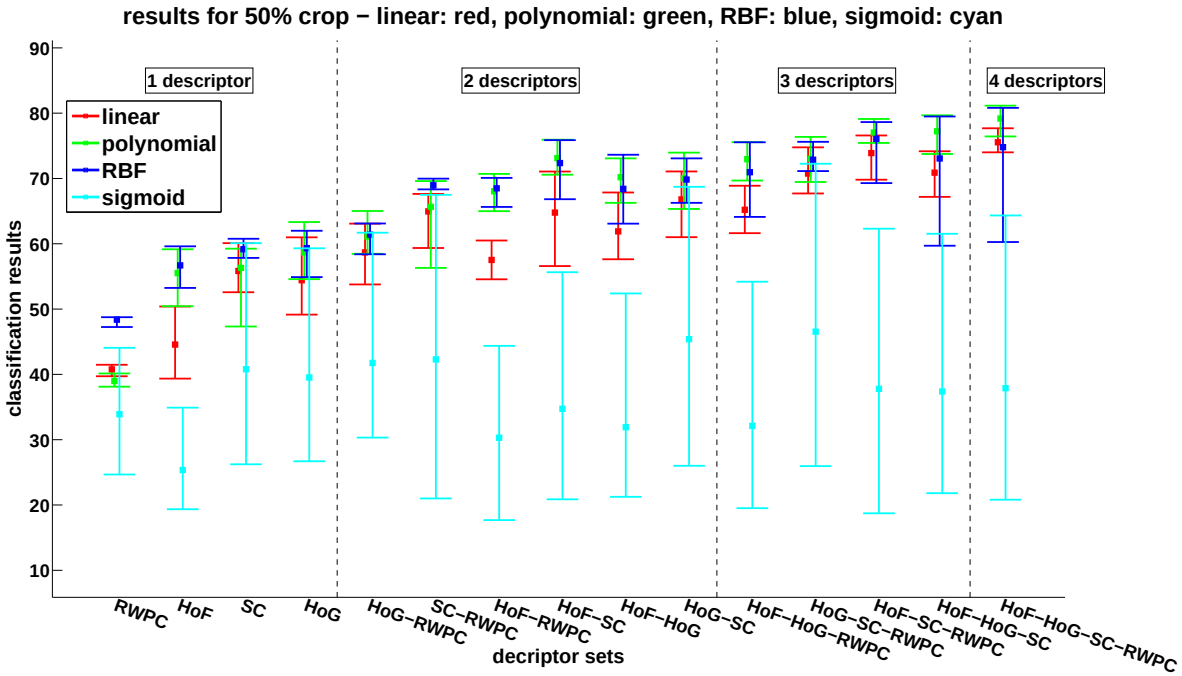


Figure 6.5: *Crop results for 50% removal*

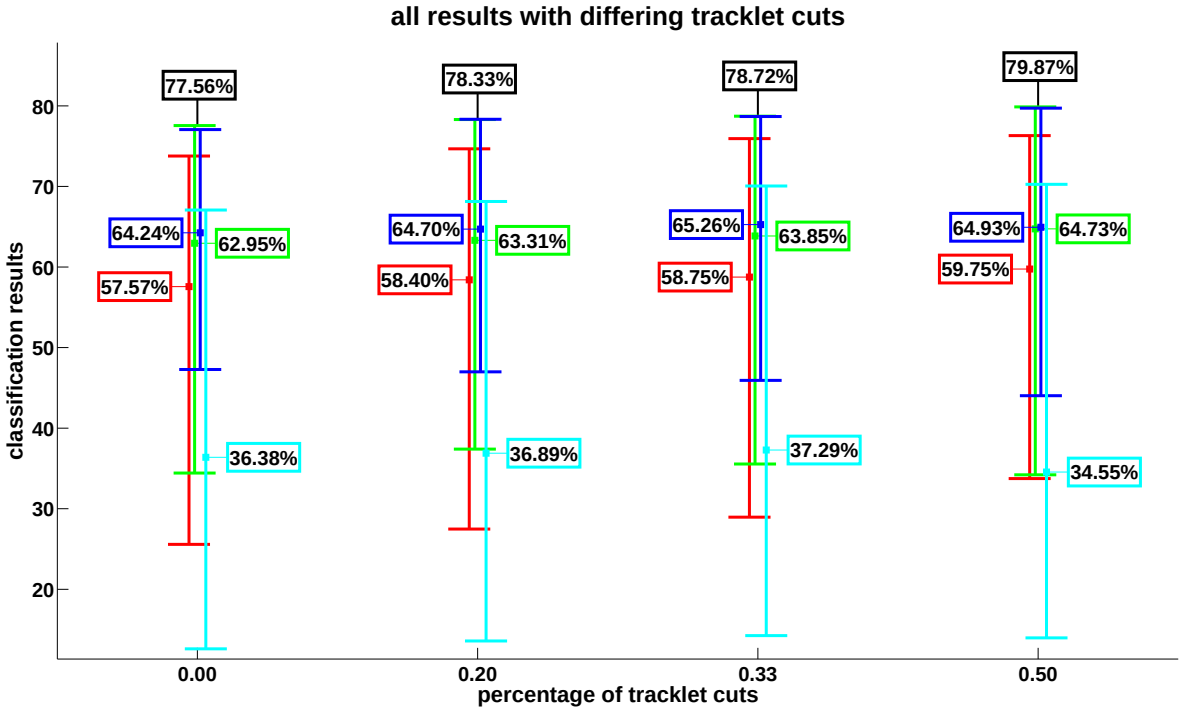


Figure 6.6: *Results for all evaluated tracklet cuts within one figure: The results are quite similar and the improvement of only 2.31% shows that the movement noise at the beginning of the activities does not negatively impact the classification process. Again, the kernels are color coded (linear: red, polynomial: blue, RBF: green, sigmoid: cyan).*

6.2.3 Influence of SVM Parametrization

As the previous part showed, the choice of one of the four tested SVM kernels has big influence on the classification results. Also the previously presented parameters c and γ are vital for good classification results. For the three evaluated SVM parameter sets $5.66/1.05$, $32.00/0.19$ and $181.02/0.03$ all results are collected in Figure 6.7. Obviously some kernels are more capable of adaption to data than others, as the average performance of rbf and polynomial kernel only differs slightly while the linear kernel works best with a small c and large γ value and the sigmoidal kernel performs in contrary fashion (nest with big c and small γ value).

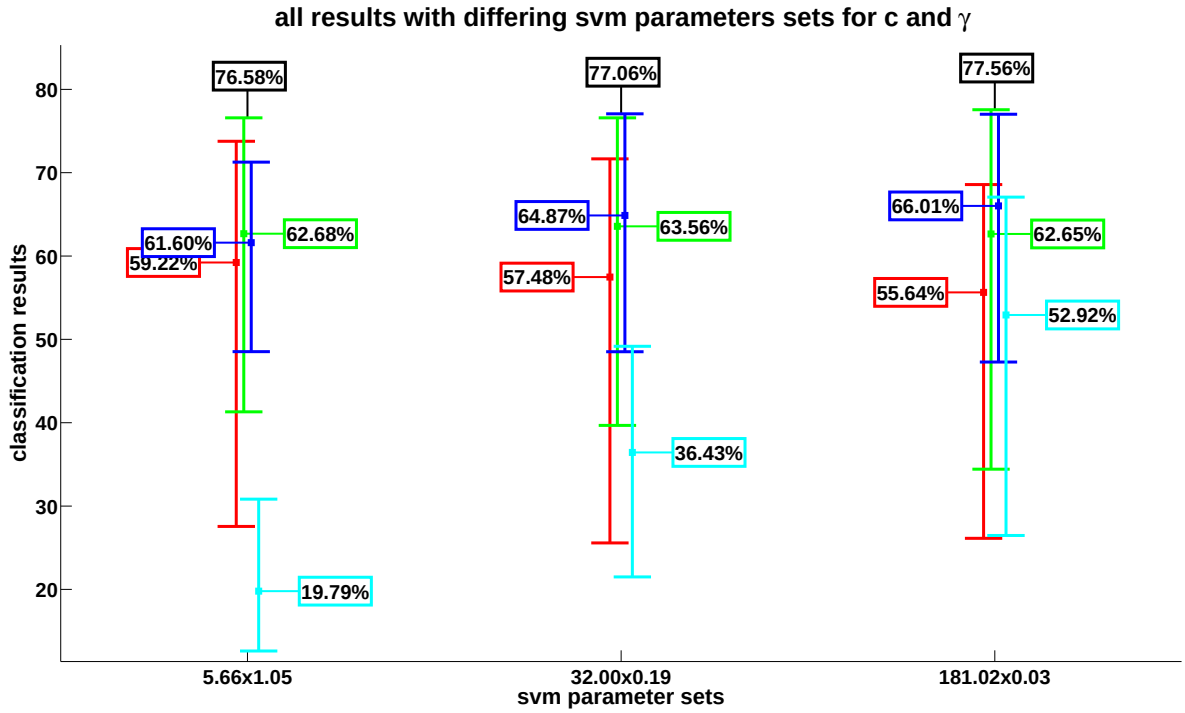


Figure 6.7: *Results for different SVM parameter sets: Best performance is achieved with* $c = 181.02$ *and* $\gamma = 0.03$*. Differences of rbf and polynomial kernel are rather small compared to linear or sigmoidal kernels, which perform in contrary manner. (linear kernel: red, polynomial kernel: blue, RBF kernel: green, sigmoidal kernel: cyan, no tracklet cut)*

6.2.4 Influence of Descriptor Parametrization

As stated above, for every descriptor and all descriptor combinations experiments were conducted. The following pages will give a detailed overlook for each of these experiments. To present the results in a clear fashion, the results were made up in bundles corresponding to kernel types and the SVM parameter sets were color coded. The best results for each kernel as well as the overall average result for the descriptor (combinations) are marked.

Results Using 1 Descriptor

Testing single descriptors, with a top result of 63.46% and an average of 52.84% over all experiments the HOG descriptor seems to express the activities best. The other descriptors result in about 10% lower performance: RWPC yields 49.82%/38.07%, HOF 55.07%/39.82% and SC 53.77%/47.25%.

The best results of the SC and HOF descriptor are comparable and after all half of the activities can be classified correctly using only information about players projected positions onto the court plane (RWPC). Still the results show, that any of the descriptors alone is not capable of good classification. Figures 6.8 (RWPC), 6.9 (HOG), 6.10 (HOF) and 6.11 (SC) display overall results separated by kernel types and with best and average results over all conducted experiments indicated. In most cases the sigmoidal kernel is not competitive and depends strongly on the SVM parameters while the polynomial and RBF kernels perform best.

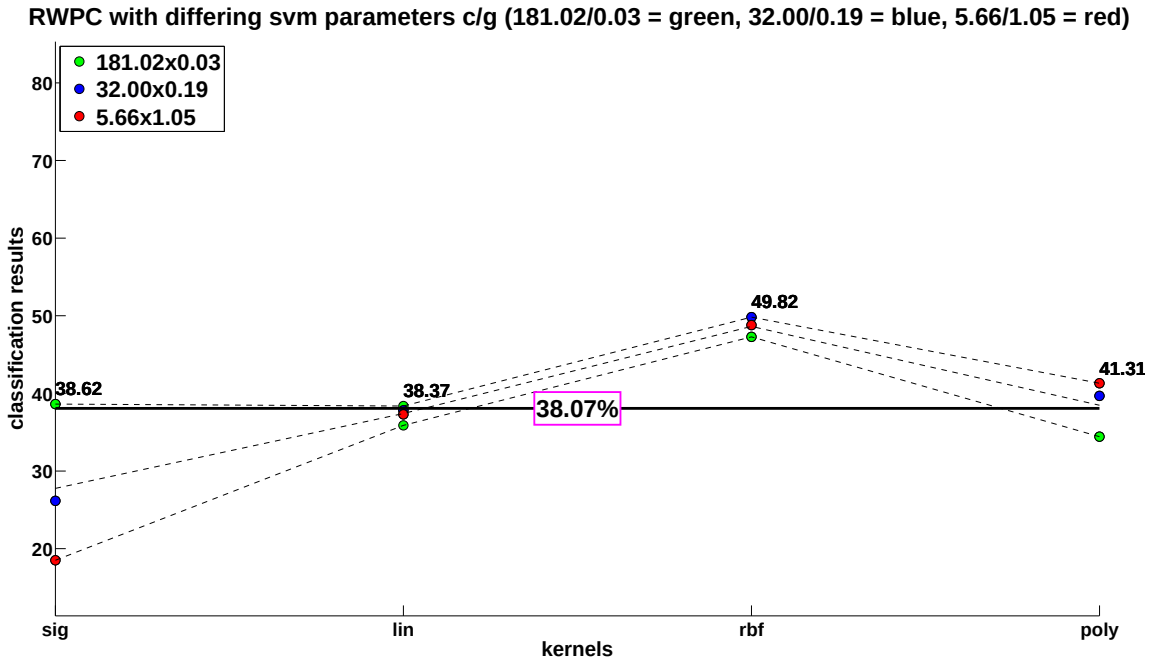


Figure 6.8: *Results for using RWPC descriptor only: Best results are achieved in combination with the rbf kernel with 49.82% at the top.*

Results Using 2 Descriptors

Combining two descriptors raises the average performance to around 70% with a top performance of 71.83% and an average result of 61.22% by combination of HOG and SC. The combinations of two descriptors deliver results that are not as differing as the single descriptor results. This shows, that every descriptor has potential in different areas and the combination reinforces the overall strength for recognition. Combinations with RWPC are slightly inferior to combinations of the other three descriptors. The following Table 6.4 shows the results in descending numbers and the Figures 6.12, 6.13, 6.14, 6.15, 6.16 and 6.17 show all results in a graphical overview.

descriptor combination	best result	average result
HOG-SC	71.83%	61.22%
HOF-HOG	69.47%	55.82%
HOF-SC	69.39%	55.76%
HOF-RWPC	67.73%	52.62%
SC-RWPC	66.88%	55.77%
HOG-RWPC	65.59%	55.42%

Table 6.4: *Results for combinations of two descriptors.*

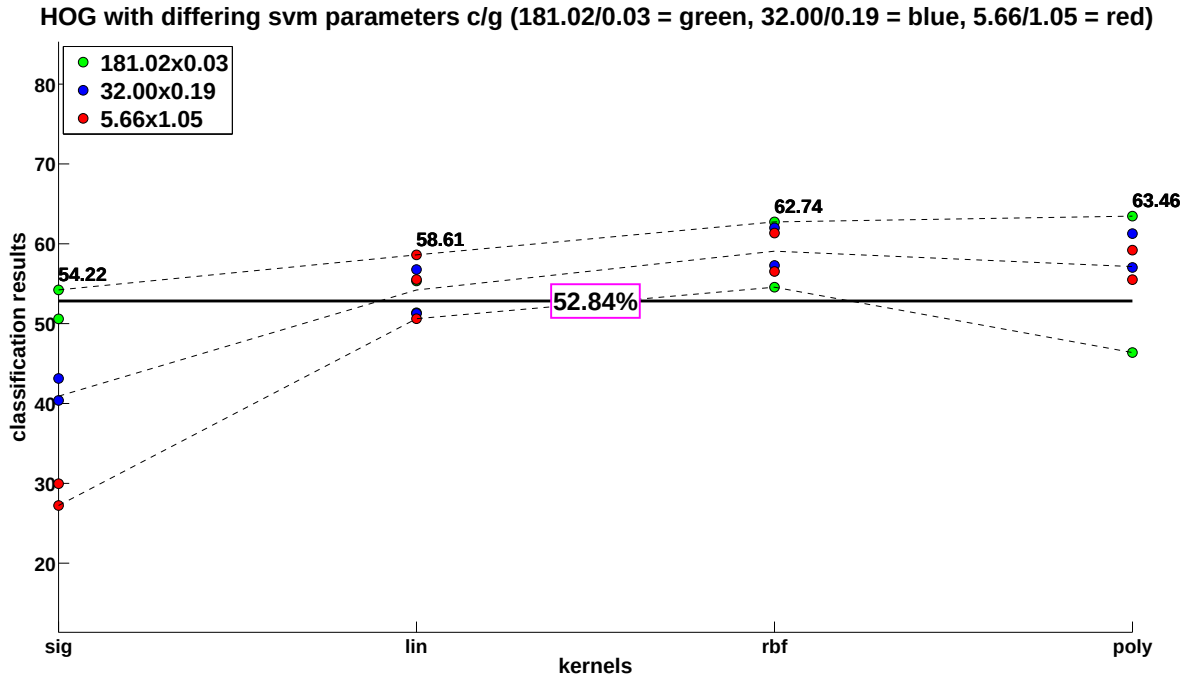


Figure 6.9: *Results for using HOG descriptor only: Best results are achieved in combination with the polynomial kernel with 63.46% at the top, which is also the best result obtained with a single descriptor.*

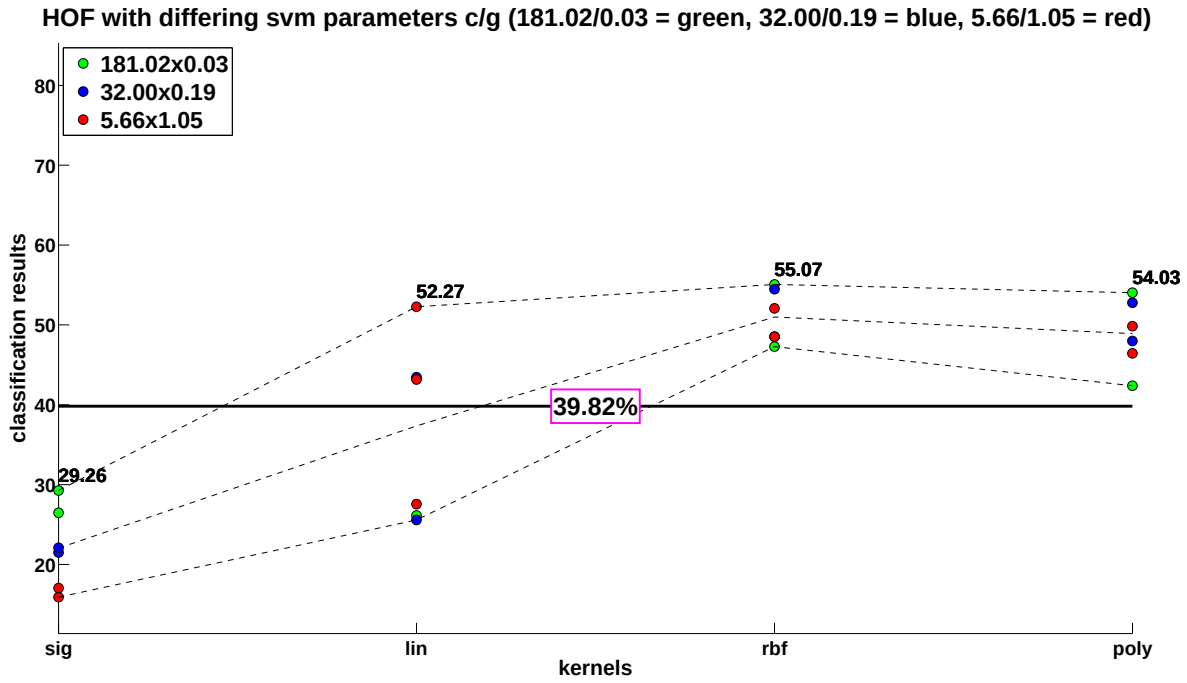


Figure 6.10: *Results for using HOF descriptor only: Best results are achieved in combination with the rbf kernel with 55.07% at the top. Results with sigmoidal and linear kernels are clearly inferior.*

