UNIVERSITY OF TROMSØ UIT

FACULTY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE

Precise Video Feedback through Live Annotation of Football

Johan Grønvik

INF-3990
Master's Thesis in Computer Science
May, 2012

Abstract

The domain of sports analysis is a huge field in sports science. Several different computer systems are available for doing analysis, both expensive and less expensive. Some specialize in specific sports such as football or ice hockey, while others are sports agnostic. However, a common property of most of these systems is that they try to give in-depth and detailed analysis of the sport in question.

This thesis proposes and describes a system that provides the user with the ability to annotate interesting happenings during a live sporting event, through a non-invasive mobile device interface. The device permits focus on important happenings by filtering out unnecessary detail. Our system provides corresponding video of the annotations on the same mobile device, thereby facilitating the process of giving video feedback to the involved coaches and players.

We have implemented a prototype of the system that enables evaluation of this idea, and through case studies with Tromsø Idrettslag, a Norwegian Premier League football club, we show its usefulness and applicability.

Acknowledgement

I would like to thank my supervisor Professor Dag Johansen who has provided me with much needed inspiration and motivation through a long process. It has proved invaluable during the work on this thesis.

A big thanks to the contributors from Tromsø Idrettslag; Truls Jensen, Agnar Christensen and Svein-Morten Johansen.

I would also like to thank the iAD group at UiT, for continued support and valuable input, with a special thanks to Magnus Stenhaug, Roger Hansen, Anders Gjerdrum and Joe Hurley for contributions that has been very helpful.

My gratitude goes to fellow students Øyvind, Erik, Arild and Simen for good ideas, good discussions and great times.

Finally, a big thank you to my family, especially my father who has helped me tremendously through the final stages of this work.

Table of Contents

Chapter 1 Introduction 1
1.1 Problem definition 2
1.2 Interpretation 2
1.3 Methodology 3
1.3.1 Theory..3
1.3.2 Abstraction..3
1.3.3 Design 3
1.4 Context..4
1.4.1 Tromsø Idrettslag (TIL) 4
1.4.2 iAD-group at the University of Oslo..4
1.5 Organization 5

Chapter 2 Background..7
2.1 Introduction..7
2.2 Low-level meta-data 8
2.2.1 Video tracking..8
2.2.2 Ball tracking problems 9
2.2.3 Sensor based tracking 9
2.3 High-level meta-data 10
2.3.1 Sports commentary notations 10
2.3.2 Sports analysis notations.. 11
2.4 Example systems 12
2.4.1 ZXY Sport Tracking 12
2.4.2 ProZone.. 12
2.4.3 Interplay-Sports 13
2.4.4 Muithu 14
2.4.5 DAVVI.. 15
2.5 Summary 15

Chapter 3 System architecture 17
3.1 User-centred processes..17
3.1.1 Assistant coach (Agnar Christensen) 18
3.2 System model 19
3.2.1 Concepts 19
3.2.2 Sessions.. 21
3.2.3 Events.. 21
3.3 Architecture..22
3.3.1 Annotation component 22
3.3.2 Video provider component.. 23
3.3.3 Video playback component 24
3.4 Summary 25

Chapter 4 Design and Implementation 27
4.1 Technical specifications..28
4.1.1 Tablet 28

4.1.2 Camera System #1 (Static cameras) 28
4.1.3 Camera System #2 30
4.2 Android application (*Vuvuzela*) 31
4.2.1 Android framework and programming patterns 31
4.2.2 Overview 32
4.2.3 Data model and storage 34
4.2.4 Event registration interface 35
4.2.5 Event viewer interface 38
4.2.6 Communication service 39
4.3 Camera systems 40
4.3.1 Static camera system with ZXY-integration (System #1) 40
4.3.2 Position flexible camera system (System #2) 42
4.4 Summary 42
Chapter 5 Case study and experiments 43
5.1 Introduction 43
5.2 Live tests at Alfheim Stadium 43
5.2.1 The role of the test-user 44
5.2.2 Annotation of events through Vuvuzela 45
5.2.3 Video recording of the matches 46
5.3 Experiments with Camera system #1 47
5.4 Summary 49
Chapter 6 Evaluation 51
6.1 Case study evaluation 51
6.1.1 Video relevance 51
6.1.2 Identifying emerging trends 55
6.1.3 Comparison to Interplay-Sports 56
6.2 Experiments with Camera system #1 57
6.2.1 Following a single player 58
6.2.2 Following multiple players 59
6.3.3 Encoding performance and file size 59
6.3 Summary 61
Chapter 7 Conclusion 63
7.1 Achievements 63
7.2 Related work 64
7.3 Concluding remarks 64
7.4 Future work 64
Appendix A 69
CD-ROM 69

List of Figures

Page

2.1 Separation of data sources in terms of meta-data relevance 8

2.2 The user interface of a typical Interplay-Sports session 14

2.3 Classification of related systems in terms of meta-data relevance and feedback type 16

3.1 Configuration of our annotation system 20

3.2 Overview of the system architecture 22

3.3 Conceptual representation of our static camera system mapped to positional data of a player 24

3.4 The need for offline storing of videos on the mobile device, in a half-time break scenario 25

4.1 Implementation overview 27

4.2 The static cameras cover the whole field from one platform 29

4.3 Examples of camera positioning in the flexible camera system 30

4.4 Vuvuzela android application implementation overview 33

4.5 Data model in Vuvuzela-android application 34

4.6 Early event registration interface design 36

4.7 Event registration interface. Red text and arrow is not part of actual screenshot 37

4.8 Session and event viewer interface 38

4.9 The interface displaying a list of videos corresponding to an event 39

4.10 Illustration of ZXY-coordinate to pixel mapping 41

5.1 Truls Jensen with his current analysis tools 44

5.2 Drag and drop interface for event registration, as used in the case study 46

5.3 Positions of cameras and user during test-matches at Alfheim Stadium 47

5.4 Simulation of real-time match annotation 48

6.1 Mismatch between an annotated event and the actual event 52

6.2 A snapshot from the two cameras showing the same event 53

6.3 Distribution of videos per match in two test cases from April 2012.
The first match to the left. 54

6.4: Position of Svein-Morten Johansen when live annotating
through Interplay-Sports 57

6.5 Tracking a single player in an event 58

6.6 Tracking multiple players involved in an event 59

List of Tables

Page

2.1 Example of sports commentary notations from the Norwegian website VGlive 13

3.1 The user-centred design: techniques, purposes and stages with corresponding involvement and collaborators 16

5.1 Keywords defined by Truls Jensen during case study 45

6.1 Distribution of keywords in the first match of the case study 55

6.2 Time taken to encode videos, and the resulting file size 60

Chapter 1 Introduction

In modern professional sports, management and coaching depend heavily on recording and evaluating individual and team achievements [1]. Systematic notations are widely used for compiling objective data on the performance of athletes during training and, particularly, during contests and matches. The use and development of notation systems followed the publication of an influential report by Franks and Miller in 1986 [2] that showed that international football coaches only remembered 30 % of successful incidents in the matches. This suggested that using only unsystematic observation was unreliable and inaccurate, and the evidence promoted the need for recording sports performance with some kind of systematic, objective and reliable notation [3]. Specifically, notational analysis today focuses on movement, technical and tactical evaluation, and gathering of statistical data [1]. Technological improvements and demands of coaches lead to increasingly sophisticated systems of notational analysis. Based on several recent papers, issues of current relevance for coaches, such as sample sizes, operational definitions, performance over time, invariant behaviour and perturbations are discussed in detail by James [3]).

Already in 1997, Olsen and Larsen [4] reported that nearly all teams in the Norwegian premier football league Tippeligaen used some kind of match analysis as management tools. In 2004, reports from England [5, 6] showed that notational analysis was almost universal in the Premier League. In professional football most teams in the world currently combine some kind of video feedback in their performance analyses. In the English leagues all matches are videotaped and the managers routinely get the tapes right after the matches and use them for post-event analysis [3].

Today's standards in this field of sports analysis demand considerable amounts of manual labour. Dedicated teams within a sporting organization often do this. In most cases these analysis teams consist of several people that spend many hours analyzing video footage. One of the problems with this is the cost of manpower, which gives analysis a steep price point for smaller sporting organizations. There are systems that try to solve this problem, but they often provide user interfaces that require you to sit down with a laptop, or in other ways make it too time consuming to annotate events, for instance Interplay-Sports, see Chapter 2.4.3 and [17].

Feedback is essential in sports analysis. Arguably, any change in performance is not going to occur without any kind of feedback. The quality of the feedback is also important, which infers the need for precision. Notational analysis can be seen as a way of objectively recording performance in terms of critical events, thus identifying what went wrong (or right) in a certain situation. The ability to play back video clips that correspond to certain, predefined (annotated), types of critical events, involving a player or group of players could be used to give the involved players precise and accurate feedback on how to improve performance.

1.1 Problem definition

This thesis shall build and evaluate a system for live notational analysis and video feedback of sporting events, with football as the specific domain. The focus will be on implementing a system that provides an annotation interface for registering events and gives precise and useful feedback through video corresponding to the annotated events. Another property of the system should be the ability to reduce the amount of video data that needs to be stored and transferred through use. The system will be deployed at Alfheim Stadium, the home ground of the Norwegian Premier League (Tippeligaen) football team Tromsø Idrettslag (TIL). The implemented prototype will subsequently be tested and evaluated by coaches from TIL.

1.2 Interpretation

Our thesis is that by using a mobile device for event annotation during a match, thus highlighting interesting situations on the fly, it is possible to process these events immediately and synchronize them with corresponding video sequences. We also look at using the same device to play back video of the annotated events to give feedback to both coaches and players through the same interface that was used for annotation.

To evaluate this idea we will design and implement a mobile device application-prototype that will provide an annotation interface and playback capabilities for use during sporting events. The video recording process will use both static cameras and cameras that can be flexibly moved to several positions around the field.

We believe that the precision of the feedback is closely related to the user of the system. We therefore consider the participation of coaches from TIL an important part of the design, the implementation and the evaluation process of the prototype.

1.3 Methodology

The final report of the ACM Task Force on the Core of Computer Science divides the discipline of computing into three major paradigms [7]. These paradigms are theory, abstraction and design. The following is a short summary:

1.3.1 Theory

Theory is the mathematical approach rooted in development of valid mathematical principles. Theorems about objects are proposed, and you seek to prove them in order to find new relationships and progress in computing.

1.3.2 Abstraction

Abstraction is rooted in the experimental scientific method. The approach is to construct models and state hypothesis, and evaluate these by simulation, thereby analyzing the results.

1.3.3 Design

Design can be described as the engineering approach. Using this method you state requirements and specifications, thereafter design and implement a system that solves the problem at hand. The system is then tested systematically according to the stated specifications and requirements, and finally evaluated.

For this thesis, the most suited paradigm is design. We have stated a specific problem and will design and implement a system to solve it. The system will then be tested systematically and evaluated according to the problem earlier stated.

1.4 Context

This thesis is part of the information Access Disruption (iAD) project. The iAD Centre is partly funded by the Research Council of Norway, is directed by Microsoft Norway and works by collaboration between commercial companies and several universities (Cornell, Dublin City, Oslo, Trondheim and Tromsø). The Centre's focus is on core research for next generation precision, analytics and scale in the information access domain. iAD's former work includes DAVVI [8], where this is explored in a video context. DAVVI is described as "A prototype for the next generation multimedia entertainment platform"[8]. More specifically DAVVI is a system for search and recommendation within the soccer video domain. It aims to "provide a personalized, topic-based user experience blurring the distinction between content producers and consumers"[8]. DAVVI annotates video by analyzing text-based commentaries from sport websites. By correlating video of a football match to the events annotated by the commentators of said websites it is possible to extract specific video segments of a certain type of event and subject (football player in this context). These segments are served to the user as a playlist of videos corresponding to the subject and event type that was searched for.

1.4.1 Tromsø Idrettslag (TIL)

This thesis is focused on providing an annotation interface on a mobile device (tablet) to coaches, as well as providing video playlists similar to DAVVI. Our main partner in this work has been Tromsø IL, the silver medallist in the 2011 season of Tippeligaen. More specifically TIL's assistant coach, Agnar Christensen, has provided us with useful feedback and discussion during this last year of work. The prototype we have implemented is designed with these sessions as a fundament, and have been tested and evaluated by Truls Jensen. Truls Jensen is the main player developer at TIL, and part of his job is to analyze the team during matches as well as giving the team feedback on their performance both during half-time breaks and in a post match setting.

1.4.2 iAD-group at the University of Oslo

Parts of the system we propose use components that were developed by researchers from the iAD-group at the University of Oslo. This includes the recorders that are used to record video from the static cameras installed on Alfheim Stadium, and the server side component that processes and provides video to the mobile device. These components are described in more detail later.

1.5 Organization

The remainder of this thesis is organized as follows. Chapter 2 introduces work that is relevant to our thesis. We examine commercialized systems that are used in the domain of football analysis today, including our partners at TIL. Research work done within video annotation and analysis is presented at the end of the chapter. Chapter 3 describes the system architecture from an overview perspective and specifies the design concepts of the system. Based on the architecture presented in Chapter 3, Chapter 4 gives a detailed description of the technical properties, design and implementation of the prototype application as developed in this work. Chapter 5 tests the usefulness of our prototype by experiments and a case study from matches at Alfheim Stadium, in close co-operation with TIL. Chapter 6 presents and discusses case studies and experiments with applying our system to recordings from three live matches at Alfheim Stadium. Chapter 7 concludes our work.

Chapter 2 Background

This chapter introduces related work that is relevant to our thesis. We have examined commercialized analysis systems that are in use by football organizations today, such as our partners at TIL. Towards the end of the chapter we describe work done by researchers in the field of video annotation and sports analysis.

2.1 Introduction

In notational sports analysis we define notations as meta-data that describes situations on the field. The notations can have different levels of detail and relevance. Our definition divides the notations into two categories, depending on the processes that produce the meta-data. We define the first category as notations that are produced by automated processes and the second category as notations produced by user operated processes. A human user has a better understanding of the semantics surrounding a situation than an automated process, which implies that a notation of the second category is elevated in terms of relevance above the first category. We therefore separate **High-level** and **Low-level** notations containing meta-data. Figure 2.1 explains how we categorize different sources of meta-data in the context of football notations.