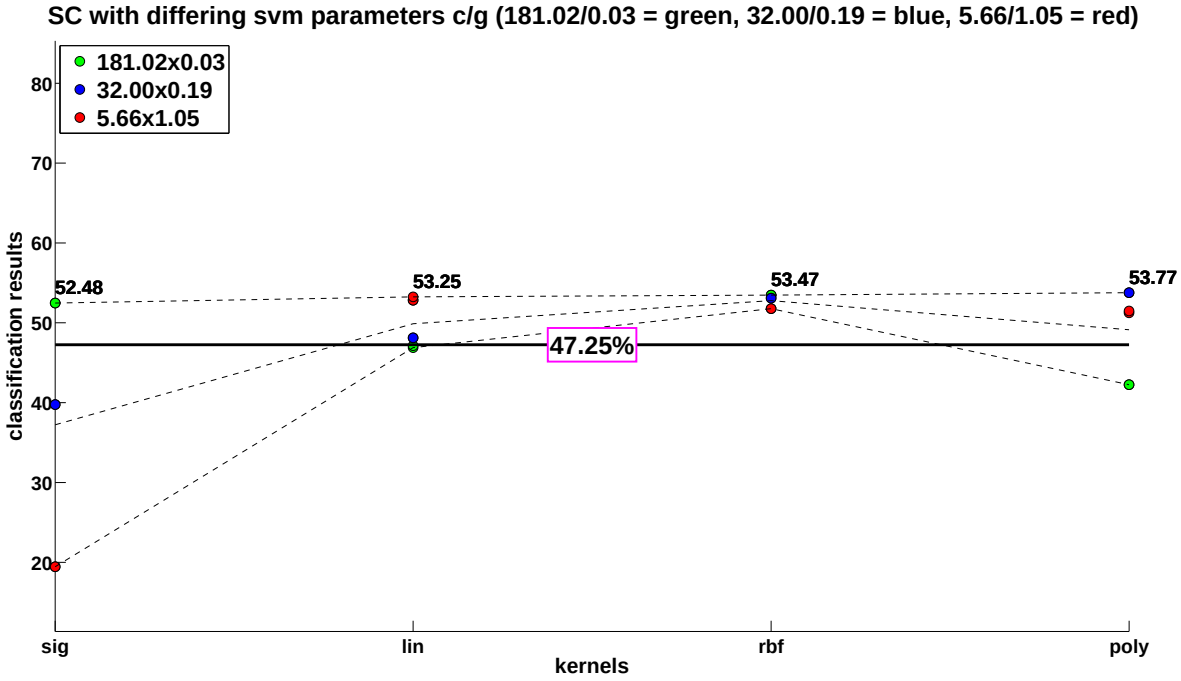


Figure 6.11: *Results for using SC descriptor only: Best results are achieved in combination with the polynomial kernel with 53.77% at the top. Obviously this descriptor is not as strongly dependent on the chosen kernel than others.*

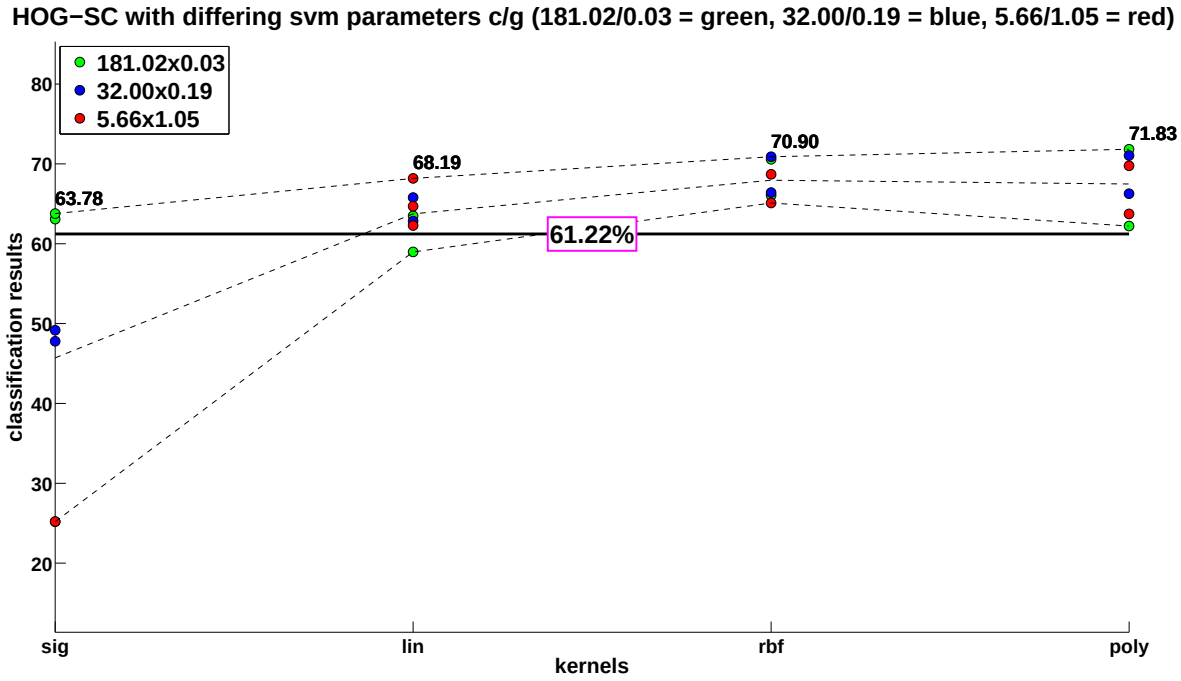


Figure 6.12: *Results for using HOG and SC: Three kernels are comparable, only the sigmoidal kernel varies. This combination is best, both on average (6% or better than the other combinations) and top (over 2% better than the next best) result.*

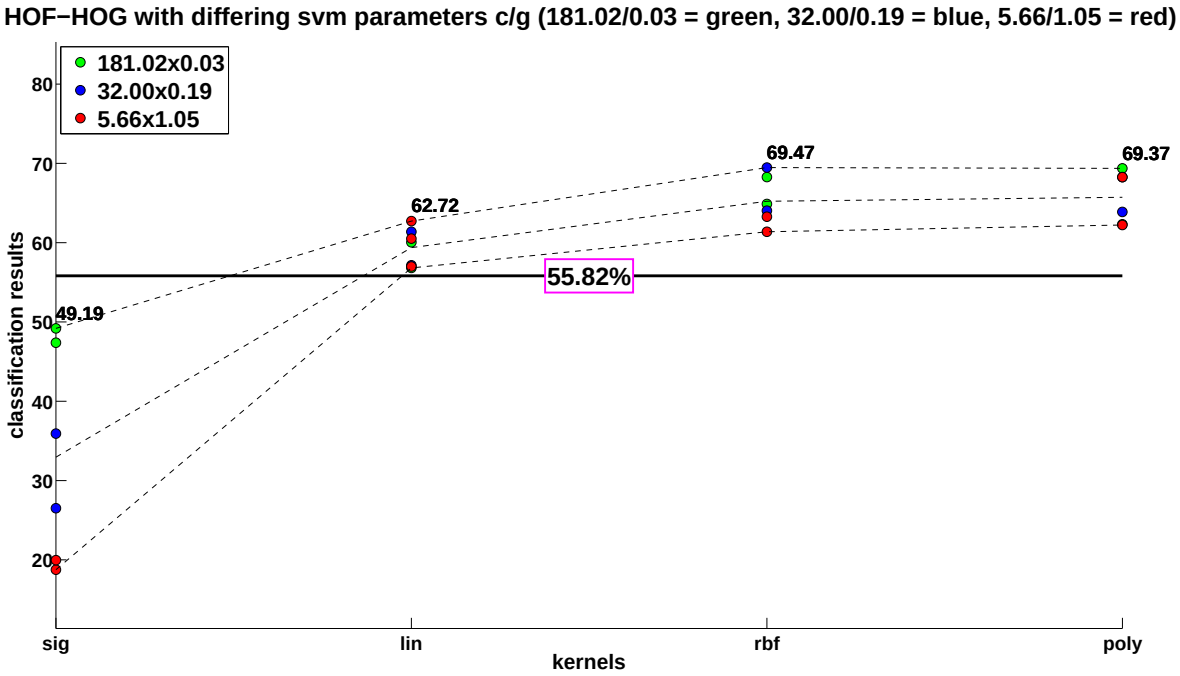


Figure 6.13: *Results for using HOF and HOG: Compared to HOG/SC, the best result of this popular descriptor combination has a 2.36% lower score. Still a 5-14% increase compared to the results of HOF or HOG alone.*

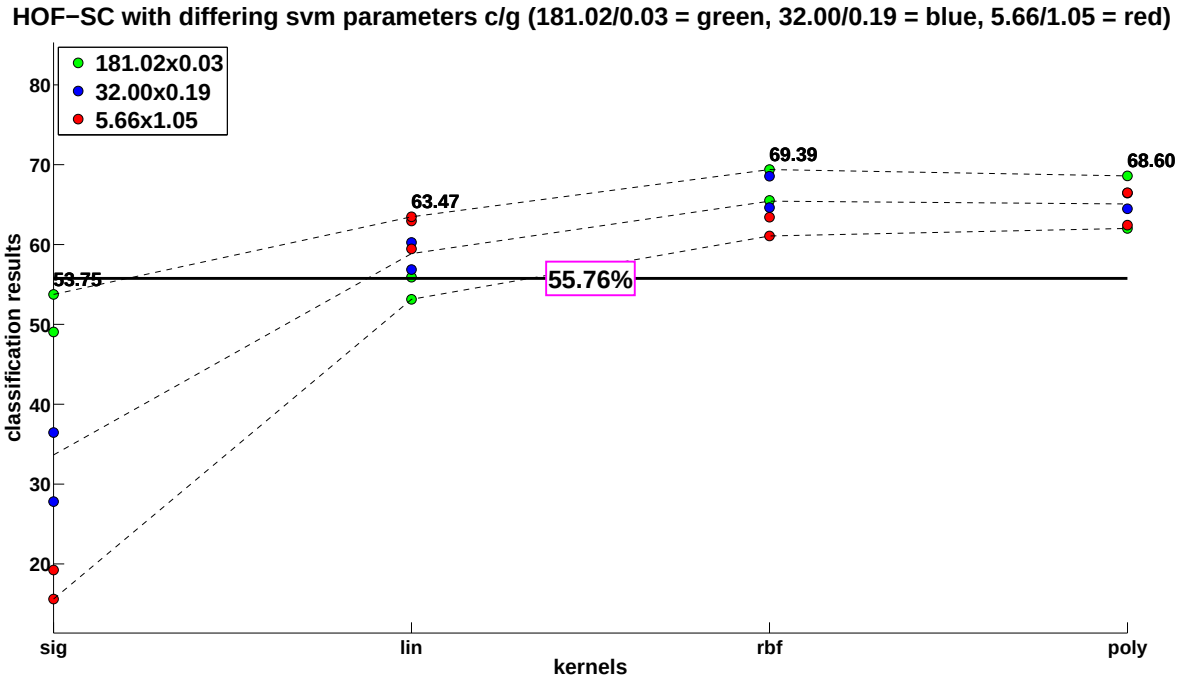


Figure 6.14: *Results for using HOF and SC: This pairing performs almost identical to the HOF/HOG result, pointing out the relevance of context information for activity recognition.*

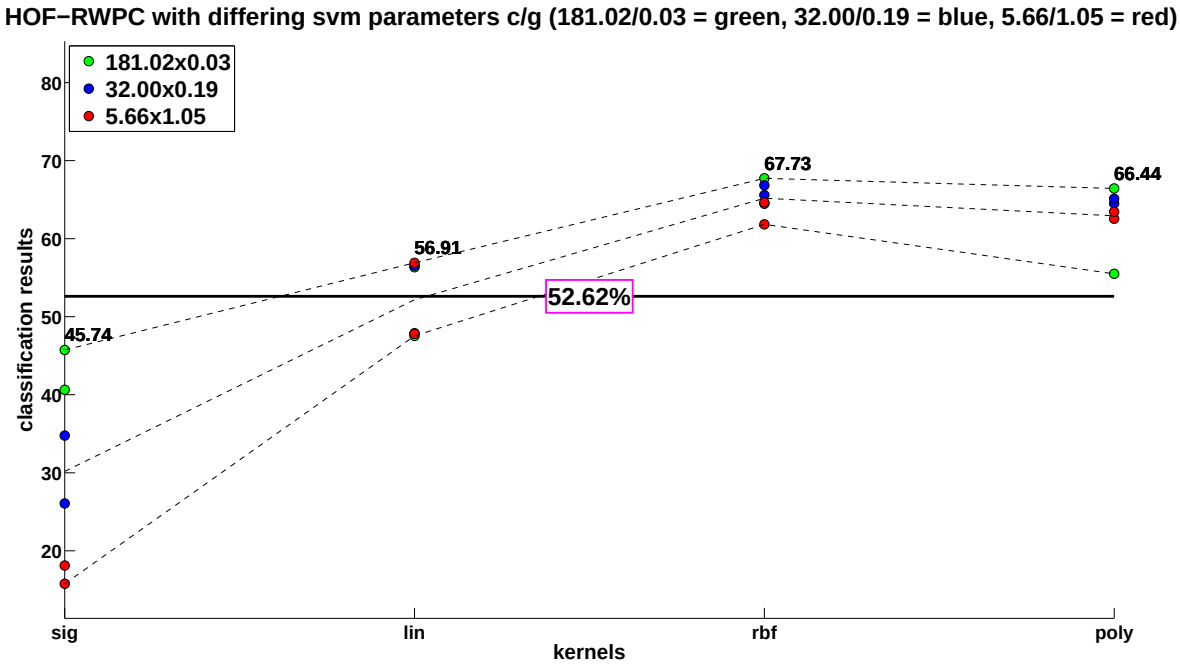


Figure 6.15: *Results for using HOF and RWPC: With 67.73% top result, this combination is not too far from the other HOF combinations (HOF/HOG, HOF/SC). On average however the results are slightly worse.*

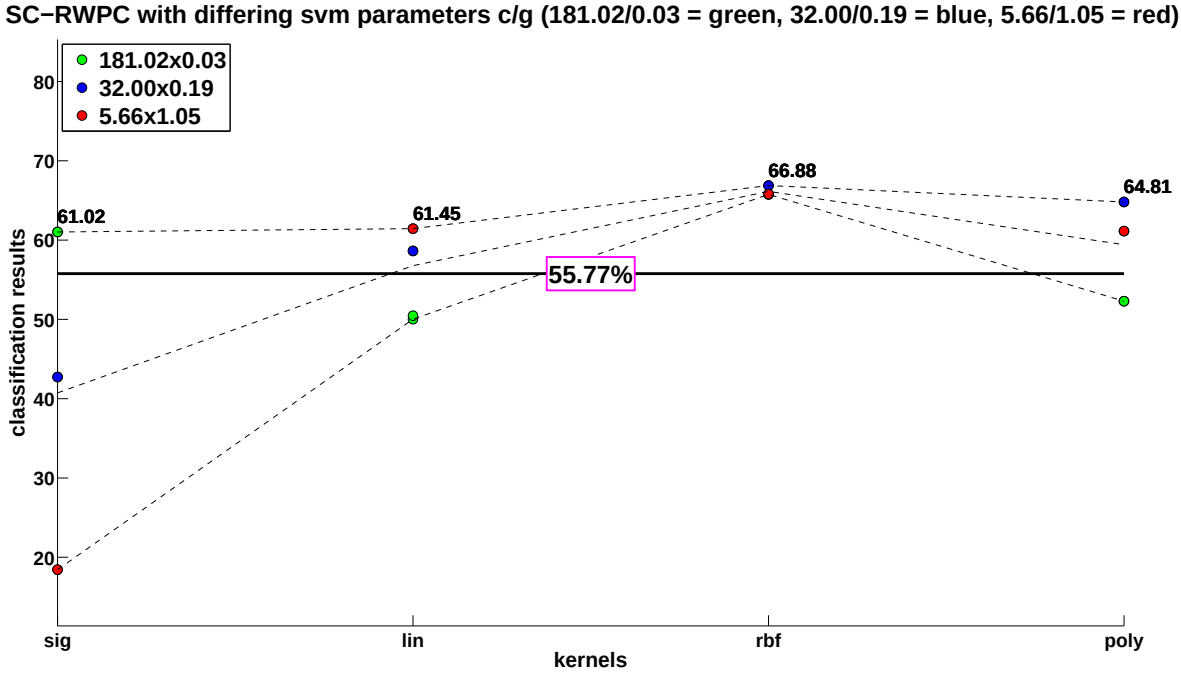


Figure 6.16: *Results for using SC and RWPC: Position information alone does not provide the same quality of information for recognition. Although the average is comparable, the best result is more than 6% worse than the HOG/SC combination.*

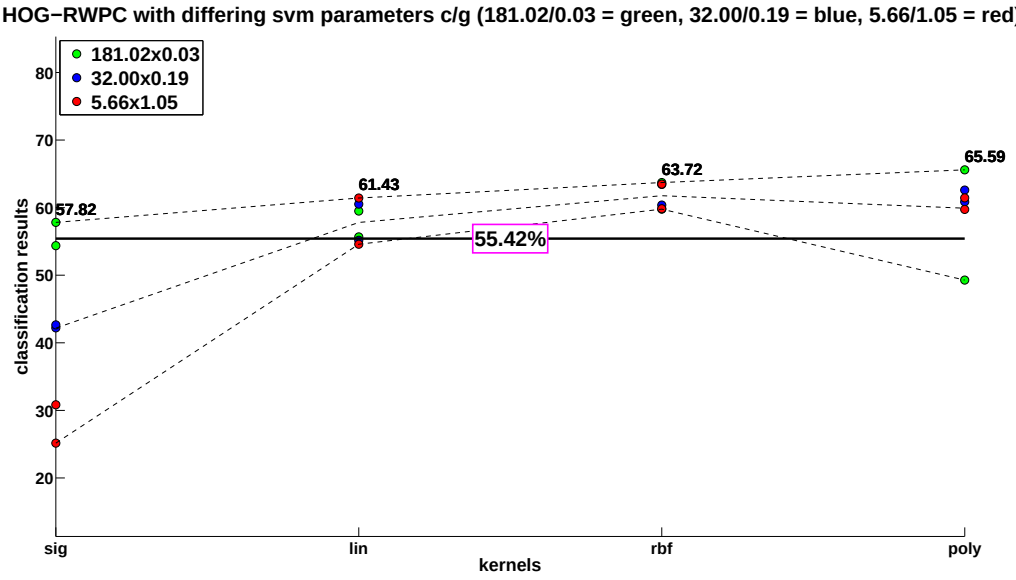


Figure 6.17: *Results for using HOG and RWPC: Surprisingly, the winner of the single descriptor evaluation (HOG) is worst when combined with RWPC. As the result is comparable to the SC/RWPC combination, the RWPC descriptor seems to be less discriminative than the other three descriptors.*

Results Using 3 Descriptors

Adding one more descriptor to the test set increases performance around 75%. The increase in performance is not so evident as before, still any combination of an three descriptors is superior to the best result with two descriptors, meaning that additional exploitable information has been added. Unsurprisingly the best combination is made up of the three best descriptors from before. HOF, HOG and SC yield 75.05% at best, while the top average result is achieved by the combination of HOG, SC and RWPC (63.57%). Again, the rbf and polynomial kernels are superior to the others. Table 6.5 shows average and best results and Figures 6.18, 6.19, 6.20 and 6.21 show the according graphical interpretations.

descriptor combination	best result	average result
HOF-HOG-SC	75.05%	61.54%
HOG-SC-RWPC	74.46%	63.57%
HOF-SC-RWPC	74.36%	61.62%
HOF-HOG-RWPC	72.13%	58.74%

Table 6.5: *Results for combinations of three descriptors.*

Results Using all Descriptors

Finally, using all four descriptors yields the best result of 77.56% and again increase in classification performance. Thus, any of the used descriptors contains information for the recognition task that cannot be fully supplied by another descriptor. Figure 6.22 shows the results with an overall average of 64.52%. Again, the rbf and polynomial kernels perform best while the sigmoidal kernel is strongly dependent on the choice of the SVM parameters c and γ.

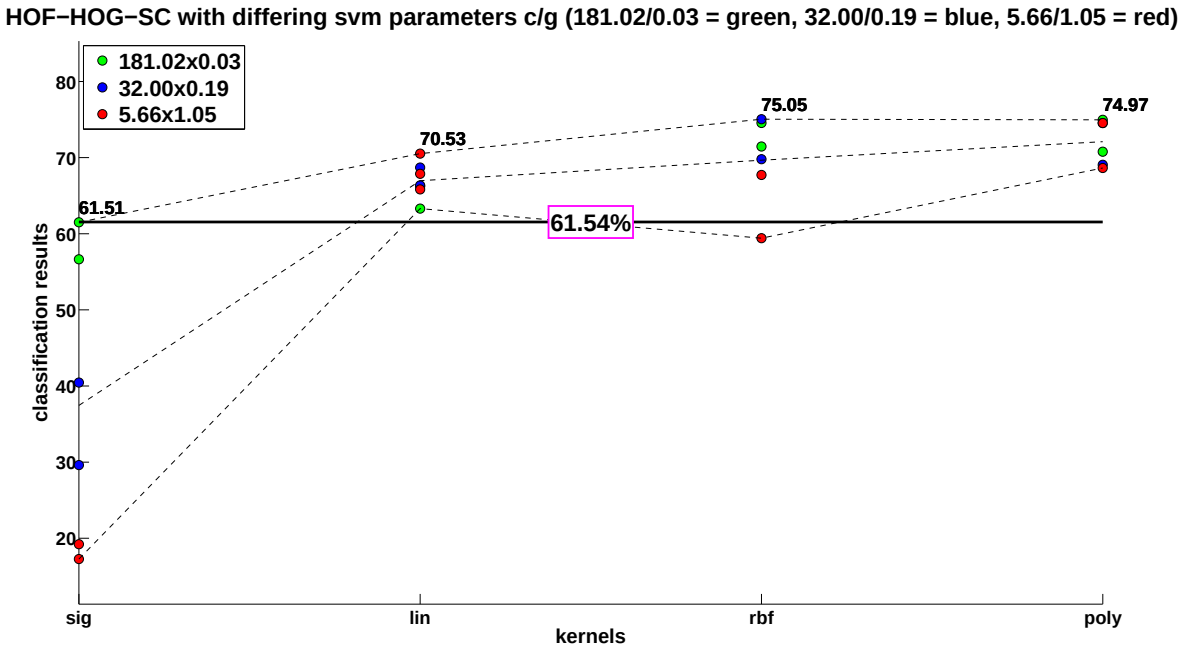


Figure 6.18: *Results for using HOF, HOG and SC: With 75.05%, this combination of three descriptors adds some more percentage to the correct classified data.*

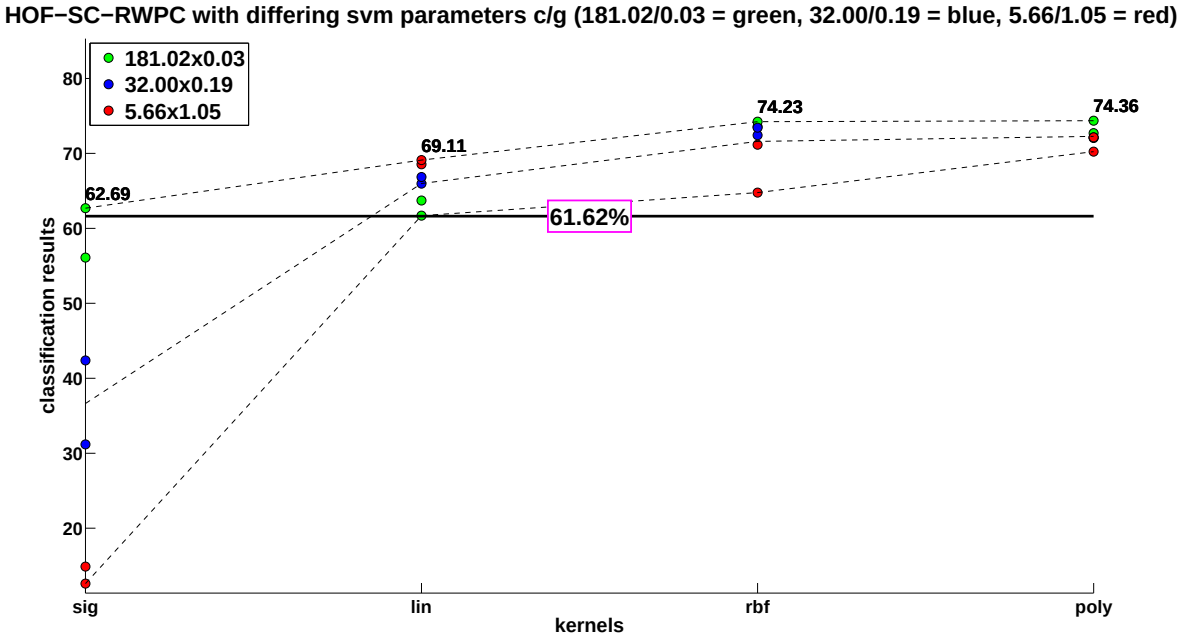


Figure 6.19: *Results for using HOF, SC and RWPC: With 74.36% there is again clear improvement compared to the twin combinations.*

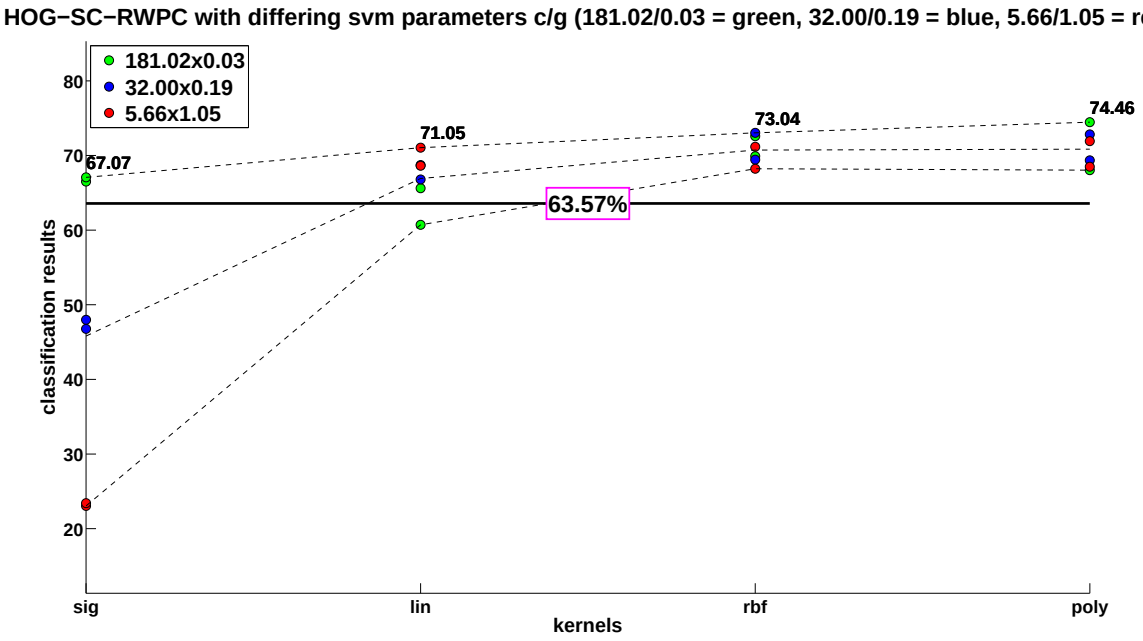


Figure 6.20: *Results for using HOG, SC and RWPC: This combination is comparable to the above one. It is remarkable, that the average performance of this combination is best.*

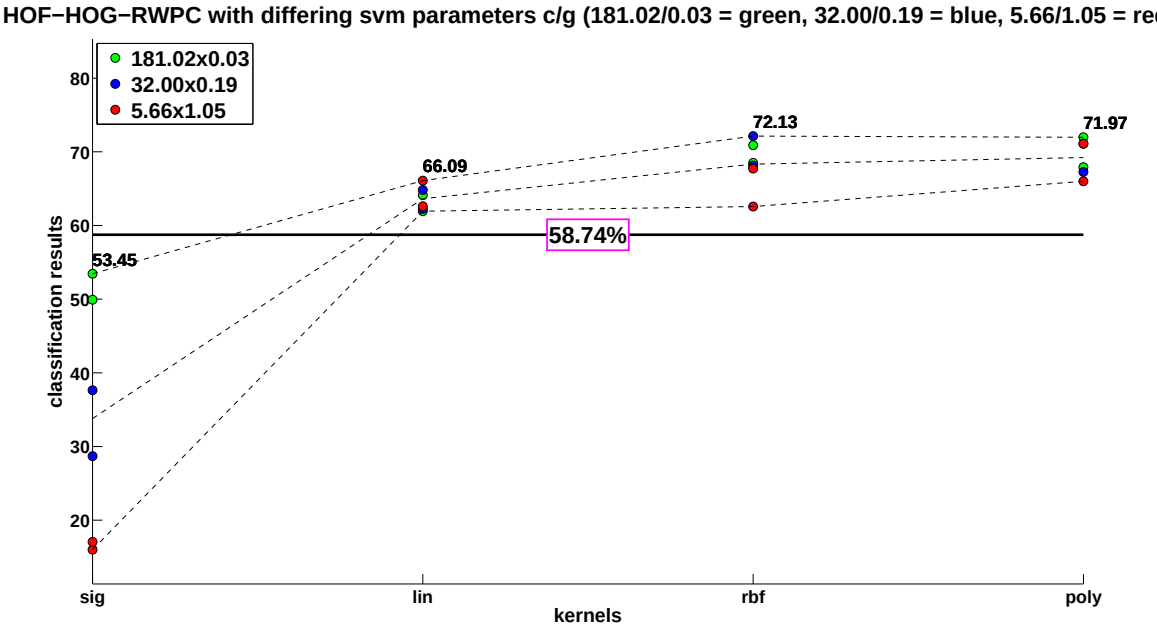


Figure 6.21: *Results for using HOF, HOG and RWPC: Again, the combination of RWPC/HOG (and HOF) is worst. Both, average and top result, are clearly inferior to the other combinations.*

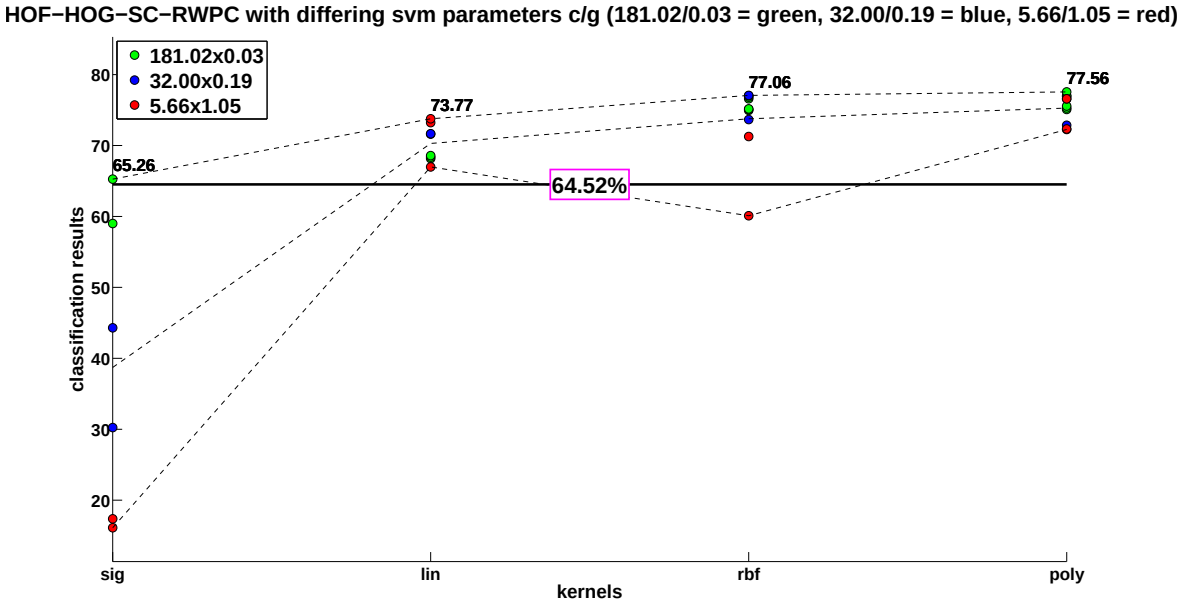


Figure 6.22: *Results for using all four descriptors: Adding another 2.5% to the former best result, the overall best result of 77.56% is achieved by combination of all four descriptors. Also, the average over all experiments is higher than for any other descriptor combination.*

Summary of Descriptor Combinations

A detailed table of all results from the descriptor combinations is shown in Table 6.6. The table lists all 15 possible combinations along with the four evaluated SVM kernels. For every kernel, the minimum, maximum and average result on a descriptor set is displayed as different configurations for the HOG, HOF and SC parameters were evaluated. Overall averages for descriptor sets are listed in the outright column, while overall averages for the kernel types are displayed in the lowest row of the table.

As stated before, the polynomial and rbf kernels perform better on any test set than the linear and sigmoidal kernels. On average, both kernels are equally good, with 66.71% for the rbf kernel and 66.17% for the polynomial kernel. The overall best result of 77.56% with all four descriptors used is achieved with a polynomial kernel, but is just marginally superior to the rbf result of 77.06%. While the linear kernel is about 5% worse on average for the best results over the descriptor sets, with 54.39% the sigmoidal performance is approximately 12% worse. Considering the averages over all kernels (outright column in the table), it is obvious that using more descriptors improves results.