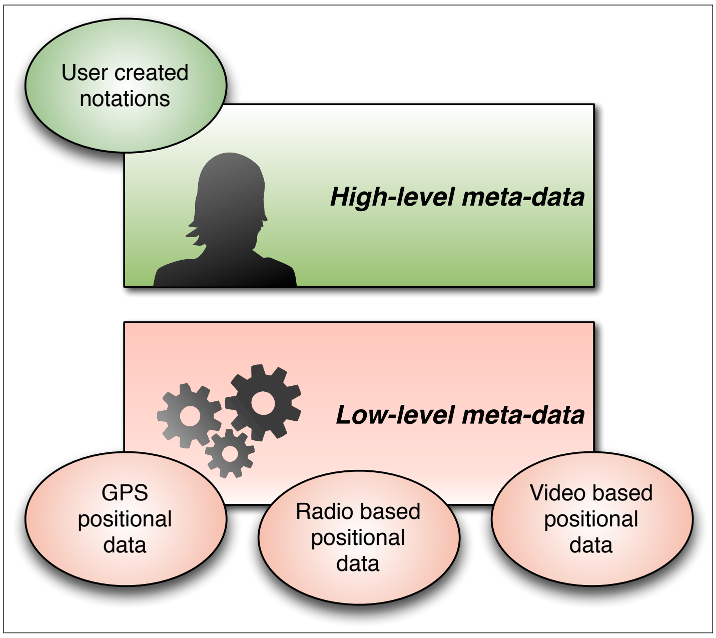

Figure 2.1: Separation of data sources in terms of meta-data relevance

2.2 Low-level meta-data

Low-level meta-data is produced by **automated** systems that process data from sensors equipped by the players, or from video based tracking that identify events from image analyzing.

2.2.1 Video tracking

Video tracking is the process of locating and identifying moving objects over time by analyzing streams of video data. Several techniques and algorithms exist in this field and they all have their strengths and weaknesses. Specific approaches have been developed in the domain of football analysis. The purposes of these approaches are typically to identify players and to track their position on the field. Player tracking is explained as multi-object tracking because several players act in a football match. In the scope of this thesis we will not compare the different approaches, however it is important to know that they are quite complex and computationally expensive [9]. This is especially true when working in high definition video and high frame rates. Uniquely identifying who or what we are tracking adds to the complexity of the process. In certain cases this is a soluble problem, for instance one could identify a tracked car by its license plate, but it is harder to identify a tracked person. Several automatic methods exist for identifying persons in video [10, 11] but we argue that the accuracy and performance of these methods is too low, especially if there are multiple subjects in a video frame.

2.2.2 Ball tracking problems

Even though ball tracking belongs to single-object tracking while player tracking falls within multi-object tracking, ball tracking is not easier than player tracking for several reasons. Usually ball blobs in images are very small, which makes it difficult to distinguish from other features, such as markings on the field. The way a ball suddenly changes its motion is another factor that makes it challenging. In addition, occlusion and overlapping with players causes a severe problem in tracking the ball continuously in video [12].

FIFA (Fédération Internationale de Football Association) has been testing several approaches to solve the problem of defining if the ball has passed the goal line or not. This problem occurs if the referee did not clearly see the situation, and is often the cause of heated debates after a match where the referee made a wrong decision. Some of the systems proposed use sensors in the ball that helps track its position (Cairos GLT[1]), while others employ elaborate video based approaches in the goal area (GoalMinder[2]). A system where the ball could be tracked through the whole match could prove very useful for a sports analysis system such as the one we propose. However, FIFA has not yet allowed any of the approaches that make this possible [see 13].

2.2.3 Sensor based tracking

In addition to low level video based systems, there are sensor-based systems that produce similar positional data. These systems typically record data on the position of players at any given time. Players are fitted with sensors of different kinds, depending on the method the system uses. Some systems use radio signals

---

[1] Cairos GLT - http://www.cairos.com/unternehmen/gltsystem.php

[2] GoalMinder - http://www.goalminder.co.uk/

from sensors carried by the players to determine positions (ZXY) while other systems equip players with GPS sensors to gather similar data (GPSports[3]). The data are then processed to determine player movement in terms of acceleration, speed and direction. The sensor-based systems often employ monitors to record the heart rate of the players in addition to the movement. Most of the systems have an application component that collects and displays the sensor data to the user in a way that makes it more understandable to humans.

2.3 High-level meta-data

High-level meta-data is produced by systems that employ **human perception** as the method to detect events. A user of the system either creates notations while watching a recording of a football match (post-match annotation) or creates notations live, during the match. This category of meta-data differs from low-level meta-data in the perceived quality of the data. A low-level notation typically describes physical facts on player movement and positioning, while a high-level notation relates closely to aspects of the sport that is being analyzed. For instance, high-level notations in the context of football could describe quality of passing, how players are positioned in certain situations, the outcome of set pieces, the performance of the keeper, and so on. Depending on the level of expertise of the user, we have different ontologies describing football. A commentator for a sports website would for instance describe a situation differently than a coach, and would use a different ontology.

2.3.1 Sports commentary notations

The role of a website covering a football match with live commentary is to provide the readers of the website with the interesting events during the match. However, note that the word *interesting* will differ in meaning when compared to what is interesting for a coach in terms of providing useful feedback to players. This means that while sports commentary falls within the category of high-level meta-data, it will not suffice as analysis notations for the purpose of a coach. An

[3] GPSports – www.gpsports.com

example of this is seen in Table 2.1, taken from VG Live[4]. The second notation in the figure says *"It snows heavily on Alfheim"*. While this information might be amusing and interesting to a football fan, it is probably not important in the analysis of the match from the perspective of the coach.

9	Nok en corner for hjemmelaget.
9	Det snør kraftig på Alfheim.
9	Corneren detter ned foran nesa på Koppinen men Knudsen plukker den like før midtstopperen får skutt.
12	Banemannskapet på Alfheim er nå ute og måker for de viktigste linjene på banen. Det er 16 - og 5-meteren.
14	Hussain går i bakken etter å ha driblet av en mann i sekstenmeteren. Forsvareren er riktignok på ballen og det dømmes ikke straffe. Men sjansen kom etter en god FFK-kontring og det er nok der bortelaget må satse på de store sjansene.
16	Fredrikstad er mer med i kampen nå. Klarer å sette et nogenlunde høyt press på hjemmelaget til tider.

Table 2.1: Example of sports commentary notations from the Norwegian website VGlive

2.3.2 Sports analysis notations

In addition to high-level notations created by persons such as sports commentators we have high-level notations that are created for use in sports analysis rather than in entertainment for sports fans. Typically such notations use another ontology than the notations described above. An ontology submitted to DAML[5] defines nearly 200 different classes for annotation of football matches. While this is very detailed, and may prove useful to get a very well described summary of a match, it is arguably hard to do in a real time situation. Additionally, for the kind of analysis we propose, a smaller and more coaching focused ontology is better suited.

---

[4] Norwegian live commentary website - www.vglive.no

[5] DARPA Agent Markup Language website - http://www.daml.org/ontologies/273

2.4 Example systems

This section will examine some of the systems that are in use today for sports analysis. We have chosen to focus specifically on football analysis because of the scope of this thesis.