6.2.5 HOG/HOF Parametrization Results

The parameter for the HOG and HOF descriptors, as visualized before in Figure 6.1, have also been evaluated. As the first results showed, that a small subdivision in cells gave resulted in low recognition rates, the testing of the parameters was subsequently limited to cell sizes of 70 and 88 and window sizes of 210×280 and 176×246 respectively. Figure 6.23 gives an overview and shows that the difference of 1.53% between the parameter configurations is negligible and the configuration with cell size 70 (78.74%) is slightly better compared to that with cell size 88 (77.21%). For linear, polynomial and RBF kernels, the 70/280x210 configuration is 2-3% better. Only for the sigmoidal

descriptor set	lin			poly			rbf			sig			overall
	min	max	avg	min	max	avg	min	max	avg	min	max	avg	
SC	46.89	53.25	49.88	42.25	**53.77**	49.13	51.77	53.47	52.78	19.45	52.48	37.23	47.25
HOF	25.58	52.27	37.35	42.38	54.03	48.91	47.28	**55.07**	50.99	15.90	29.26	22.04	39.82
HOG	50.62	58.61	54.22	46.38	**63.46**	57.15	54.57	62.74	59.08	27.22	54.22	40.92	52.84
RWPC	35.88	38.37	37.41	34.43	41.31	38.47	47.28	**49.82**	48.63	18.51	38.62	27.76	38.07
HOF-SC	53.14	63.47	58.87	62.02	68.60	65.08	61.07	**69.39**	65.43	15.60	53.75	33.65	55.76
HOG-SC	58.97	68.19	63.75	62.22	**71.83**	67.48	65.10	70.90	67.96	25.19	63.78	45.71	61.22
HOF-HOG	56.81	62.72	59.37	62.23	69.37	65.72	61.38	**69.47**	65.22	18.76	49.19	32.95	55.82
SC-RWPC	50.02	61.45	56.77	52.30	64.81	59.41	65.76	**66.88**	66.15	18.44	61.02	40.73	55.77
HOF-RWPC	47.53	56.91	52.17	55.50	66.44	62.93	61.84	**67.73**	65.19	15.77	45.74	30.18	52.62
HOG-RWPC	54.59	61.43	57.81	49.28	**65.59**	59.92	59.78	63.72	61.77	25.16	57.82	42.17	55.42
HOF-HOG-SC	63.30	70.53	66.97	68.64	74.97	72.09	59.42	**75.05**	69.67	17.27	61.51	37.45	61.54
HOF-SC-RWPC	61.70	69.11	65.98	70.23	**74.36**	72.27	64.77	74.23	71.60	12.61	62.69	36.64	61.62
HOG-SC-RWPC	60.71	71.05	66.92	68.03	**74.46**	70.85	68.23	73.04	70.73	23.05	67.07	45.80	63.57
HOF-HOG-RWPC	61.94	66.09	63.64	66.00	71.97	69.22	62.57	**72.13**	68.32	15.99	53.45	33.79	58.74
HOF-HOG-SC-RWPC	66.98	73.77	70.30	72.26	**77.56**	75.28	60.10	77.06	73.77	16.12	65.26	38.72	64.52
average	52.98	61.81	57.43	56.94	66.17	62.26	59.39	**66.71**	63.82	19.00	54.39	36.38	54.77

Table 6.6: *Results of the spatial context activity recognition (without cropping): polynomial and rbf kernels perform best. Adding descriptors improves performance, all four combined yield the best result.*

kernel the configuration 88/256x177 is better by 2.5%. The other two configurations (32/320x160 and 64/320/192) were omitted during testing for reduction of configuration possibilities, as the first results with a finer subdivision were not promising and the two coarser configurations were considerably better.

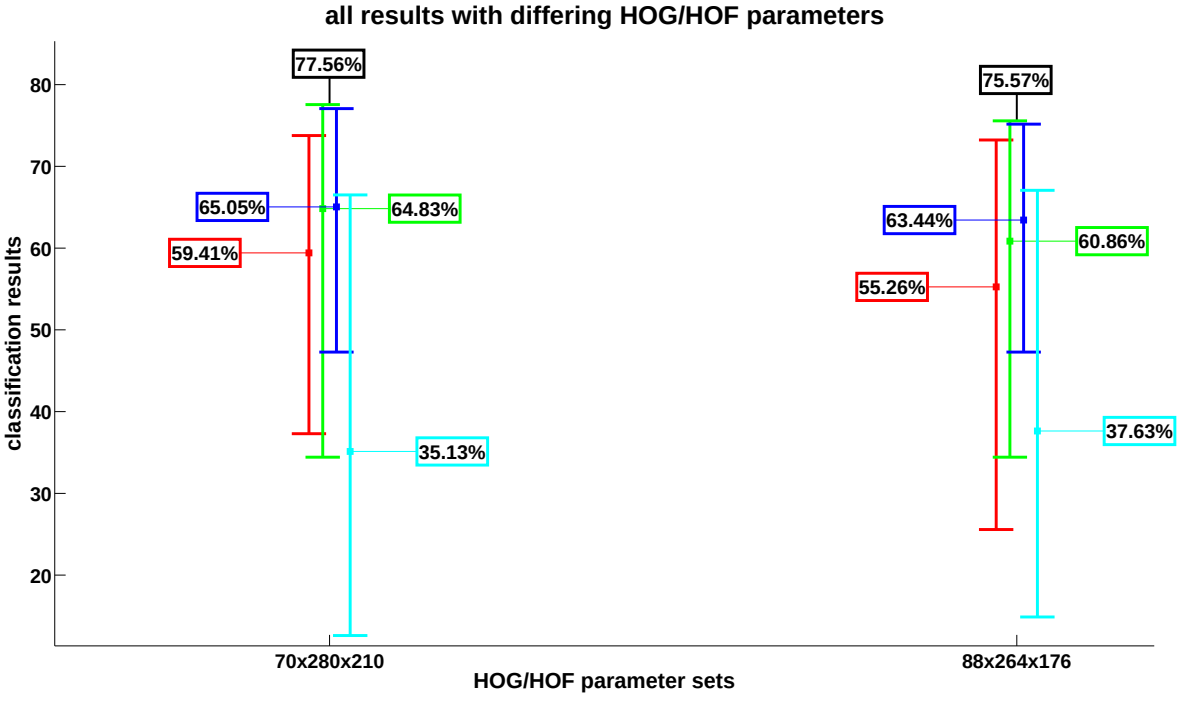


Figure 6.23: *Results for differing HOG/HOF parameters: The combination of cell size 70 and window size of* 280×210 *is apparently better for three of the kernels (linear, polynomial, rbf) than* $88/264 \times 176$*. The average result over all kernels is better by 2%. The two other configurations* $(32/320 \times 160, 64/320 \times 192)$ *were not competitive and omitted to reduce the number of conducted experiments.*

6.2.6 Detailed Results for Spatial Context Player Activity Recognition

While the previous pages displayed average results over all classes, kernels, SVM parameters, tracklet cuts, this part shows and discusses some results in detail. The following Figure 6.24 shows detailed class performance for the best result achieved with spatial context activity player recognition. The best results are received for the classes "Service" with 91.01% and "Block" with 92.96% correct classified activities. The next best performance is on class "Attack" with 82.74%, followed by three classes with about 75% performance: "Stand" (75.37%), "Reception" (73.27%) and "Setting" (74.95%). Only the very inhomogeneous and such difficult class "Defense/Move" obtains a score of 52.36%. On second sight, the confusion of this class with the "Reception" class is apparent. Almost 18% are misclassified in both ways, equally leveling down results of "Reception" and "Defense/Move". The reason for this correlation is that the movements in "Reception" and "Defense" are quite similar, as once the players try to decelerate the service and once the attack of an opponent player.

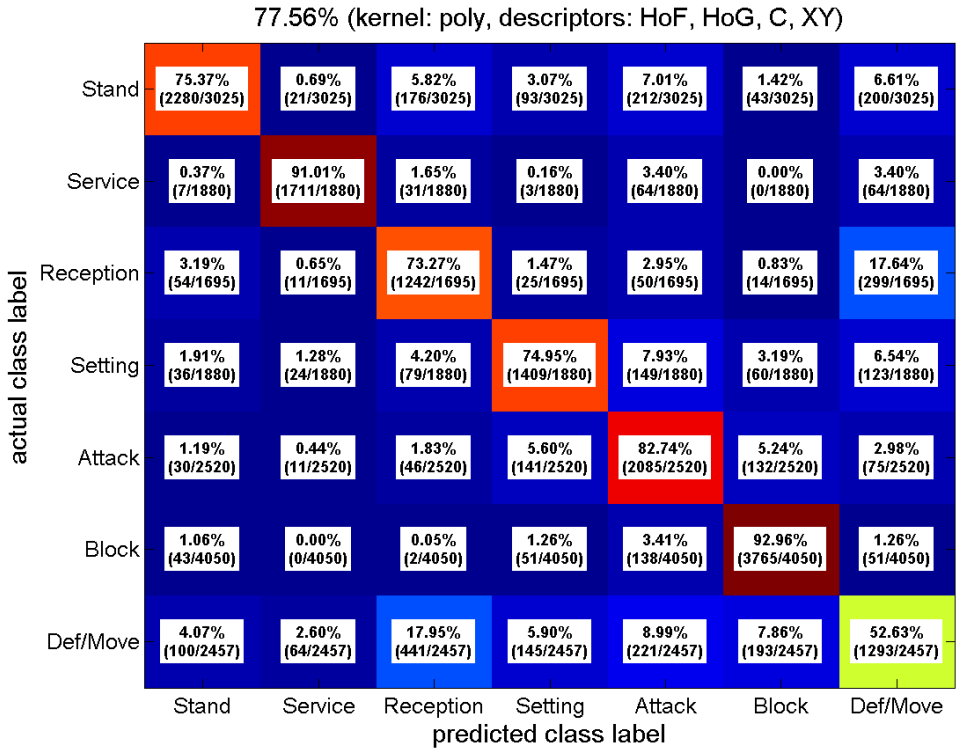
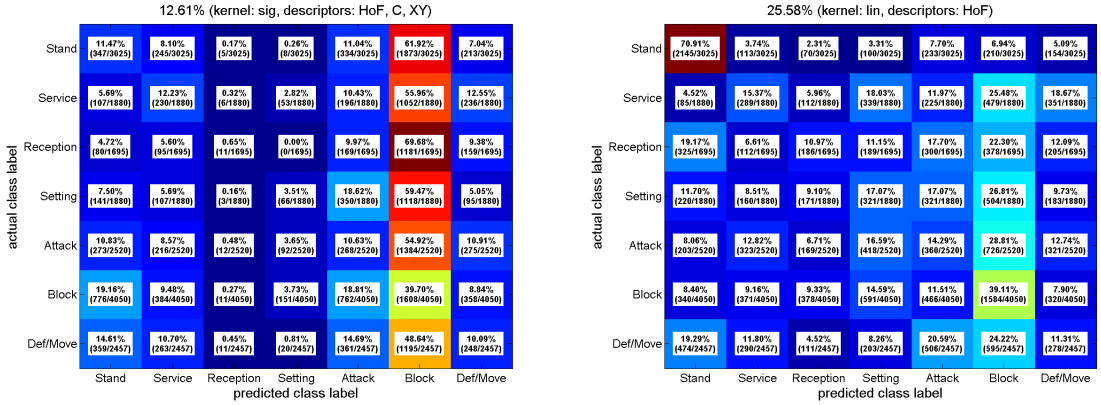


Figure 6.24: *Best result from spatial context activity recognition: Only the very inhomogeneous class "Move/Defense" has a low performance of 52.63%. The other six classes perform rather well, from 73.27% to 92.96% correct classified activities. The "Move/Defense" class is mostly confused with the "Reception" class and vice versa. This is no surprise as these activities are often very similar in execution and position.*

Figure 6.25 shows the two worst results. While the first illustrates the worst result with sigmoid kernel (12.61%) and a three descriptor combination of HOF, SC and XY, the second displays the worst result generated from a linear SVM using HOF features (25.58%). For the sigmoidal kernel under this parameter configuration this result is meaningless. The linear kernel shows one good class performance for the "Stand" class, which is comprehensible as almost no motion is encoded. All other classes (except of "Block") are comparable to randomly choosing from 7 classes.



(a) **(b)**

Figure 6.25: *Worst results from spatial context activity recognition: (a) Using a sigmoid kernel and HOF, SC and XY features is with 12.61% average result the worst tested parameter set. All classes are strongly biased to the "Block" class, so that this result is useless and only an indication for the importance of parameter selection. (b) For linear kernel and HOF features only, the result is a little bit better with 25.58% correct classified average percentage. Only the "Stand" class is recognized well, probably due to the exploitation of (missing) motion in this class by the HOF descriptor. The other classes are presumably too similar and have to big intra-class variance to be linearly discriminated.*

Figures 6.26 to 6.40, present the mean performance of the descriptor combinations presented on the previous pages. Each of the figures contains one descriptor (combination) and the results for every class and averaged over all runs with different parameter sets. This should give an impression about the class-wise strengths of the descriptor sets independent of the parameters chosen. Bear in mind, that these figures express an overall average over all executed runs and while not being representative for the final results they give some general view about the operation of descriptors and their various combinations. Also, as presented in the previous Figures 6.9 to 6.22, the different kernels influence the results in varying manner and operate differently on the descriptor sets.

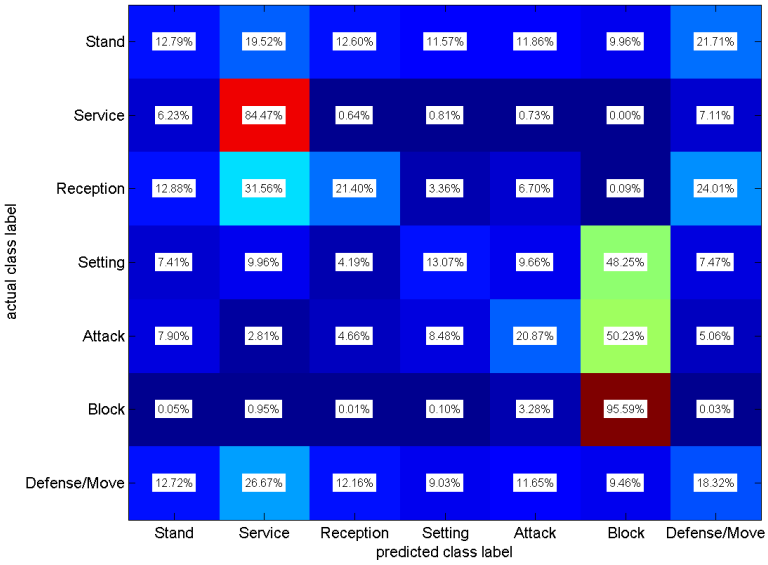


Figure 6.26: *Average over all runs with RWPC descriptor: Obviously it is hard to discriminate activities solely by the players position. This becomes apparent as "Setting", "Attack" and "Block" are all more or less classified into the class "Block" due to the proximity to the net. On the other hand, "Service" can be good discriminated from other classes as many positions are behind the court, opposite to all other classes. "Stand", "Move/Defense" and "Reception" can be distributed on the court such that a clear distinction is hard to make.*

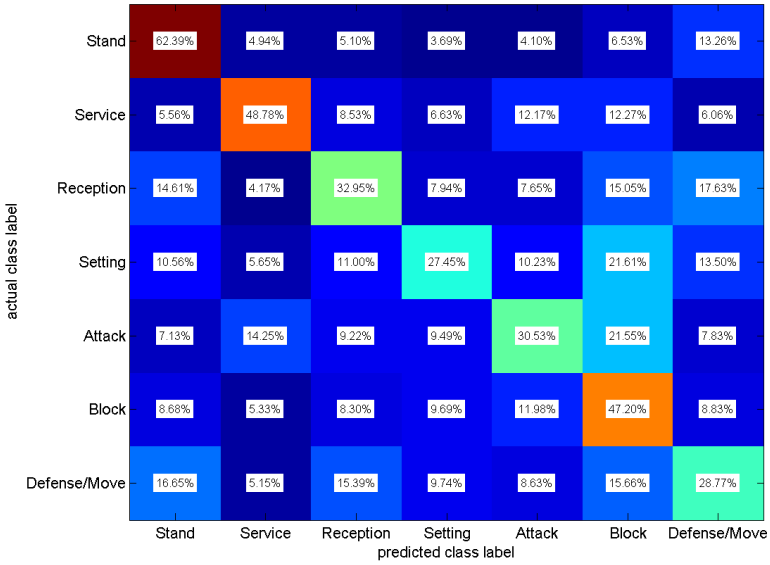


Figure 6.27: *Average over all runs with HOF descriptor: As expected, "Stand" can be discriminated best, as there is far less motion compared to any other activity. "Service" is well classified, maybe because of the proximity to the camera and such a slightly better characterization of the motion. "Block" is also rather good, as the players mostly only move in a up/down manner and with arms above the head. The other classes are very mixed concerning the movements and are such hard do differentiate.*

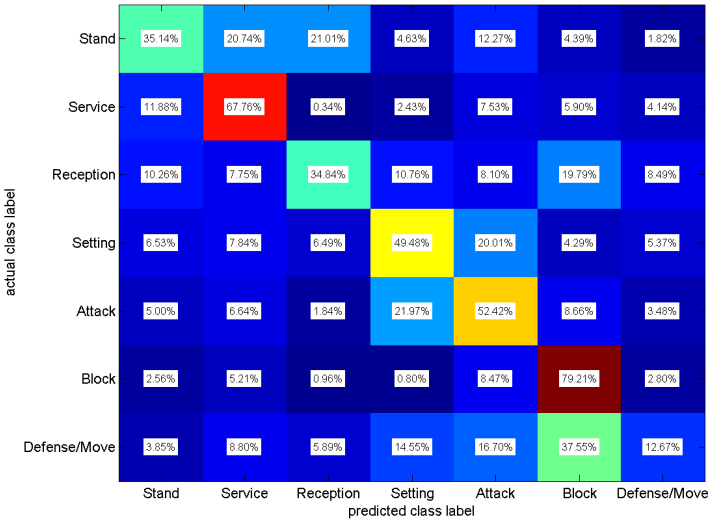


Figure 6.28: *Average over all runs with SC descriptor: "Service" is good distinguishable as the non serving players gather in the middle of the court awaiting the serve to take positions. "Block" also works good, as mostly in blocking situations the middle of the court is empty and players distribute at the borders of the court for defense. "Setting" and "Attack" have very similar player distributions, as the attackers move out of the court awaiting the set. This is also indicated by the confusion of 20% between those classes. The rest of the classes are harder to discriminate, as there exist multiple lineups for "Reception" and the two other classes do not follow any lineup or player distribution rules.*

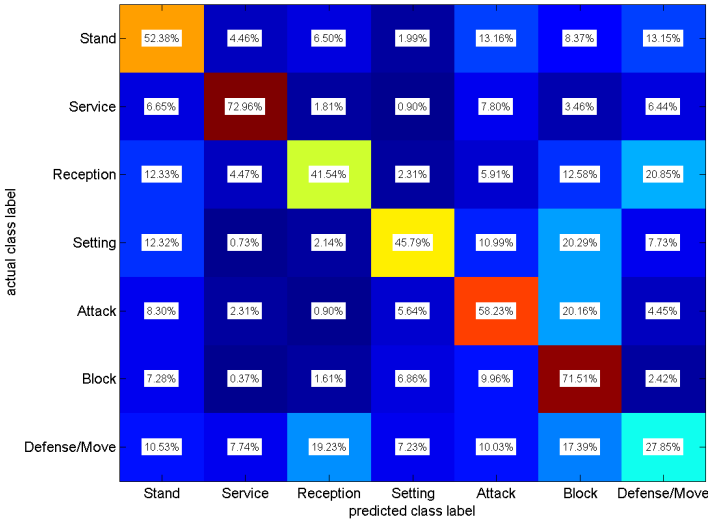


Figure 6.29: *Average over all runs with HOG descriptor: "Service" and "Block" are best discriminated, the first probably because of the proximity to the camera (similar to HOF) and the latter because of the straightened body form with raised hands above the head. "Stand" is rather good but confused with "Attack" and "Defense/Move" maybe due to similar upright body posture. "Attack" is mostly confused with "Block", again probably because of the similar body posture when jumping, the same way "Setting" is confused with "Block". "Reception" is often confused with "Move/Defense" due to similar stooped body positions.*

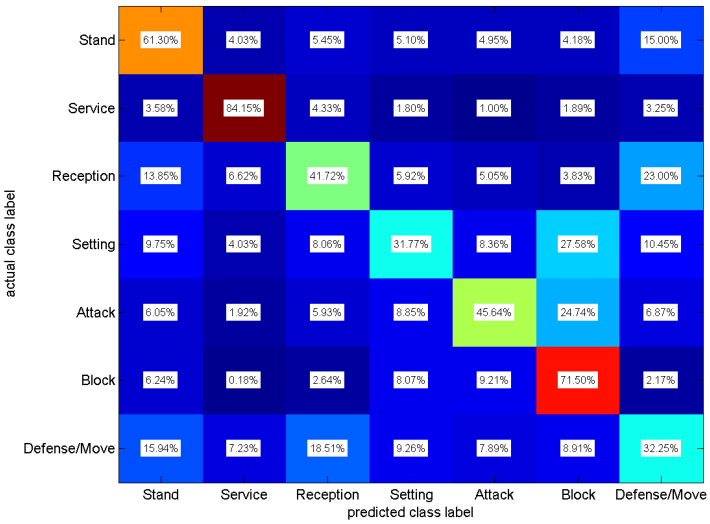


Figure 6.30: *Average over all runs with HOF-RWPC descriptor set: Unsurprisingly this combination works good for classes the single descriptors perform good on. RWPC works good for "Service" and "Block" and HOF is best on "Stand" such that these classes prevail. For the rest the performance is only mediocre.*

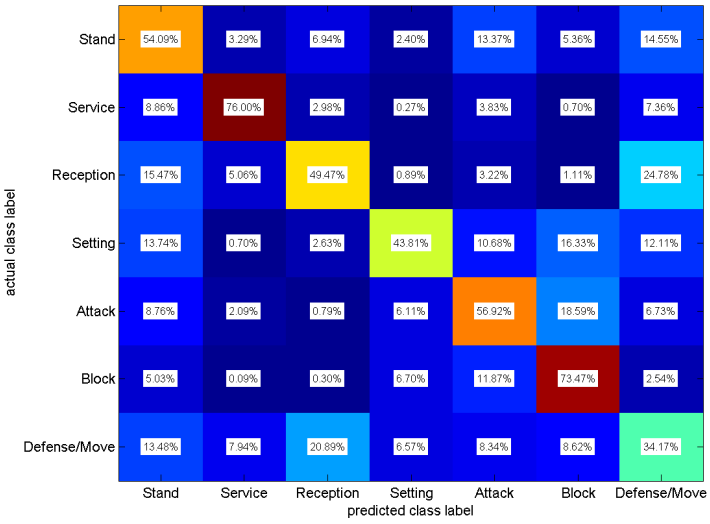


Figure 6.31: *Average over all runs with HoG-RWPC descriptor set: Like before, the combination of the single descriptor results is evident. Classes where both perform bad are not improved with exception of the "Defense/Move" class.*

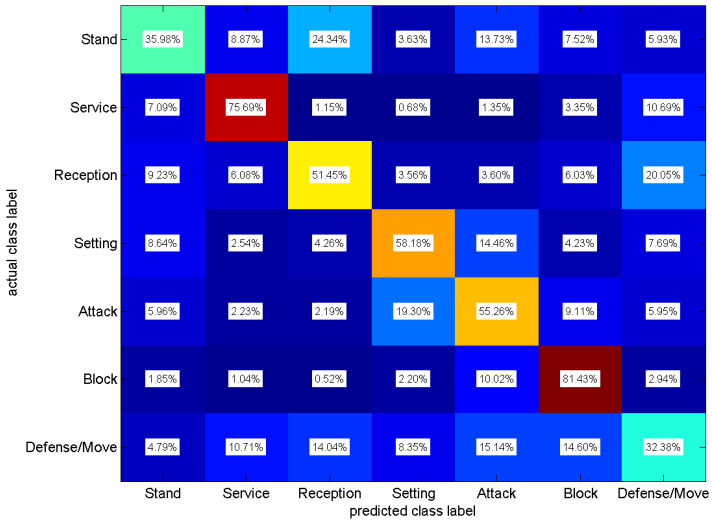


Figure 6.32: *Average over all runs with SC-RWPC descriptor set: This combination of spatial information descriptors is obviously weak for activities without position specific background like "Stand" or "Move/Defense". As with the single descriptors the two classes "Service" and "Block" are classified best.*

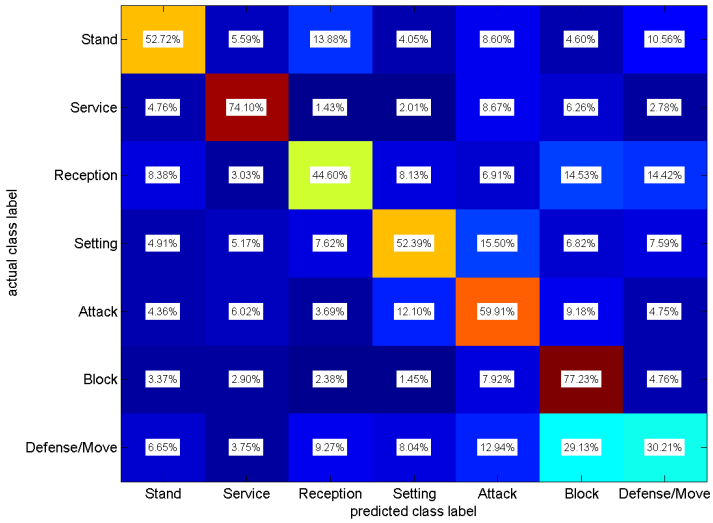


Figure 6.33: *Average over all runs with HOF-SC descriptor set: As the SC descriptor cannot help with activities that have divergent on court player distributions like "Defense/Move" and "Reception", the HOF results are bettered for "Block" (+30%), "Attack" (+30%), "Setting" (+25%) and "Service" (+25%). "Stand" is negatively influenced by the SC descriptor (-10%).*

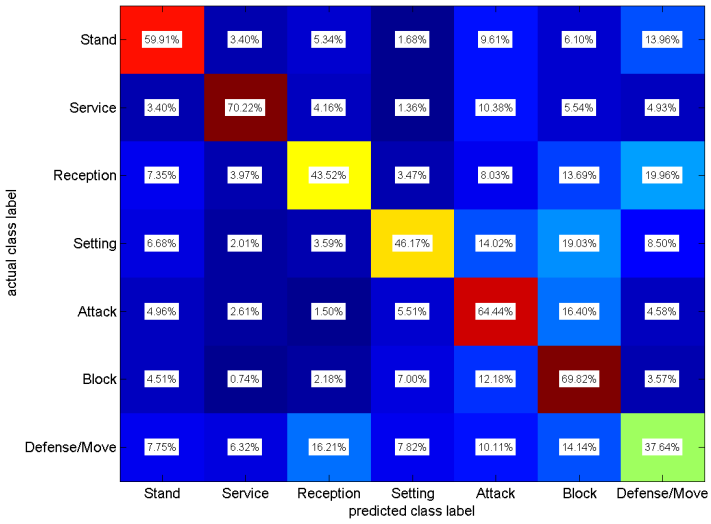


Figure 6.34: *Average over all runs with HOF-HOG descriptor set: This popular computer vision combination leads to only a slight improvement over all classes compared to the single descriptors. Obviously the use of spatial information is needed for the recognition task examined within this thesis.*

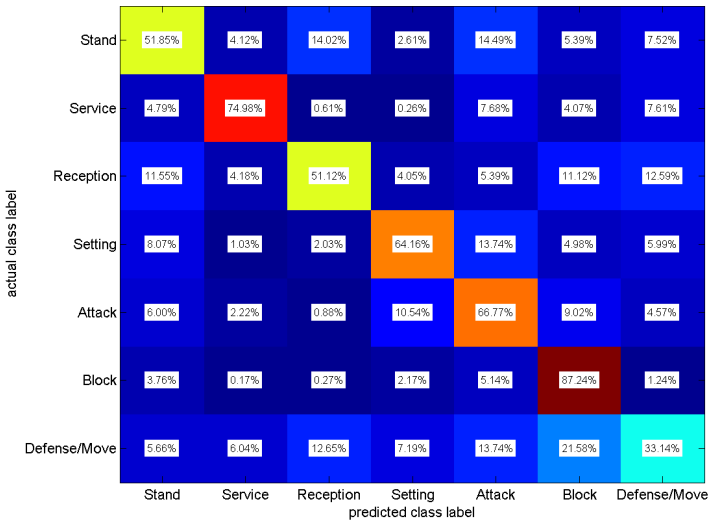


Figure 6.35: *Average over all runs with HOG-SC descriptor set: This combination improves all class results (2%-18%), only worsening the result of "Stand" due to SC influence (like with HOF-SC).*

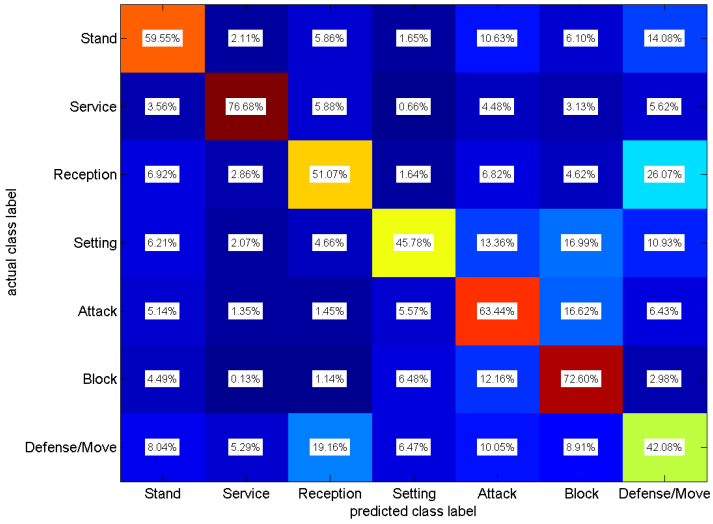


Figure 6.36: *Average over all runs with HOF-HOG-RWPC descriptor set: While many classes are rather good discriminated by this three descriptor set, "Reception" is again often confused with "Defense/Move" (19% and 26%). Situation is similar for the classes "Attack" and "Block" with a confusion of 12% and 16% respectively.*

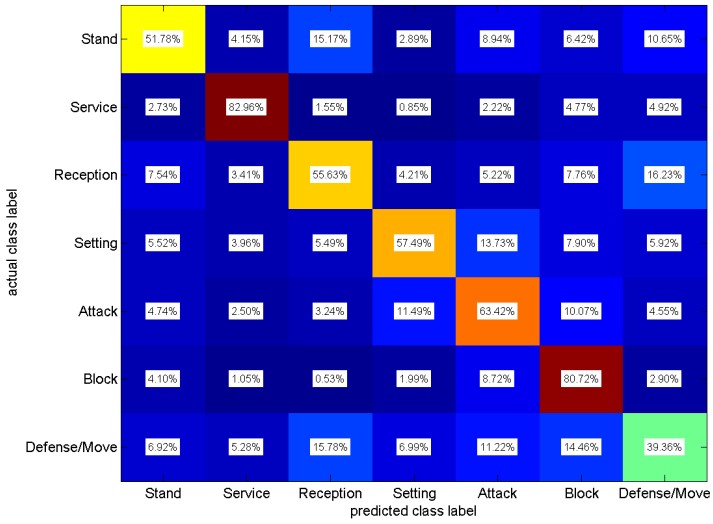


Figure 6.37: *Average over all runs with HOF-SC-RWPC descriptor set: "Service" and "Block" are best classified, followed by "Attack", "Setting", "Reception" and "Stand". "Defense/Move" remains the hardest class for most descriptor sets.*

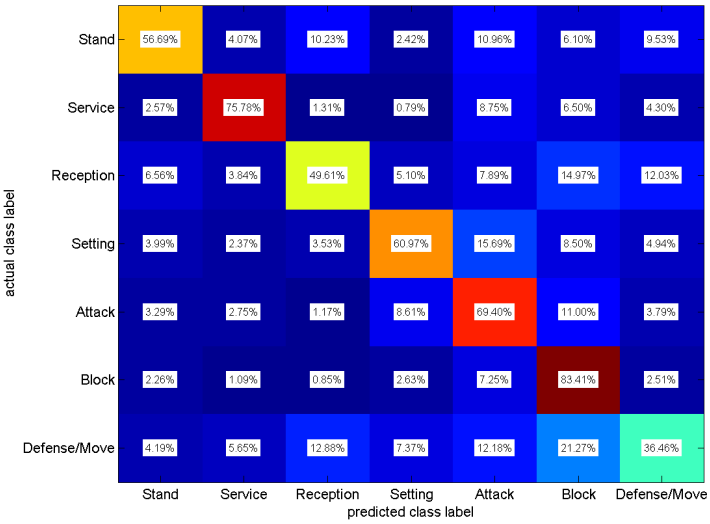


Figure 6.38: *Average over all runs with HOF-HOG-SC descriptor set: Like before, the classes "Block", "Service" and "Attack" are classified best, followed by "Setting" and "Stand". It seems, that "Reception" needs the RWPC descriptor for better results. As usual "Defense/Move" is the hardest class.*

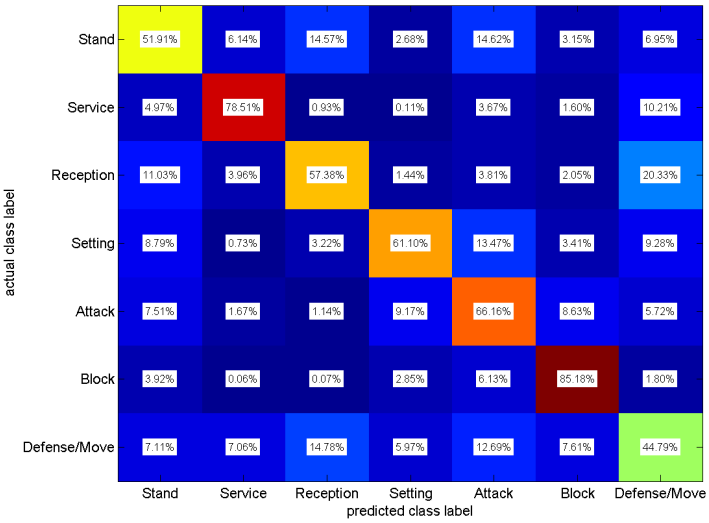


Figure 6.39: *Average over all runs with HOG-SC-RWPC descriptor set: Opposite to the previous combination, "Reception" is classified better while the top three ("Block", "Service", "Attack") and "Setting" remain the same. Classification of "Stand" is worse than with the HOF descriptor since the typical lack motion is not utilized with this combination.*

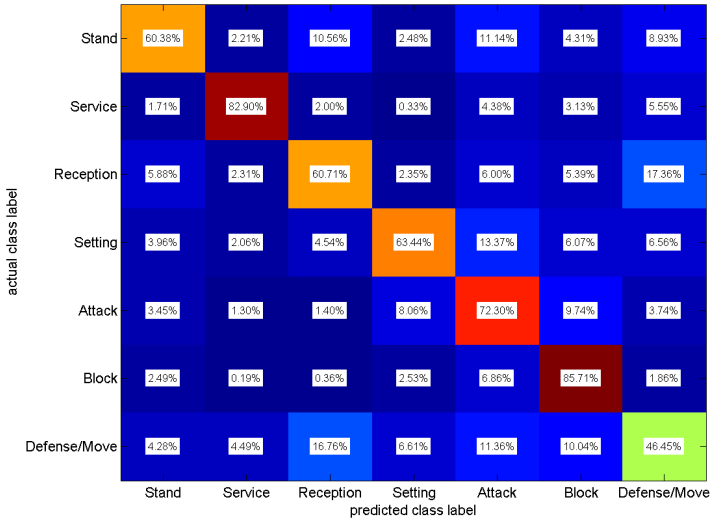


Figure 6.40: *Average over all runs with HOF-HOG-SC-RWPC descriptor set: With the highest average result and at least similar results for single classes compared with other descriptor sets the combination of all four descriptors again yields the best overall performance. Except for the "Defense/Move" class, all activities are classified with notable performance.*

6.2.7 Discussion of Spatial Context Player Activity Recognition Results

The results of the single frame activity recognition have proven the context information to be very valuable. Especially the HOF descriptor alone, evaluated with multiple different parameter sets, is not capable of classifying the activities. The best descriptor (when used alone) is the HOG descriptor with a 63.46% recognition rate, followed by HOF (55.07%), SC (53.77%) and RWPC (49.82%). For combination of two descriptors the results range between 71.83% (HOG-SC) and 65.59% (HOG-RWPC), such improving in every case over the best one descriptor result. Using three descriptors gives 75.05% (HOF-HOG-SC) to 72.13% (HOF-HOG-RWPC) raising the performance for another 3.22% and the use of all descriptors together gives the best result of 77.56%. This proves that all four descriptors add exploitable information for the recognition task and using more descriptors correlates to better description of the activities.
The choice of the SVM kernel is crucial for the classification result. A linear kernel is obviously not able to separate the activities, while the rbf and polynomial kernels yield the best results. Worst results are generated by the sigmoidal kernel, which is alternating for different descriptor sets but in any case is far from top performance of the rbf and polynomial kernels.
The best result of 77.56% shows that six of the seven classes can be differentiated quite good with results of 73.27% to 92.96%. Only the class "Move/Defense" has a relatively low performance of 52.63%, as this is a very inhomogeneous class and defending postures and movements look like activities from the class "Reception".

6.3 Activity Context Player Activity Recognition

The activity context player activity recognition introduces the time component into the description of the activities. Like the SC descriptor that uses information about locations of other players, the AC descriptor includes information about all player activities and positions over time.
This can be seen as a higher level recognition task, as it uses the previous trained SVM as underlying

base for estimating the persons activities on court. From the seven annotated and learnt activities, two are no specific volleyball situations and such are not target of this recognition task as they can happen along with any of the other classes: ”Stand” and ”Move/Defense”. These two activities can occur at any time of play and do not correlate with neither activity happening within a certain time period, nor with position (distributions) on the court. The other five express state of play situations in volleyball, executed by the team as whole. These five activities are the ones that should be improved within the activity context player activity recognition. An overview about the activities is presented in Section 4.3.