2.4.1 ZXY Sport Tracking

The ZXY Sport Tracking is a radio-based positioning system that delivers information on physical and tactical player performances in real-time [14]. TIL employs the system at Alfheim Stadium. ZXY transmits data from sensors the players wear on a belt around their waist to a set of radio-receivers that are placed around the field. The sensors monitor the actions of the player on the field up to 40 times per second. The system allows real-time monitoring of parameters like position on the field, heading, effort and pulse [15]. The positioning data from the ZXY sensors are stored as Cartesian co-ordinates, where the co-ordinate system has its origin in one of the corner arcs on the field. In our thesis, we will combine ZXY-positional data with video from stationary video cameras (see Fig. 3.3). The latest version of the ZXY-belts also employs a heart rate monitor for recording physiological data.

2.4.2 ProZone

ProZone is a video-based, computerized tracking system aimed at analysing movement patterns in team sports, particularly football [16]. It allows motion measurement and tracking of all players and referees on the field in real time. It is based on tapes from multiple video cameras that are placed at fixed positions around the field. An operator transfers the video files to dedicated servers that instigate automatic tracking of the files and combine them to one dataset. The video images are then converted into field co-ordinates via a calibration process. Operators identify individual players by start position, position during the game and correspondence with an outside broadcast feed, and verify them during the game.

A test of the system was performed by comparing ProZone results of taped test-runs with results from time-gates that were placed at the start and the end of the test-runs [16]. The correlation coefficients were very high, showing that ProZone allows reliable tracking of all players in a football game. The disadvantage is that the costs are very high and that it is necessary to have one or several skilled and dedicated operator(s) to run the system and perform the analysis.

2.4.3 Interplay-Sports

Because much of this work is done in close cooperation with TIL it is important to understand the systems that are in use at Alfheim today and how they compare to what is proposed here. Interplay Sports is a video analysis system in use by TIL. The system is mainly used by Svein-Morten Johansen, who works as a part time match analyst at TIL. It is used to produce in depth and detailed analysis of matches in a post-match scenario. Through discussions with Johansen we have been given demos and have been explained how Interplay is used.

Interplay-Sports is described as “a standalone system but also a complementary and mobile component in cooperation to the large and expensive video analyze tools as ProZone, Amisco, ZXY and TrackAB.” [17]. It is an application developed for Windows operating systems where the user applies self-defined variables to situations in a football match. An important property of the system is the close relation between the notations and the video. As such, an annotation does not exist outside the context of the actual video file it describes.

Another characteristic of analysis through Interplay is how detailed it enables the user to annotate plays. You can add up to eight variables to a single situation, including what players were involved in, as well as defining who passed the ball to whom. This type of in-depth analysis is made possible by a user interface with many options and interaction elements. Figure 2.2 shows the user interface of a typical Interplay session. It displays the video picture from the field, the tracking lines (lower screen) and a situation list (right screen).

Figure 2.2: The user interface of a typical Interplay-Sports session [6]

An Interplay session consists of loading a video file from some source and tagging situations in that video through the Interplay user interface. Typically footage produced for television is used as the video source. Interplay enables tagging of events both in a post-match scenario, where the user uses stored video footage of the match, and live tagging during the match. This is made possible by connecting the system directly to a camera and annotating the video stream in real time. To do this during matches on Alfheim, Svein-Morten situates himself on the camera platform from where the TV-camera crew is filming the matches and connects a laptop to one of the crew-operated cameras. This entails that the analysis is directly related to the footage that single camera provides.

2.4.4 Muithu

In close co-operation with the present work, the Department of Computer Science at the University of Tromsø has developed a portable, light-weight video based system called the Muithu sports notational system. In contrast to the

[6] Interplay-sports screenshot - www.interplay-sports.com

present system, it is based on coach notation on mobile telephones, not the tablet we develop here and provide to the users in the TIL coaching team. Also, we include both stationary and portable video cameras in our work while Muithu employs only portable cameras. The camera system developed for Muithu is integrated with the system we propose. Muithu is described in detail in a manuscript by Dag Johansen and co-workers [18].

2.4.5 DAVVI

DAVVI [8] is a next generation entertainment platform that provides a personalized user experience for multi-quality video content, for example from football match broadcasts. It is mainly aimed for the general public, but it may also find some use in professional football analysis because the users can annotate and analyse the videos with a set of extraction tools. Through applied search and advanced personalization and recommendation technologies the end-user can efficiently search and retrieve highlights in a customized manner.

In the DAVVI prototype [8], the unstructured commentaries and cites from football matches found on the Internet were converted to annotation metadata. This allows users to search for a much richer set of keywords. DAVVI then uses the annotations to create and return a playlist of events with event description, video object identifier and time interval. It relates to our work in the way it provides playlists of video sequences that are tailored for the user similar to what we propose for the users of our system.

2.5 Summary

This chapter has described some systems that are related to what we propose in the thesis. We have defined two levels of meta-data quality in the sports analysis domain. Our system is focused on using high-level notations to annotate matches and generate video sequences that show the annotated events. By using low-level meta-data from the ZXY-system that is deployed on Alfheim Stadium we can determine the correct camera feed from our static camera system, and subsequently produce video that is focused on the players involved in the annotations. This is made possible by mapping ZXY positional data to video streams from our cameras covering the field. Figure 2.3 places the current systems we have examined in terms of the meta-data categorization, and in relation to our system.

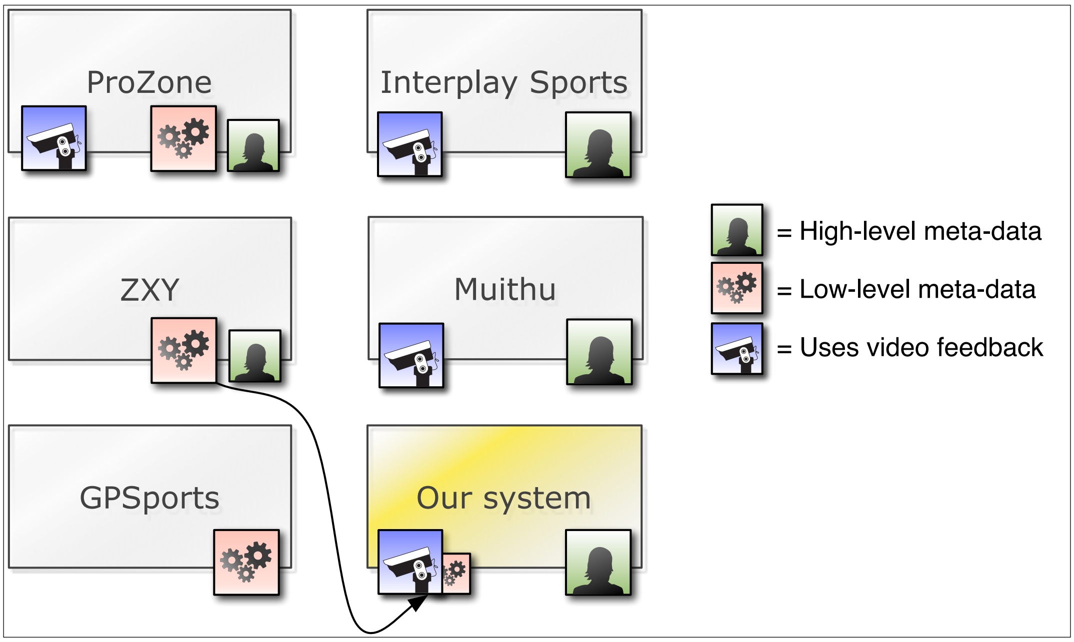

Figure 2.3: Classification of related systems in terms of meta-data relevance and feedback type

Chapter 3 System architecture

This chapter will describe the system from an overview perspective and specify the different concepts that were taken into consideration during the design of the system.

3.1 User-centred processes

'User-centered design' (UCD) is a broad term used to describe design processes in which end-users influence how a design takes shape. It is both a broad philosophy and variety of methods. There is a spectrum of ways in which users are involved in UCD but the important concept is that users are involved one way or another. For example, some types of UCD consult users about their needs and involve them at specific times during the design process; typically during gathering of requirements and usability testing. At the opposite end of the spectrum are UCD methods in which users have a deep impact on the design by being involved as partners with designers throughout the design process [19].

The user of the system is an integral part of the annotation system we propose. We describe the user as an expert filter that through the annotation interface identifies the events that are relevant for further examining. Already from the conception of this project we have cooperated with TIL, and we consider their knowledge and experience to have an important role in our work. The involvement of key people from TIL, such as the assistant coach Agnar Christensen, has had a deep impact on the way the annotation component was designed. In addition to the involvement from TIL, we have had internal sessions in the research group at the Department of Computer Science (DCS), University of Tromsø to further improve the design on certain stages.

Table 3.1 shows the design and development of the user-centred process in key-word format. The first three columns specify the different techniques, their purposes, and the stages in the design cycle as described by Preece et al. [20], starting with the background interviews and the initial collection of data and expectations of the users. For each stage, the last column details the coupling to the involvement of the DCS research group and iAD and, particularly, the sessions and users at TIL.

Technique	Purpose	Stage of design cycle	Involvement
Background interviews and questionnaires	Collecting data related to the needs and expectations of users	At the beginning of the design process	Agnar Christensen and Svein-Morten Johansen
Sequence of work interviews and questionnaires	Collecting data related to the sequence of work to be performed with the artefact	Early in the design cycle	Agnar Christensen and Truls Jensen
Focus groups	Many participants discuss issues and requirements	Early in the design cycle	Internal group, iAD
On-site observation	Collecting information concerning the environment in which the artefact is used	Early in the design cycle	Surveys and sessions on Alfheim stadium
Role Playing, walkthroughs, and simulations	Evaluation of alternative designs and gaining additional information	Early and mid-point in the design cycle	Internal group, iAD
Usability testing	Collecting data related to usability criteria	Final stage of the design cycle	Tests done with Truls Jensen during matches
Interviews and questionnaires	Collecting qualitative data related to user satisfaction with the artefact	Final stage of the design cycle	Interviews with Truls Jensen

Table 3.1: The user-centred design: techniques, purposes, and stages with corresponding involvements and collaborators

3.1.1 Assistant coach (Agnar Christensen)

Agnar Christensen works as the assistant coach on the A-team of Tromsø Idrettslag. He interacts closely with the players to improve their performance, and has extensive experience on how to give feedback to players and on the types of feedback that are most effective. Christensen has been a part of this project since the start, and we have had several meetings and work sessions to discuss system properties. Together with Christensen we identified important aspects that define the requirements of the system. A list of the requirements follows here:

- The system should be used to analyse matches (not training exercises).
- The system should handle real-time annotation.
- The system should focus on team events, but also enable player event-annotation for pre-selected players.
- The system should enable annotation and video playback on the same device.

3.2 System model

To understand the system model and architecture we propose, we explain concepts and ideas that define the scope of the system and the context in which it should operate.

3.2.1 Concepts

An *annotation* system is a system that produces annotations that correlate to a *data stream* as *meta-data*. The *meta-data* that are produced by the system describe the *data stream* with different *notations* (or tags) that later can be used to understand the data in the context where the annotation system and *data stream* exist.

For our purpose in this thesis we define the *context* as football match analysis. In this context we look at a football match as a series of *events* that are more or less interesting in terms of *meta-data relevance*. The users use the annotation system to identify the *events* that are interesting, thus reducing the amount of events that need to be stored for future reference as shown in Figure 3.1.

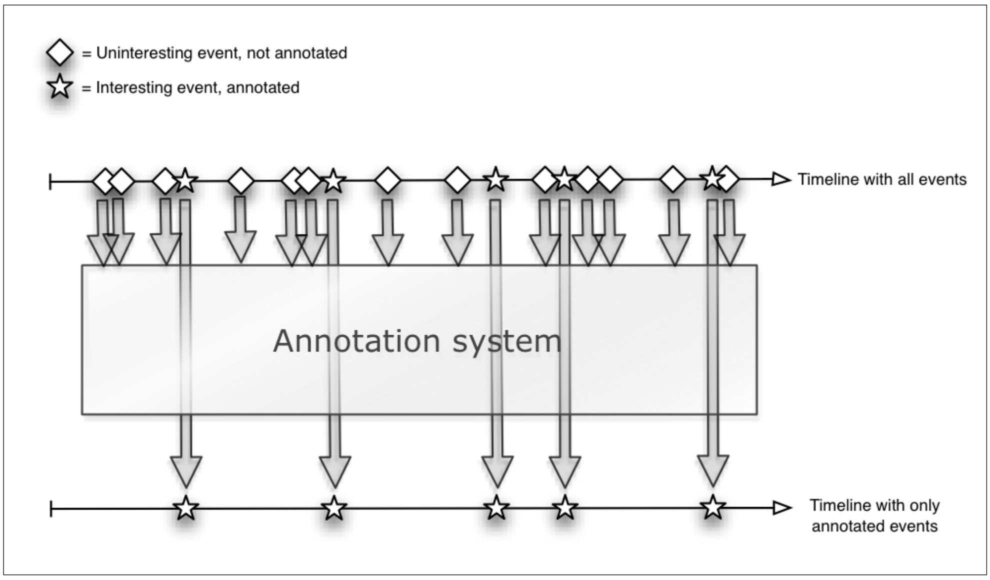

Figure 3.1: Configuration of our annotation system

As noted earlier, notations of videos from football matches can be made at different levels of experience and understanding of football as a sport. In our case we distinguish between low (automated) and high (human perception) levels of expertise. The number of notations tends to decrease with increasing level of expertise. At the low level (mostly crude, sensor-based notations), a high variety of events will be noted, small and large, important and unimportant. At the high level (coaches and analysts), annotations will be fewer because a strong professional focus filters out unimportant events (see Figure 3.1). TIL has several high level experts (assistant coach, player developer) who use the mobile devices we provide in our study to annotate home ground matches at Alfheim in real time. We define these high level notations as the *annotation system* in our thesis.

We consider videos of a football match as the *data streams* that the system should annotate. Today, multiple cameras cover most football matches, and for our test cases we have had several cameras recording the matches. As a product of that we have several data streams covering the same match. An integral idea in our system model is that a **single** stream of *meta-data* can be used to describe **multiple** video data streams as long as the different streams are **synchronized** in the time dimension, and the different video streams relate to the same football match.

We propose time to be represented as a timeline with a defined start point and a defined end point. All annotations produced by the system conform to this timeline-concept because the time when it was annotated is an integral part of each notation. We have decided to define a timeline of notations as a ***session***, and each notation that relates to the given timeline as an ***event*** generated during the session**.**

3.2.2 Sessions

A ***session*** in our system is specified as the time period in which a user actually generates events through the annotation interface that is provided. We divide a single match into two sessions, one per half match time. Each session relates to both the match half it covers and to the events that are generated during the session. Therefore, a match has *two sessions* and each session can have *zero to many events*.