6.3.1 Verification of Player Localization

The AC descriptor is built on examination of all players in an activity scene. Therefore the player positions need to be known. As during the annotation mostly only one player was manually marked per video frame, the other players - or more specifically their on court positions - needed to be found. This was realized by using the segmented foreground areas found in Section 3.1 to get approximate positions of the non-annotated players.
To test the correctness of the player localization, a number of correct classified frames from the spatial context activity recognition was taken and examined with offsets. The offsets were calculated on the court plane, and mapped back into the image plane. Up to a certain offset the performance is very good, for an offset of 15cm left/right, 15cm back and 30cm forward, the accuracy remains at 93.25% on average. This is approximately the range of the foreground area projections on the court. For the total investigated offset of 60cm left, right and forward and an offset of 15cm backward, the average accuracy stayed at 79.28%. Figure 6.41 shows the detailed results along with some examples.

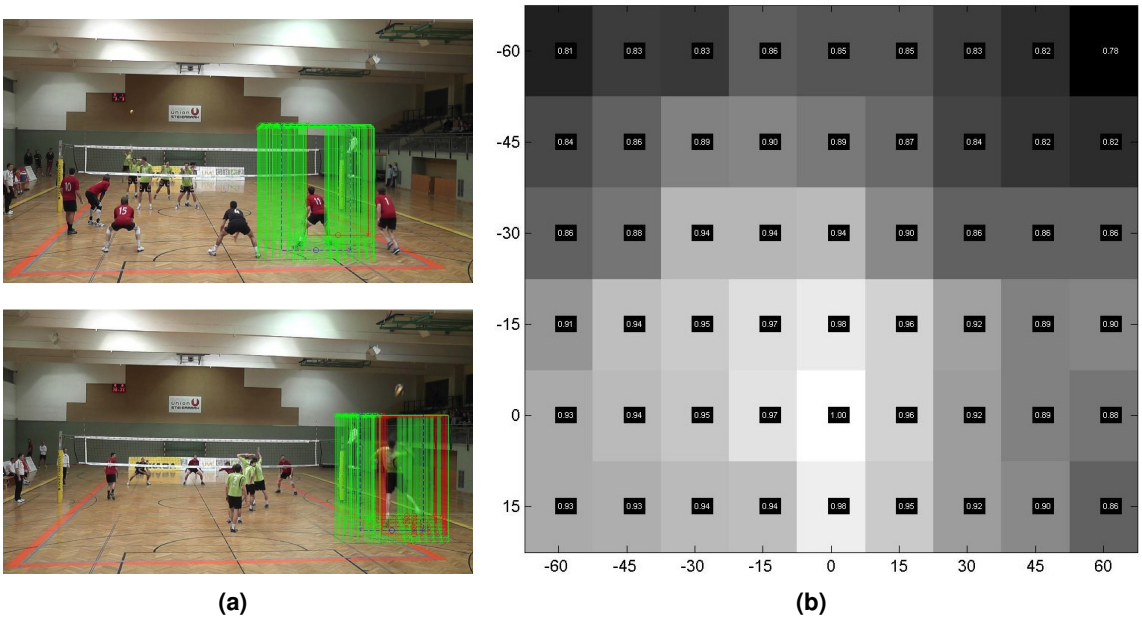


Figure 6.41: *Accuracy of the classification in dependence of projected location. 200 tested frames where chosen randomly from the correct classified activities and put into the SVM with different offsets. (a) Example for offset results (top: reception, bottom: service). The blue dashed line denotes the original annotation, green boxes are correct and red false classified offset samples. (b) Overall results (top view), 100% at the origin and slowly decreasing with added offset. Offset in direction of the net has more impact on the results, as the scale changes opposed to vertical offset.*

6.3.2 Results for Activity Context Player Activity Recognition

The results obtained with the AC descriptor vary for each of the classes. For some classes, few investigated frames improve performance while many decrease performance. Other classes need information about many frames to be collected so that the classification result reaches its optimum. The same is valid for the number of votes parameter (p). Including all 7 classes introduces noise, using only 1 class is not sufficient as possible correct classes might be suppressed. The best parameter for p was found with 3 classes included for description. The best parameter for τ was 40, meaning 1.6 seconds of video occurring before the activity were incorporated forming the descriptor.
Tables 6.7 and 6.8 show the best results achieved with use of the AC descriptor with variable binning (10×10 and 15×15). Obviously for the two no-specific classes, the results get worse as the descriptor can not support classification of these classes because of their random appearance during the game. For "Stand" this means a decline of -20.43% (10×10)/-27.07% (15×15) and -15.97%/-22.16% for "Defense/Move". For the rest of the activities more or less improvement is achieved so that for the five main volleyball classes an average result of 90.19%/87.57% is accomplished, equaling 7.20%/4.59% average recognition enhancement compared to the results without AC descriptor. With 18.35%/12.57% the most profit from the AC descriptor is produced for the "Reception" activity, while "Attack" has the least benefit with 1.15%/-2.90%. The other three activities "Service", "Setting" and "Block" improve by around 5%.
The last three columns of the table show the best performance of different activities over all runs with varying parameter sets. The second last column points out how different the parameters for optimal results are. Three classes perform best with many frames considered for the AC descriptor (parameter $\mathbf{k}$), while the others are best differentiated by investigating half or less frames. Also the second parameter $\mathbf{p}$ varies within the class-wise optimal results.

class	accuracy before[1]	accuracy with best parameter set (τ=40, p=3)[2]	difference	best accuracy over all parameter sets	best parameter set τ/p	difference
Stand	75.37%	54.94%	-20.43%	66.58%	200/3	-8.79%
Service	91.01%	97.13%	6.12%	97.93%	40/1	6.92%
Reception	73.27%	91.62%	18.35%	91.68%	200/7	18.41%
Setting	74.95%	80.69%	5.74%	82.29%	200/7	7.34%
Attack	82.74%	83.89%	1.15%	85.83%	100/7	3.09%
Block	92.96%	97.60%	4.64%	99.16%	70/1	6.20%
Def/Move	52.63%	36.39%	-16.24%	45.05%	100/3	-7.58%
average (all classes)	77.56%	77.47%	-0.09%	81.22%	-	3.66%
average (5 classes)	82.99%	**90.19%**	**7.20%**	91.38%	-	8.39%

Table 6.7: *Results from spatial context activity recognition[1] compared with activity context player activity recognition[2] and* 10×10 *binning: Best result is achieved with τ=40, p=3. For the non-specific classes the performance decreases strongly while the five specific classes improve by up to 18.35%.*

The following Figures 6.42 to 6.47 show results of different parameter sets for the activity descriptor. Parameters on the x axis denote number of winners (p=1,3 or 7) and the number of frames used (τ=10-200). The dashed lines denote the average recognition results before using the AC descriptor for five (black, 82.99%) and seven (orange, 77.56%) classes respectively. The solid orange line shows results for all seven classes with use of AC descriptor. The solid black line depicts the average over the five volleyball specific activity classes with AC context and has results of the configurations displayed. The other colored lines (purple="Stand", pink="Service", cyan="Reception", blue="Setting", dark green="Attack", brown="Block", light green="Defense/Move") denote results with AC descriptor dependent on parameters and have the top results for each class indicated. This coloring makes it easy to compare which classes profit from the AC descriptor and which don't. With about 90% on the

class	accuracy before[1]	accuracy with best parameter set (τ=60, p=3)[2]	difference	best accuracy over all parameter sets	best parameter set τ/p	difference
Stand	75.37%	48.30%	-27.07%	70.31%	200/1	-5.06%
Service	91.01%	96.76%	5.75%	96.76%	20/7	5.75%
Reception	73.27%	85.84%	12.57%	89.73%	200/7	16.46%
Setting	74.95%	79.15%	4.20%	80.90%	70/1	5.95%
Attack	82.74%	79.84%	-2.90%	86.43%	150/7	3.69%
Block	92.96%	96.27%	3.31%	98.05%	150/3	5.09%
Def/Move	52.63%	30.20%	-22.43%	45.34%	10/7	-7.29%
average (all classes)	77.56%	73.76%	-3.80%	81.08%	-	3.51%
average (5 classes)	82.99%	**87.57%**	**4.59%**	90.37%	-	7.39%

Table 6.8: *Results from spatial context activity recognition[1] compared with activity context player activity recognition[2] and* 15×15 *binning: Best result is achieved with* τ*=60,* p*=3. For the non-specific classes the performance decreases strongly while the five specific classes improve by up to 12.57%.*

five specific classes the results for 10×10 are approximately 3% better than for 15×15 with around 87%.

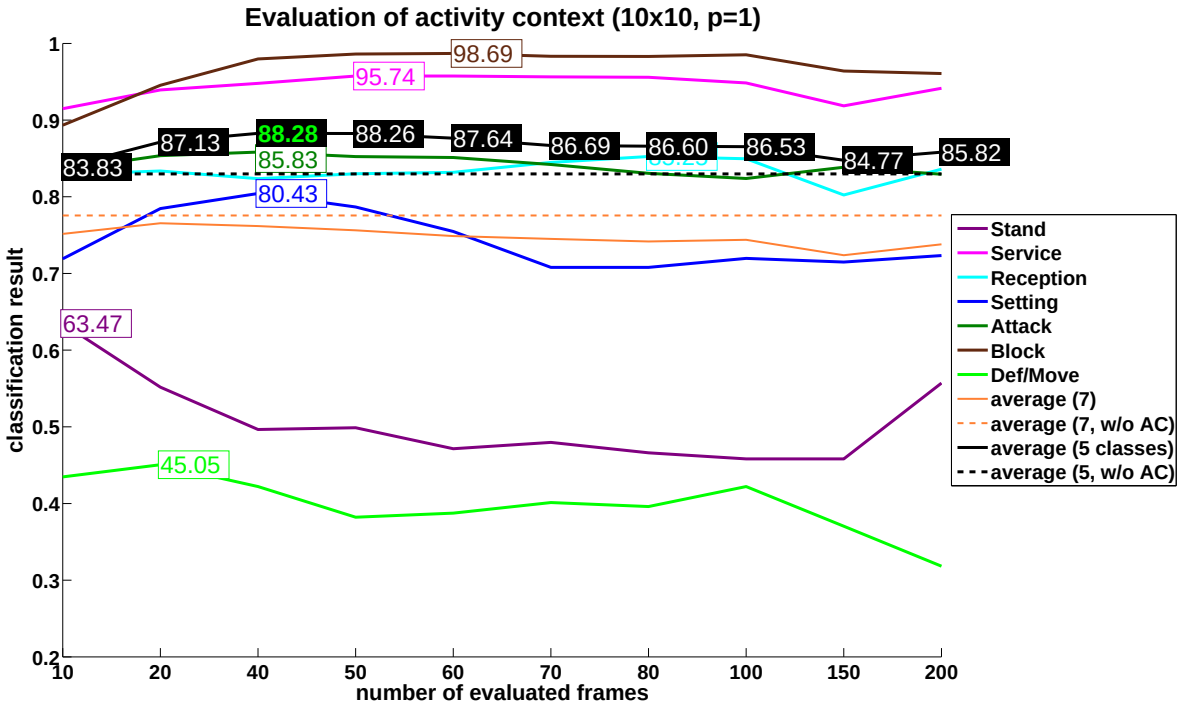


Figure 6.42: *Results for activity recognition with AC (bin size 10x10 and p=1): Best result is achieved with* $\tau = 40$ *frames and 88.28% performance. All five important classes are better than before, the two general classes are always worse than without AC.*

6.3.3 Discussion of Activity Context Player Activity Recognition Results

Use of the AC descriptor improves the overall recognition rate on the volleyball specific activities by over 7% on average compared to exclusive use of spatial information, showing that the context supports discrimination of complex activities involving multiple persons (players). It seems that using

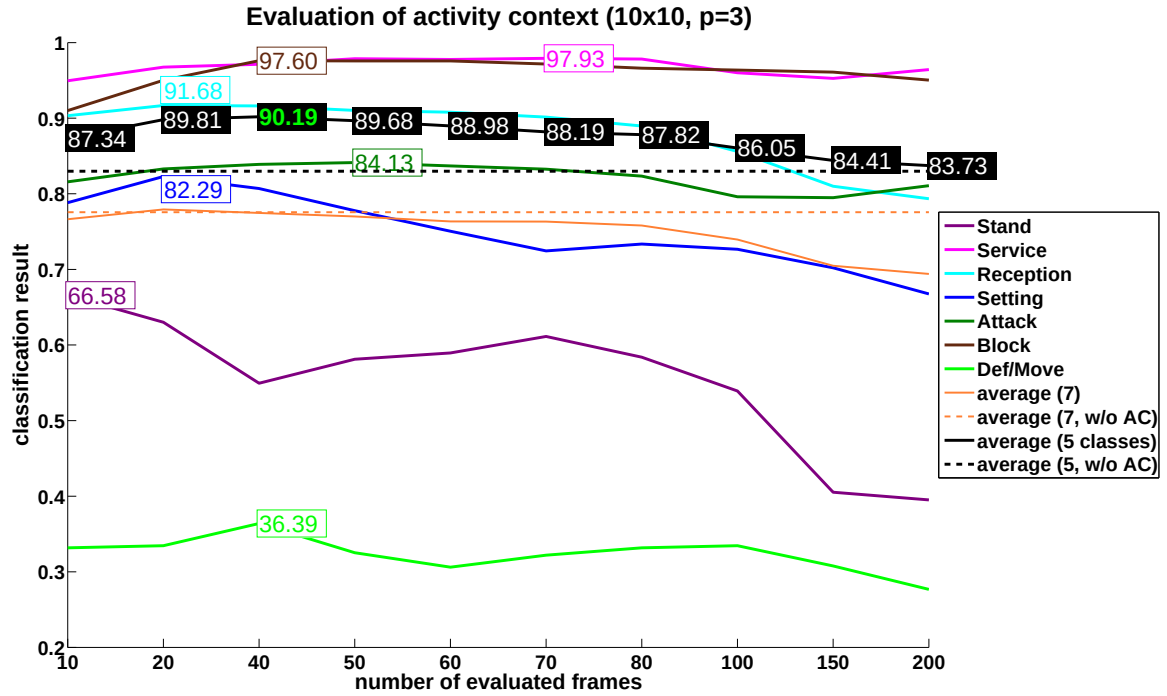


Figure 6.43: *Results for activity recognition with AC (bin size 10x10 and p=3): With 90.19% best result and also overall best result of all AC parameter configurations is achieved with* $\tau = 40$*. Three classes ("Service", "Block" and "Reception") are above 90%, the other performances are with 80.69% ("Setting") and 83.89% ("Attack") also very high.*

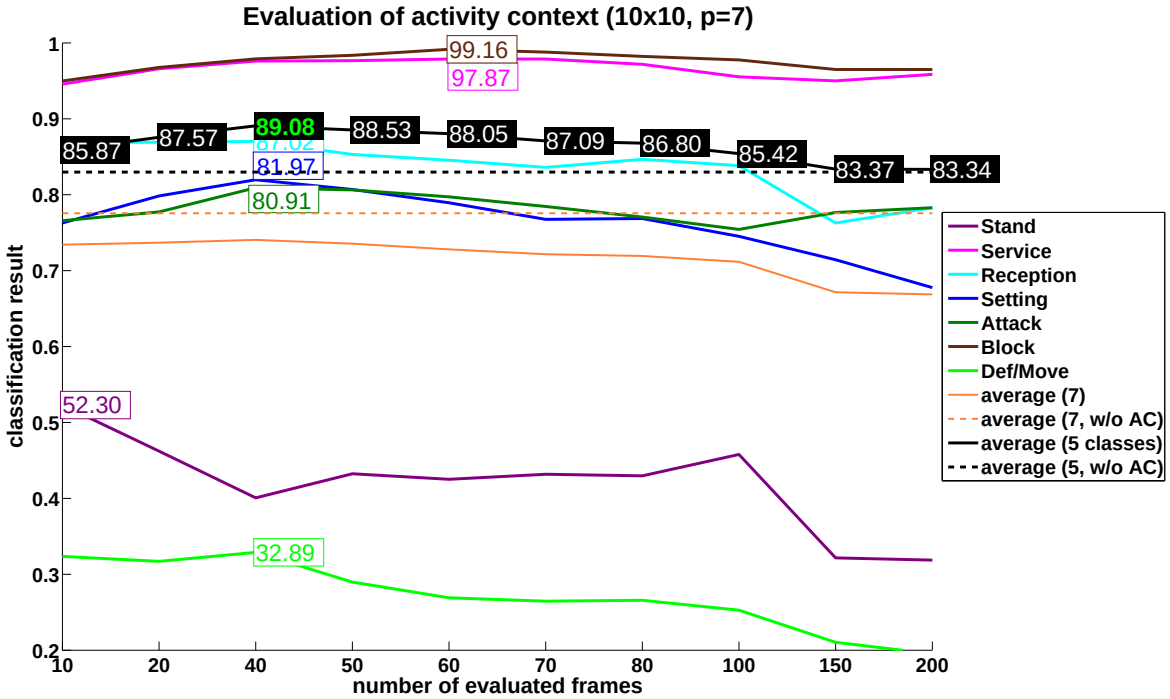


Figure 6.44: *Results for activity recognition with AC (bin size 10x10 and p=7): Again* $\tau = 40$ *yields best results. The average performance is around 2% worse than in the previous optimal case but still an improvement.*