3.2.3 Events

We define an ***event*** as a notable happening during the course of a session. An event has *three key properties*, the *time* it occurred, *who* was involved and a *keyword* (or key-sentence) that describes the actual situation of the event. To formalize the definition, we specify a general event as a happening that involves an object at a certain point in time.

In the context of this thesis we separate the object-property into two types. The objects of an event can be either a single player or several players involving a specific part of the team, or it can be the team as a whole. We have therefore decided to define both a *player-event* entity and a *team-event* entity. An important distinction between the two entities is the belt sensors that record the positions of the players on the field. Therefore, the players can be potentially tracked on the arena of the match if we pair video to the positional data. This means that we can extract video that is focused on the player without manually identifying the player from the different camera feeds. The positional data is gathered from sensor-based systems like the ones described in Chapter 2.4. This is further explained in Section 3.3.2.

3.3 Architecture

The system we propose consists of three components; an annotation component, a video provider component and a video playback component. A definition like this is helpful to explain the information flow of the system. First, the annotation component generates events that are sent to the video provider component, which provides videos of the generated events to the video playback component as dispalayed in Figure 3.2. Another reason for defining a three-component system like this is that each component handles a single task. Both the annotation component and the video provider component generate data that could potentially be used in other systems. This has been realised through correlation with the server component used in the Muithu-system (see Section 2.4.4), where events and videos are stored for reliable saving and reference outside the context of the mobile device.

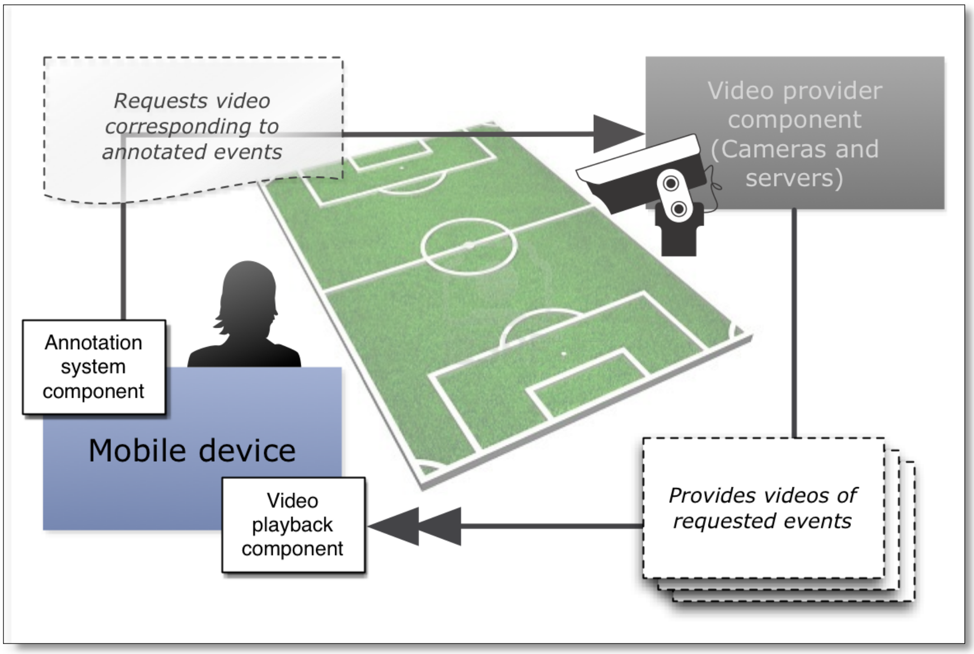

Figure 3.2: Overview of the system architecture.

3.3.1 Annotation component

We propose using a mobile device for event annotation. A reason for doing this is that it is, arguably, and particularly in the context where our system is used, more convenient to use a touch interface for event annotation than a classic keyboard and mouse interface,. As noted earlier, our system will be used during live football matches to annotate events in real time. The user will typically stand up (see Case study #1) and would therefore be unnecessary hindered by a device that is stationary.

We evaluated different kinds of mobile devices that could suit our system, specifically devices of different sizes. Modern, touch-based, mobile devices today generally consist of a screen covering the whole front and a few hardware buttons. The size of the device is therefore closely related to the screen size, which in turn plays an important role in how the interface is designed. Because both the annotation component and the video playback component of our system will run on the same device (see Figure 3.2), we concluded that a small device would defeat the purpose of usefulness in terms of video feedback. We therefore made the decision to use a device in the *tablet* category for our system.

The event registration interface was designed through workshops with TIL.

3.3.2 Video provider component

The video provider component in our system handles recording and delivering of video corresponding to the events generated by the annotation component. Throughout the work on this thesis we have used two approaches for this task. The two approaches differ on two key properties; position flexibility of the cameras and correlation with sensor data. The first system (System #1) uses static cameras that are mounted on a platform, while the second system (System #2) uses cameras that can be flexibly placed around the field for each session.

Because of the static nature of System #1 we conjecture that it is possible to integrate sensor-based positional data with the videos. This means that we can identify which camera is filming a player that is wearing a sensor. In our case, with positional data from ZXY, each player is represented by co-ordinates in a two dimensional plane representing the field. By mapping the co-ordinates to the areas covered by each camera, we can identify which camera zone a player is in at any time. Figure 3.3 explains this in a scenario where four static cameras are covering the field, dividing it into four camera zones. The figure shows an event happening in camera zone 4 (green), thus making the video captured by camera 1 and camera 2 (red) uninteresting for the given event. Because camera 3 (yellow) covers the neighbouring zone, it is possible that the footage it records may be useful for describing the event as well.

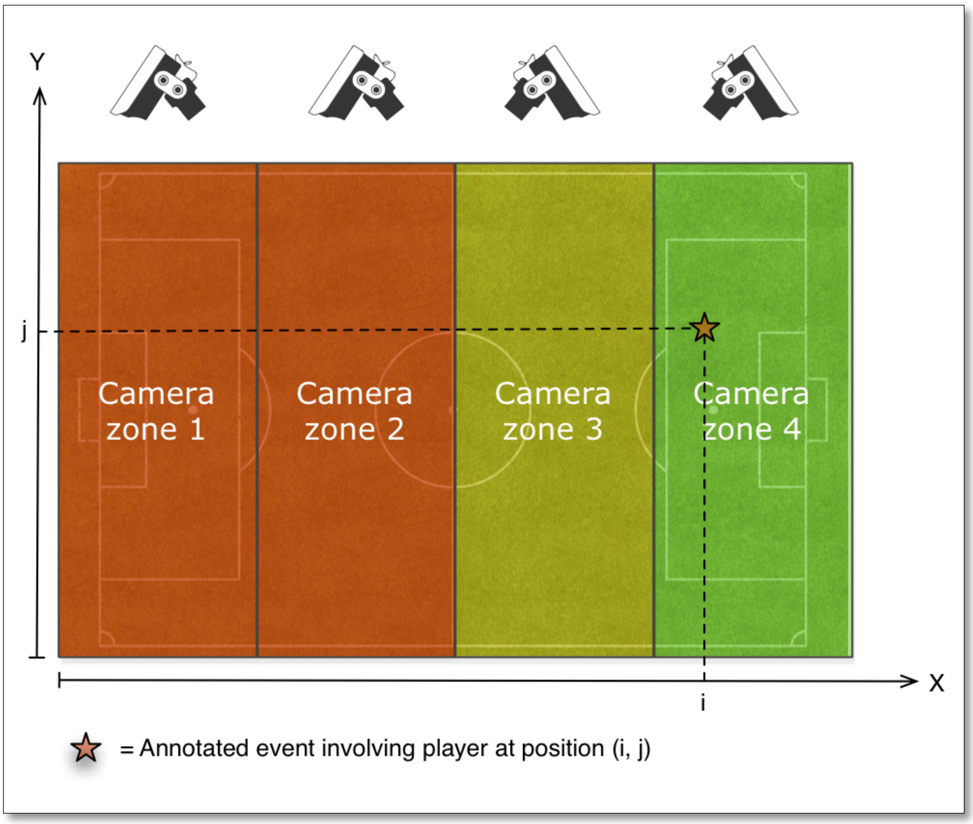

Figure 3.3: Conceptual representation of our static camera system mapped to positional data of a player.

System #2 uses movable cameras that are positioned on different locations around the field for each match. This makes it hard to do a precise mapping to positional data from sensor-based systems; we therefore need the user to evaluate which camera footage is applicable to a certain event.

Both System #1 and System #2 have been used for recording matches during this work and we have used video from both approaches when evaluating our system.

3.3.3 Video playback component

The video playback component in our system will run on the mobile device that is used for event annotation, as per the requirements we identified with Coach Christensen. We propose using a list of events displayed per session, and providing a video viewer interface per event. The device should store video files that are fetched from the video provider components, so that it is possible to view an event even if there is no network available. This means that the video needs to be loaded to the device when a network is available. To emphasize this, one should think of a scenario where the user of the system is placed in an elevated position on the stadium to get a good view of the field while annotating events. In this position, the device is connected wirelessly to a video provider component that can provide videos of annotated events during the match. To show these events to the players or coaches during half-time break in the locker room, the user brings the tablet out of the wireless range of the video provider component. This is explained in Figure 3.4.

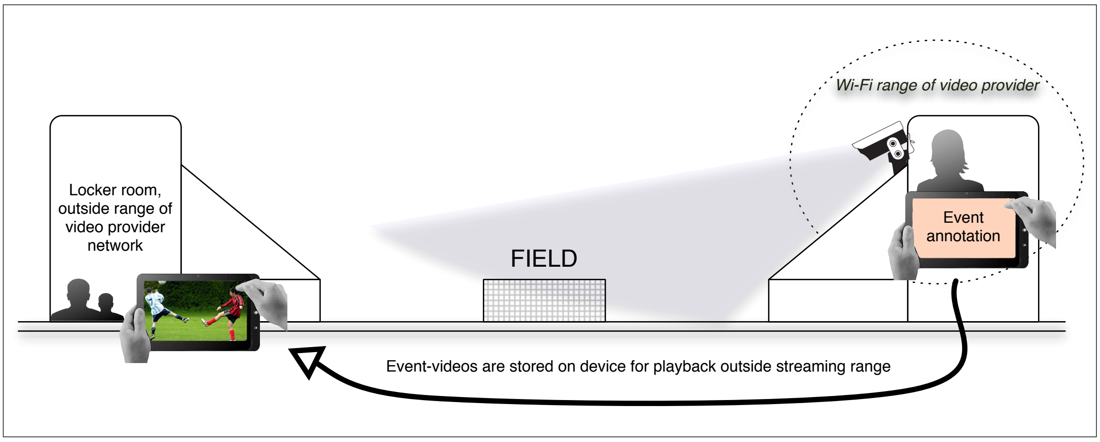

Figure 3.4: The need for offline storing of videos on the mobile device, in a half-time break scenario

3.4 Summary

This chapter has described our *annotation system* and how we define *sessions* and *events* in the context of our thesis. We have divided our system into three main components; *an annotation component, a video provider component and a video playback component*. In addition we have explained how an integral part of this work has been done in cooperation with our partners at Tromsø Idrettslag. This will be further explained in the evaluation and concluding chapters.

Chapter 4 Design and Implementation

This chapter describes the design and implementation of *Vuvuzela*, a prototype application built on the architecture described in Chapter 3. The name "Vuvuzela" comes from the plastic horns that were a huge topic during the FIFA 2010 World Championships in South Africa [21]. The vuvuzelas produce a loud, distinct noise, and we chose the name is an ironic reference to the purpose of the prototype: to remove the "noise" of uninteresting events and focus on events that can help produce relevant, and live, feedback during football matches. *Vuvuzela* is built to evaluate our thesis that live event annotation with corresponding video sequences can prove useful for precise feedback particularly during, but also after football matches. The focus in this implementation is to create a system that works as a proof-of-concept, with enough features that an evaluation of the architecture is possible. *Vuvuzela* consists of several components, and an overview of the implementation can be seen in Figure 4.1.

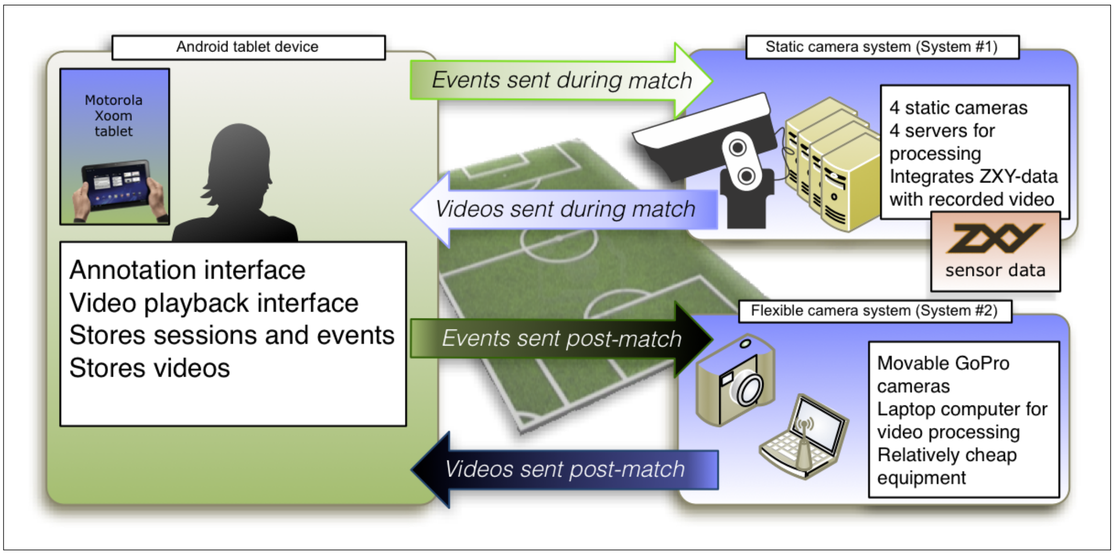

Figure 4.1: Implementation overview

4.1 Technical specifications

For the implementation of our prototype we have used the following technologies and devices.

4.1.1 Tablet

The tablet we chose for the mobile application is the Motorola Xoom. It has a 10, 1" capacitive multi-touch screen with a resolution of 1280x800 pixels. The operating system is Android 4.03, and we have used the Android Software Development Kit (SDK) provided by Google to develop the application in the open source Eclipse IDE. While Android applications are written mainly in Java, Google also provides a Native Development Kit (NDK) that allows running code written in C and C++. For our prototype we have not used the NDK, and the whole application is written in Java. The reason for this is that we have not run into any performance limitations with Java for our purpose.