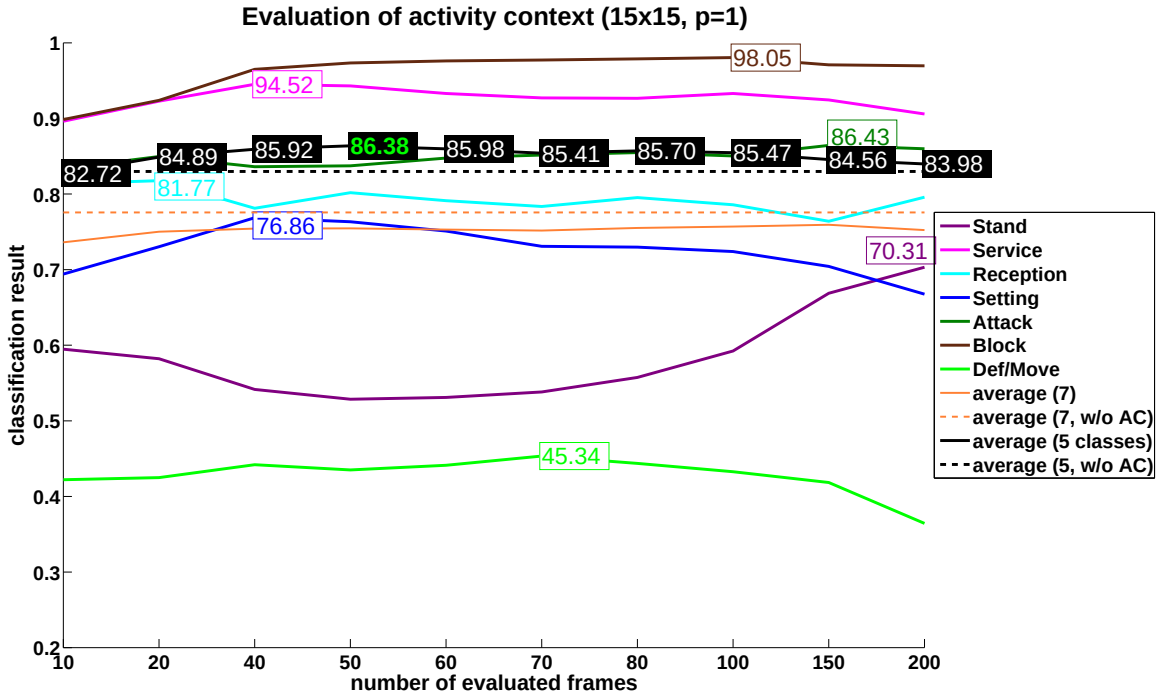


Figure 6.45: *Results for activity recognition with AC (bin size 15x15 and p=1): The results for this binning are best with* $\tau = 50$*. Again, except for "Stand" and "Defense/Move" all classes improve.*

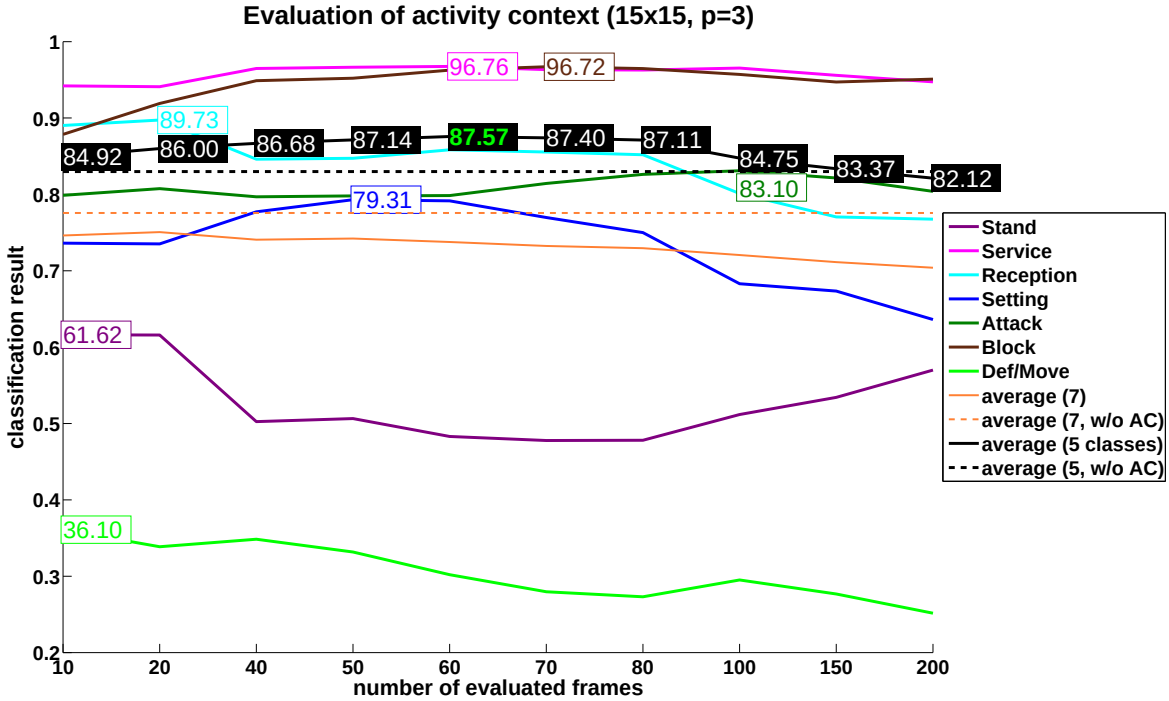


Figure 6.46: *Results for activity recognition with AC (bin size 15x15 and p=3): Like for* 10×10*, the best results are achieved with* $p = 3$*. In this case* $\tau = 60$ *gives best results for the five interesting classes.*

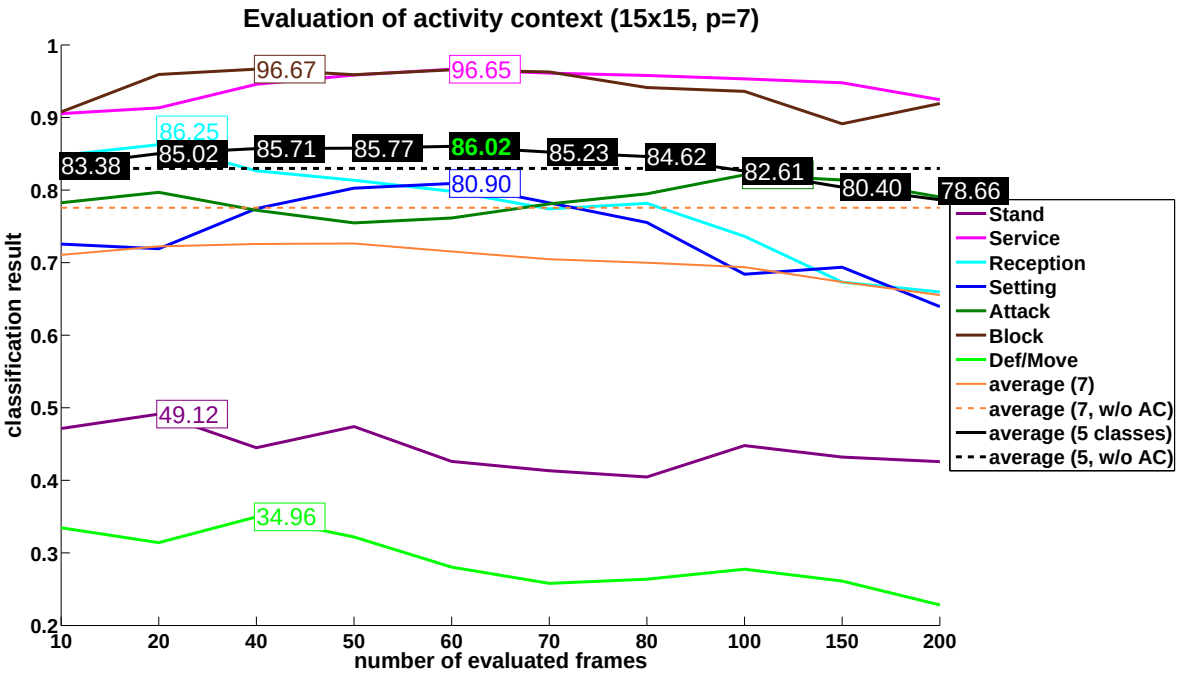


Figure 6.47: *Results for activity recognition with AC (bin size 15x15 and p=7): The results are best with* $\tau = 60$ *again, yet a performance of 86.02% is the worst of the tested configurations.*

40-60 frames before the actual evaluated frame is best for recognition. This is plausible, as the previous activity usually takes place within this timespan of 1.5-2.5 seconds. Looking back further in time by increasing the numbers of frames investigated, for activities like ”Service” or ”Setting” no further activities occur but only unclassified player movements. This is due to the fact that for example ”Service” marks the begin of a new rally. The general classes ”Stand” and ”Move/Defense” deteriorate more and more as n is increased. This is conclusive, as those classes do not occur within the rally at specific times or orderings but almost randomly between or during rallys. So the incorporation of context about other activities is not helpful. For the other five activities, with over 90% correct classified samples the motivation of using context to improve the recognition results is confirmed.

6.4 Overall Discussion

This section detailed all evaluated experiments and used parameters for descriptors and classifiers. The introductory section 6.1 gives an overview on the possibilities for parameter selection. With an optimum of $\mathbf{78.74}\%$ the results for spatial context player activity recognition (Section 6.2) are good, regarding that the recognition is done frame wise and only spatial context is employed. The idea of adding information about the other player activities is tested in the result section of activity context player activity recognition (Section 6.3), where the automatically localized surrounding players are evaluated with respect to their activities over time. While the average for all classes stays the same, the recognition rate shifts away from the neutral classes ”Stand” and ”Move/Defense” towards the other five discriminative classes. For these five volleyball specific classes the result is improved by $\mathbf{7.20}\%$ to $\mathbf{90.19}\%$. This proves that the information provided by the other players strongly supports the classification of the investigated player activity.

Chapter 7

Outlook

Although this thesis presents extensive work on activity recognition in sport with focus on Volleyball, there are always ideas that could be applied for alternate or enhanced solutions to the posed problem or improvement of the presented results. The following points are suggestions for possible future work on the task of sport activity recognition in volleyball.

7.1 Finer Division of Classes (Action- and Location-Wise)

A further division into action (sub activity) classes like jump, land, move right/left/back/forward, bend, squat and subsequent recombination into the activity classes defined in this thesis would facilitate the classification job. Clearly this would mean much more effort - especially in the annotation process which was already a very time consuming step.
A finer declaration of classes position-wise could help. For example introduction of three service positions (left, middle, right) or several attack positions (left, middle, right plus same for backrow attacks) would raise the impact of the position descriptors. At the moment, attack, block and setting classes are somewhere in front, service somewhere in the back and move/defense is spread across the field. Thus more detailed spatial information could help to improve determination of the other players activities.

7.2 Merging Information about Second Team

Combination of own activities with the other teams activities could lead to more information about the occurring overall game activity. If one team is in attack formation, it is very likely that the other team is in block/defense formation. Same for service and receive.

7.3 Information about Referees

Involving the referees would help prediction of the future and past activities. For example, the referee must allow the service player to start the rally. After the end of the rally the referee decides which team wins the point and supports his decision with explicit gestures.

7.4 Adaptive Classification

The presented results showed, that the performance on a class depends on the parameter set of the descriptors and the SVM. A weighted classification scheme with adapted parameters for each class might further improve the recognition performance.

7.5 Tracking/3D Information

Tracking needs to be solved. The foreground segmentation results are promising, but for example a multi-camera setup would lead to 3D information (depth, distance). This information could also be used for the activities - a block activity can only occur at the net and such could no more be confused with service activities from behind the court. Also, blocking activities do not involve movement to the net but only parallel to the net, while service and attack activities involve movement in orthogonal direction. The distance from net would discriminate service against attack activities.

7.6 Ball Detection and Tracking

Within this thesis the presence of the ball was ignored. Adding the position of the ball as context information should further improve results. Other features like direction or velocity should give clues about preceding and future activities and game states.

7.7 Automated Calibration and Preprocessing

To reach the goal of an autonomous software, as much work as possible should be undertaken by the computer system. For example, a initialization phase could be started to learn the background model and afterwards the player models. Having learnt the background model, the estimation of the court boundaries, marked by lines, could be started as the background also includes the lines on the floor. This would also allow for some change in camera position.

7.8 Additional Spatial-Temporal Features

Within this work, only positions on the field are used as spatial features, once solely within one frame, once over multiple frames. One could extend the players position information to give information about velocities, directions and also measure the distances to other players while executing one distinct activity. As use of the four presented descriptors together yields best performance for recognizing the activities, adding more descriptors and such information could further improve the results.

7.9 Inclusion of Audio Information

In many ball sports the referees make calls supported by whistles. The average used camera for recording sport games is capable of recording a synchronized audio stream. This information could be easily extracted and would give strong clues about beginning and ending of rallies and such about the possible activities that follow a whistle. In volleyball this could be a service activity or a timeout activity. Furthermore, the cheering of the crowd could be incorporated, as normally during the rallies

the crowd is silent and cheers more should the home team scores than in case of a point for the opponent.

7.10 Parameter Optimization

Although the parameters have been tested in various variants, there might be still place for improvement. Especially the not tested χ^2-kernel or other kernel variants could improve performance. Due to the number of possible combinations not all parameter sets could be evaluated and had to be limited in some way.

Chapter 8

Concluding Remarks

This section gives a short summary about the ideas presented and implemented within this thesis, a presentation of my work in the field of activity recognition for sport analysis. After a introductory Section 1, a summary of related work has been presented in Section 2. Research on activity recognition has been conducted since the beginning of computer vision science and is still a partly solved topic needed for various applications. Activity recognition in sports is a topic of interest for multiple parties like athletes, coaches, referees or TV broadcasters. Within the big variety of sports, all of them underlie certain game rules and these specify a scheme of play that should be exploited for recognition. In every team sport, an insulated view on single players is not sufficient, as the team is performing as whole. Every team sport has certain formations to maximize winning probability, mostly categorized into defense (trying to prevent opponents success) and attack (trying to score). This motivated the use of contextual information for recognition of player activities.
For the evaluation of the methods presented within this thesis, an adequate dataset was required. As sport datasets are rare, much effort was put into generating a Volleyball dataset with sufficient data for a serious evaluation of activity recognition methods. For the annotation of the available six videos, a simple Matlab framework (Section 5.2) has been designed and three users (of these one volleyball expert) have made 8k manual annotations that were interpolated resulting in a dataset of 36k annotations in seven activity classes. Of these seven classes, two are very general motion classes (”Stand”, ”Defense/Move”) and five are volleyball specific activities (”Service”, ”Reception”, ”Setting”, ”Attack”, ”Block”). Section 5 gives a short information over the international volleyball rules for a better understanding of the evaluated data, which is summarized in Section 5.4.
For a consistent interpretation of the data, the videos were first calibrated to a common court plane via a planar homography calculation. This procedure, as well as the generation of foreground and background color models is outlined in Section 3, along with two prominent descriptors that have been proven to be working well for activity recognition (HOG, HOF). Section 4 then proposes three purposely designed descriptors. Two of them exploit spatial information: Position of the players on the court (RWPC) as well as spatial context information in form of player distribution probabilities (SC). The third introduces the behavior of players as activity context information over time (AC).
The structure of the recognition system consists of three steps: Preprocessing, spatial context player activity recognition and activity context player activity recognition. The first step is needed to prepare the data for examination. In the second step, a SVM classifier is trained on features from four descriptors (HOG, HOF, RWCP, SC) to generate predictions about executed activities. This classification is based on shape, motion and spatial information. In the third step, first the foreground and background models are used to find all players on and around the court. Then these players are classified in terms of their activity using the SVM trained in step two. This is repeated over multiple frames prior to the evaluated one, such generating temporal activity context information. Supported with this activity

context, a new SVM is trained including the features from the previous step. Both SVMs were trained with a 50% training and 50% testing split. All parameter configurations were extensively tested (see Section 6).

The results of the conducted experiments are promising. Compared to a classification result of $\mathbf{69.47}\%$ for sole description of motion and shape through the HOG/HOF descriptor combination, for spatial context player activity recognition an average rate of $\mathbf{77.56}\%$ correct classified samples could be achieved. For the activity context player activity recognition task, due to the bad results on the general classes "Stand" and "Move/Defense" the average over all seven classes remains the same. For the five volleyball specific classes however, the results could be improved by $\mathbf{7.20}\%$ on average and up to $\mathbf{18.35}\%$ for "Reception", resulting in an classification accuracy of $\mathbf{90.19}\%$ on these classes. This shows, that the exploitation of spatial and temporal context improves the recognition rate significantly. Clearly the availability of information about the sport scene is very important. Although only one single camera in the rear of the field was available to supply data, the results are promising and prove the success of the presented methods. The final section 7 intends to illustrate some thoughts about further research, that could be carried out in future within the context of sport activity recognition on volleyball.

The proposed activity recognition system should be extendable to other activity recognition tasks like surveillance or medical homecare where the observed persons interact with surrounding persons or objects making the exploitation of spatio-temporal context reasonable. For (team) sports, only slight adaptions should be needed to evaluate the presented methods. Although the player executions in game differ from sport to sport, every sport has reoccurring movement patterns and interactions with other players or objects (ball, racket, bat,...).

Bibliography

Atmosukarto, I., Ghanem, B., Ahuja, S., Muthuswamy, K. and Ahuja, N. (2013). *Automatic Recognition of Offensive Team Formation in American Football Plays*. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pages 991–998. IEEE. 12

Beauchemin, S. S. and Barron, J. L. (1995). *The computation of optical flow*. ACM Comput. Surv., 27(3):433–466. 29

Belongie, S., Malik, J. and Puzicha, J. (2002). *Shape matching and object recognition using shape contexts*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24(4):509–522. 11, 13

Bialkowski, A., Lucey, P., Carr, P., Denman, S., Matthews, I. and Sridharan, S. (2013). *Recognising Team Activities from Noisy Data*. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on, pages 984–990. 12, 14

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford University Press, Oxford, UK. 22

Boghossian, B. A. and Velastin, S. A. (1999). *Image processing system for pedestrian monitoring using neural classification of normal motion patterns*. Measurement and Control (Special Issue on Intelligent Vision Systems), 32:261–264. 9

Boghossian, B. A. and Velastin, S. A. (2002). *Motion-based machine vision techniques for the management of large crowds*. Electronics, Circuits and Systems, 1999. Proceedings of ICECS '99. The 6th IEEE International Conference on, 2:961–964 vol.2. 9

Bregonzio, M., Gong, S. and Xiang, T. (2009). *Recognising action as clouds of space-time interest points*. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 1948–1955. IEEE. 11

Brendel, W. and Todorovic, S. (2011). *Learning spatiotemporal graphs of human activities*. In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 778–785. IEEE. 12

Bruhn, A., Weickert, J. and Schnörr, C. (2005). *Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Methods*. International Journal of Computer Vision, 61(3):211–231. 29

Burgos-Artizzu, X. P., Dollár, P., Lin, D., Anderson, D. J. and Perona, P. (2012). *Social behavior recognition in continuous video*. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 1322–1329. 13

Center for Biometrics and Security Research (2007). Casia action database for recognition. http://www.cbsr.ia.ac.cn/english/ActionDatabasesEN.asp. Accessed: 2014-02-12. 14

Chaquet, J. M., Carmona, E. J. and FernÃ¡ndez-Caballero, A. (2013). *A survey of video datasets for human action and activity recognition*. Computer Vision and Image Understanding, 117(6): 633–659. 14

Chen, H., Chou, C., Tsai, W., Lee, S. and Yu, J. (2011). *Extraction and representation of human body for pitching style recognition in broadcast baseball video*. In Multimedia and Expo (ICME), 2011 IEEE International Conference on, pages 1–4. 10

Christian Schuldt, I. L. and Caputo, B. (2002). Caviar: Context aware vision using image-based active recognition. http://www.nada.kth.se/cvap/actions/. Accessed: 2014-02-12. 14

Collins, R., Lipton, A. and Kanade, T. (1999). *A System for Video Surveillance and Monitoring*. In American Nuclear Society 8th Internal Topical Meeting on Robotics and Remote Systems. 9

Collins, R. T., Lipton, A. J. and Kanade, T. (2000). *Introduction to the Special Section on Video Surveillance*. IEEE Trans. Pattern Anal. Mach. Intell., 22(7):745–746. 9

Cunado, D., Nixon, M. S. and Carter, J. N. (1999). *Automatic Gait Recognition via Model-Based Evidence Gathering*. In O'Gorman, L. and Shellhammer, S., editors, Proceedings AutoID '99: IEEE Workshop on Identification Advanced Technologies, pages 27–30. IEEE. 9

Dalal, N. and Triggs, B. (2005). *Histograms of oriented gradients for human detection*. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, page 886. 11, 27

Dalal, N., Triggs, B. and Schmid, C. (2006). *Human detection using oriented histograms of flow and appearance*. Computer Vision-ECCV 2006, pages 428–441. 11

Dalal, N. (2006). *Finding people in images and videos*. PhD thesis, Institut National Polytechnique de Grenoble. v, 28, 30

Davis, J. W. and Bobick, A. F. (1997). *The Representation and Recognition of Human Movement Using Temporal Templates*. In Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), CVPR '97, pages 928–, Washington, DC, USA (1997). IEEE Computer Society. 11

Dollár, P., Rabaud, V., Cottrell, G. and Belongie, S. (2005). *Behavior recognition via sparse spatio-temporal features*. In Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on, pages 65–72. 11

Duda, R. O., Hart, P. E. and Stork, D. G. (2001). *Pattern Classification*. John Wiley & Sons, New York, NY, 2 edition. 23

EC Funded CAVIAR project/IST 2001 37540 (2002). Caviar: Context aware vision using image-based active recognition. http://homepages.inf.ed.ac.uk/rbf/CAVIAR/. Accessed: 2014-02-12. 14

Efros, A. A., Berg, A. C., Mori, G. and Malik, J. (2003). *Recognizing Action at a Distance*. In Computer Vision, IEEE International Conference on, volume 2, page 726, Los Alamitos, CA, USA (2003). IEEE Computer Society. 11

Ekin, A., Tekalp, A. and Mehrotra, R. (2003). *Automatic soccer video analysis and summarization*. IEEE Transactions on Image Processing, 12(7):796–807. 9

Fisher, R. (2001). Behave: Computer-assisted prescreening of video streams for unusual activities. http://groups.inf.ed.ac.uk/vision/BEHAVEDATA/INTERACTIONS/index.html. Accessed: 2014-02-12. 14

Forsyth, D. A., Arikan, O., Ikemoto, L., O'Brien, J. and Ramanan, D. (2005). *Computational Studies of Human Motion: Part 1, Tracking and Motion Synthesis*. Foundations and TrendsÂ® in Computer Graphics and Vision, 1(2/3):77–254. 11

Frati, V. and Prattichizzo, D. (2011). *Using Kinect for hand tracking and rendering in wearable haptics*. In World Haptics Conference (WHC), pages 317–321. 9

Freeman, W. T. and Weissman, C. D. (1995). *Television control by hand gestures*. In International Workshop on Automatic Face and Gesture Recognition, pages 179–183. 9

Freund, Y. and Schapire, R. E. (1995). *A desicion-theoretic generalization of on-line learning and an application to boosting*. In Computational learning theory, pages 23–37. Springer. 11

Gade, R. and Moeslund, T. B. (2013). *Sports Type Classification Using Signature Heatmaps*. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pages 999–1004. IEEE. 12

Ghanem, B., Kreidieh, M., Farra, M. and Zhang, T. (2012). *Context-aware learning for automatic sports highlight recognition*. In Pattern Recognition (ICPR), 2012 21st International Conference on, pages 1977–1980. IEEE. 14

Gorelick, L., Blank, M., Shechtman, E., Irani, M. and Basri, R. (2005). Weizmman actions as space-time shapes. http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html. Accessed: 2014-02-12. 14

Haritaoglu, I., Harwood, D. and Davis, L. S. (2000). *W4: Real-time surveillance of people and their activities*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22:809–830. 9

Harris, C. and Stephens, M. (1988). *A combined corner and edge detector*. In In Proc. of Fourth Alvey Vision Conference, pages 147–151. 11

Hashimoto, S. and Ozawa, S. (2006). *A system for automatic judgment of offsides in soccer games*. In International Conference on Multimedia and Expo, pages 1889–1892. 9

Hoiem, D., Efros, A. A. and Hebert, M. (2006). *Putting Objects in Perspective*. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2, CVPR '06, pages 2137–2144, Washington, DC, USA (2006). IEEE Computer Society. 13

Hollingworth, A. and Henderson, J. M. (1998). *Does consistent scene context facilitate object perception?* Journal of Experimental Psychology: General, Vol 127(4). 13

Horn, B. K. and Schunck, B. G. (1981). *Determining optical flow*. Artificial intelligence, 17(1): 185–203. 29

Huang, P., Harris, C. and Nixon, M. S. (1999). *Human Gait Recognition in Canonical Space Using Temporal Templates*. IEE Proceedings - Vision, Image and Signal Processing, 146(2):93–100. to be published. 9

Ikizler, N., Cinbis, R. G. and Duygulu, P. (2008). *Human action recognition with line and flow histograms*. In Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, pages 1–4. 11, 29

INRIA (2006). Inria xmas motion acquisition sequences (ixmas). http://4drepository.inrialpes.fr/public/viewgroup/6. Accessed: 2014-02-12. 14

Jain, M., Jégou, H. and Bouthemy, P. (2013). *Better exploiting motion for better action recognition*. In CVPR - International Conference on Computer Vision and Pattern Recognition, Portland, États-Unis (2013). 11

Kimeldorf, G. S. and Wahba, G. (1971). *Some Results on Tchebycheffian Spline Functions*. Journal of Mathematical Analysis and Applications, 33(1):82–95. 31

Kingston University (2010). Muhavi: Multicamera human action video data. http://dipersec.king.ac.uk/MuHAVi-MAS/. Accessed: 2014-02-12. 14

Kohle, M., Merkl, D. and Kastner, J. (1997). *Clinical gait analysis by neural networks: issues and experiences*. In Proceedings of the 10th IEEE Symposium on Computer-Based Medical Systems (CBMS '97), CBMS '97, Washington, DC, USA (1997). IEEE Computer Society. 9

Kolonias, I., Christmas, W. and Kittler, J. (2004). *Use of context in automatic annotation of sports videos*. In Progress in Pattern Recognition, Image Analysis and Applications, pages 1–12. Springer. 13

Lakany, H., Hayes, G., Hazlewood, M. and Hillman, S. (1999). *Human walking: tracking and analysis*. IEE Seminar Digests, 1999(103):5–5. 9

Lan, T., Wang, Y., Mori, G. and Robinovitch, S. N. (2012). *Retrieving actions in group contexts*. In Trends and Topics in Computer Vision, pages 181–194. Springer. 12

Laptev, I., Marszalek, M., Schmid, C. and Rozenfeld, B. (2008). *Learning realistic human actions from movies*. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8. 11, 29

Laptev, I. (2008). Hollywood Human Actions dataset. http://www.di.ens.fr/~laptev/download.html. Accessed: 2014-02-12. 14

Laptev, I. (2009). Hollywood-2 Human Actions and Scenes dataset. http://www.di.ens.fr/~laptev/download.html. Accessed: 2014-02-12. 14

Laptev, I. and Lindeberg, T. (2003). *Interest Point Detection and Scale Selection in Space-time*. In Proceedings of the 4th International Conference on Scale Space Methods in Computer Vision, Scale Space'03, pages 372–387, Berlin, Heidelberg (2003). Springer-Verlag. 11

Leo, M., D'Orazio, T., Mazzeo, P. and Distante, A. (2009a). *Multi-view Player Action Recognition in Soccer Games*. Proceedings of MIRAGE 2009, pages 46–57. 9

Leo, M., D'Orazio, T. and Trivedi, M. (2009b). *A multi camera system for soccer player performance evaluation*. In Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on, pages 1–8. 9

Liang, D., Liu, Y., Huang, Q. and Gao, W. (2005). *A scheme for ball detection and tracking in broadcast soccer video*. Advances in Multimedia Information Processing-PCM 2005, pages 864–875. 9

Little, J. J. and Boyd, J. E. (1996). Recognizing People by Their Gait: The Shape of Motion. 9

Lowe, D. G. (2004). *Distinctive Image Features from Scale-Invariant Keypoints*. Int. J. Comput. Vision, 60(2):91–110. 11, 27

Lu, W. and Little, J. (2006). *Simultaneous tracking and action recognition using the pca-hog descriptor*. In Computer and Robot Vision, 2006. The 3rd Canadian Conference on, page 6. 28

Lucey, P., Bialkowski, A., Carr, P., Foote, E. and Matthews, I. (2012). *Characterizing Multi-Agent Team Behavior from Partial Team Tracings: Evidence from the English Premier League.* In AAAI. 14

Marszalek, M., Laptev, I. and Schmid, C. (2009). *Actions in context*. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 2929–2936. IEEE. 13

Mauthner, T., Roth, P. and Bischof, H. (2010). *Temporal feature weighting for prototype-based action recognition*. Computer Vision-ACCV 2010, pages 566–579. 29

Mauthner, T., Koch, C., Tilp, M. and Bischof, H. (2008). *Visual Tracking of Athletes in Beach Volleyball Using a Single Camera*. International Journal of Computer Science in Sport, 6(2):21–34. 10

Maybank, S. and Tan, T. (2000). *Introduction*. Int. J. Comput. Vision, 37(2):173–173. 9

McQueen, A., Wiens, J. and Guttag, J. (2014). *Automatically Recognizing On-Ball Screens*. MIT Sloan Sports Analytics Conference. 14

Meyer, D. and Denzler, J. (1997). *Model Based Extraction of Articulated Objects in Image Sequences for Gait Analysis*. In Proceedings of the 1997 International Conference on Image Processing (ICIP '97) 3-Volume Set-Volume 3 - Volume 3, ICIP '97, pages 78–, Washington, DC, USA (1997). IEEE Computer Society. 9

Oliva, A. and Torralba, A. (2007). *The role of context in object recognition*. Trends in Cognitive Sciences, 11(12):520–527. 13

Oshin, O., Gilbert, A., Illingworth, J. and Bowden, R. (2009). *Action recognition using randomised ferns*. In Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, pages 530–537. IEEE. 11

Perše, M., Kristan, M., Pers, J. and Kovacic, S. (2006). *A template-based multi-player action recognition of the basketball game*. In In: Janez Pers, Derek R. Magee (eds.), Proceedings of the ECCV Workshop on Computer Vision Based Analysis in Sport Environments, Graz, Austria, pages 71–82. 11

Perše, M., Kristan, M., Kovačič, S. and Perš, J. (2008). *A Trajectory-Based Analysis of Coordinated Team Activity in Basketball Game*. In Computer Vision and Image Understanding. 11

Poppe, C., De Bruyne, S., Verstockt, S. and Van de Walle, R. (2010). *Multi-Camera Analysis of Soccer Sequences*. In 2010 Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance, pages 26–31. 9, 10

Raptis, M. and Soatto, S. (2010). *Tracklet descriptors for action modeling and video analysis*. In Computer Vision-ECCV 2010, pages 577–590. Springer. 29

Remagnino, P., Tan, T. and Baker, K. (1998). *Multi-agent visual surveillance of dynamic scenes*. Image and Vision Computing, 16(8):529–532. 9

Ryoo, M. S. and Aggarwal, J. K. (2010). UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html. Accessed: 2014-02-12. 14

Ryoo, M. S. and Aggarwal, J. K. (2009). *Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities*. In Computer Vision, 2009 IEEE 12th International Conference on, pages 1593–1600. IEEE. 12

Schyns, P. G. and Oliva, A. (1994). *From blobs to boundary edges: Evidence for time-and spatial-scale-dependent scene recognition*. Psychological Science, 5(4):195–200. 12

Scovanner, P., Ali, S. and Shah, M. (2007). *A 3-dimensional sift descriptor and its application to action recognition*. In Proceedings of the 15th international conference on Multimedia, pages 357–360. ACM. 11

Scovanner, P. and Tappen, M. F. (2009). *Learning pedestrian dynamics from the real world*. In Computer Vision, 2009 IEEE 12th International Conference on, pages 381–388. IEEE. 12

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA. ISBN 0521813972. 32

Shutler, J. D., Nixon, M. S. and Harris, C. J. (2000). *Statistical Gait Recognition via Velocity Moments*. In IEE Colloquium: Visual Biometrics, number 018 in 2000, pages 11/1–11/5. IEE. 9

Smith, P., da Vitoria Lobo, N. and Shah, M. (2005). *Temporalboost for event recognition*. In Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, volume 1, pages 733–740. IEEE. 11

Spagnolo, P., Mosca, N., Nitti, M. and Distante, A. (2007). *An Unsupervised Approach for Segmentation and Clustering of Soccer Players*. In International Machine Vision and Image Processing Conference, 2007. IMVIP 2007., pages 133–142. IEEE. 9

Tang, J., Chan, J. and Leung, H. (2011). *Interactive dancing game with real-time recognition of continuous dance moves from 3D human motion capture*. In Proceedings of the 5th International Confernece on Ubiquitous Information Management and Communication, page 50. 9

TeX Users Group (2004). *TeX Users Group Home Page*. http://www.tug.org/.

Thurau, C. and Hlavac, V. (2008). *Pose primitive based human action recognition in videos or still images*. In 2008 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8, Anchorage, AK, USA (2008). 11

Tomasi, C. (2004). Estimating Gaussian Mixture Densities with EM - A Tutorial. 23

Tran, D. and Sorokin, A. (2008). *Human Activity Recognition with Metric Learning*. In Proceedings of the 10th European Conference on Computer Vision: Part I, ECCV '08, pages 548–561, Berlin, Heidelberg (2008). Springer-Verlag. 11

Tu, Z. (2008). *Auto-context and its application to high-level vision tasks*. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8. IEEE. 13

Turaga, P., Chellappa, R., Subrahmanian, V. and Udrea, O. (2008). *Machine Recognition of Human Activities: A Survey*. IEEE Transactions on Circuits and Systems for Video Technology, 18(11): 1473–1488. 10

Uebersax, D., Gall, J., den Bergh, M. V. and Gool, L. V. (2011). *Real-time Sign Language Letter and Word Recognition from Depth Data*. In IEEE Workshop on Human Computer Interaction: Real-Time Vision Aspects of Natural User Interfaces, pages 383–390. 9

University of Central Florida (2008). UCF Sports Action Dataset. http://vision.eecs.ucf.edu/datasetsActions#UCF%20Sports%20Action%20Dataset. Accessed: 2014-02-12. 14

University of Central Florida (2009). UCF Youtube Action Dataset. http://vision.eecs.ucf.edu/datasetsActions#UCF%20YouTube%20Action%20Dataset. Accessed: 2014-02-12. 14

University of Surrey and CERTH-ITI (2009). i3dpost multi-view human action datasets. http://kahlan.eps.surrey.ac.uk/i3dpost_action/. Accessed: 2014-02-12. 14

Urtasun, R., Fleet, D. and Fua, P. (2005). *Monocular 3D tracking of the golf swing*. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 2, pages 932–938. 10

Video Computing Group. (2010). Videoweb dataset. http://www.ee.ucr.edu/~amitrc/vwdata.php. Accessed: 2014-02-12. 14

Visual Geometry Group (2010). TV Human Interactions Dataset. http://www.robots.ox.ac.uk/~vgg/data/tv_human_interactions/index.html. Accessed: 2014-02-12. 14

Wang, L. and Suter, D. (2007). *Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model*. In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, pages 1–8. IEEE. 11

Weickert, J. and Schnörr, C. (2001). *Variational Optic Flow Computation with a Spatio-Temporal Smoothness Constraint*. J. Math. Imaging Vis., 14(3):245–255. 29

Weinland, D., Ronfard, R. and Boyer, E. (2011). *A survey of vision-based methods for action representation, segmentation and recognition*. Computer Vision and Image Understanding, 115(2): 224–241. 6

Wu, Y. and Huang, T. (1999). *Vision-Based Gesture Recognition: A Review*. In Braffort, A., Gherbi, R., Gibet, S., Teil, D. and Richardson, J., editors, Gesture-Based Communication in Human-Computer Interaction, volume 1739 of *Lecture Notes in Computer Science*, chapter 10, pages 103–115. Springer Berlin / Heidelberg, Berlin, Heidelberg. 9

Yu, X. and Farin, D. (2005). *Current and emerging topics in sports video processing*. In International Conference on Multimedia and Expo, pages 526–529. 9

Yuan, J., Liu, Z. and Wu, Y. (2009). Discriminative video pattern search for efficient action detection. http://users.eecs.northwestern.edu/~jyu410/index_files/actiondetection.html. Accessed: 2014-02-12. 14

Zelnik-Manor, L. and Irani, M. (2001). Weizmann event-based analysis of video. http://www.wisdom.weizmann.ac.il/~vision/VideoAnalysis/Demos/EventDetection/EventDetection.html. Accessed: 2014-02-12. 14

Zhu, G., Xu, C., Huang, Q., Gao, W. and Xing, L. (2006). *Player action recognition in broadcast tennis video with applications to semantic analysis of sports game*. In Proceedings of the 14th annual ACM international conference on Multimedia, pages 431–440. 10

Zhu, Y., Nayak, N. M. and Roy-Chowdhury, A. K. (2013). *Context-Aware Modeling and Recognition of Activities in Video*. In Computer Vision and Pattern Recognition (CVPR), Portland,OR,USA (2013). 12