4.1.2 Camera System #1 (Static cameras)

The cameras and servers used for camera system #1 is part of an ongoing project with the iAD-group from the University of Oslo (UiO). We have mounted four cameras on a platform on the north-western part of Alfheim Stadium. Figure 4.2 shows how they are mounted.

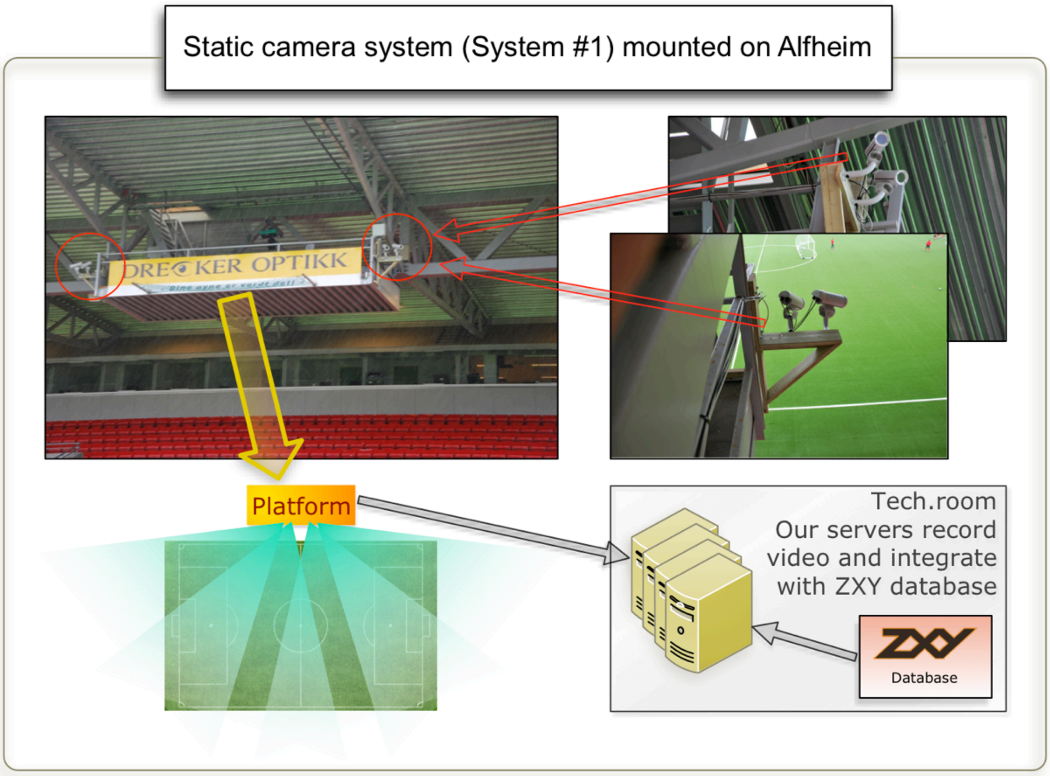

Figure 4.2: The static cameras cover the whole field from one platform.

The cameras are of type Basler Ace 1300[7], and are mounted inside weatherproof cases. They are capable of recording images at a 1296x966 pixels resolution at 30 frames per second and are described as small and powerful cameras that "*offer a unique combination of extremely high performance and low cost*" on Basler's website. Their Ace- range of cameras is targeted for industrial, medical and traffic applications. This differs from the consumer-targeted cameras used in our second camera system (System #2). Each camera is connected to a trigger box that handles frame-synchronization. This means that a camera that is connected to the trigger box records every frame in synchronization with the other cameras connected to the same box, which is crucial for combining the four camera feeds into one single feed through image stitching. Image stitching is the process of combining multiple images with overlapping fields of view to produce a segmented panorama or high-resolution image. We explain this further in Section 4.3.1.

[7] Basler Ace 1300 - http://www.baslerweb.com/products/ace.html?model=167

As there is no storage capability on the camera itself, the cameras employ a gigabit Ethernet interface for communicating image streams to connected computers. In our setup we have connected one server which handles recording and storing the video to each camera. The servers run the Linux based Ubuntu operating systems, and employs video recorders developed by the iAD-group at UiO.

4.1.3 Camera System #2

The second camera system we have used as a video provider is also part of the Muithu-system, which is correlated to our work with TIL. It consists of several cameras that can be flexibly placed around the field. Figure 4.3 shows examples of camera placement on Alfheim.

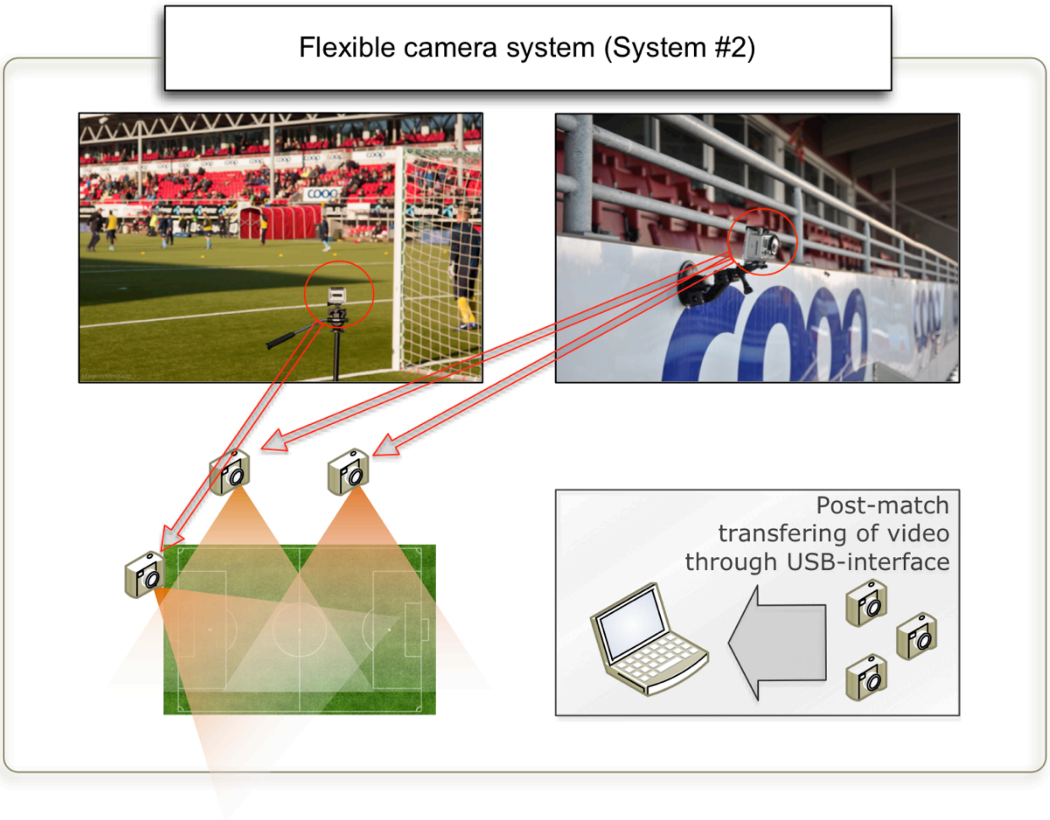

Figure 4.3: Examples of camera positioning in the flexible camera system

GoPro[8] produces the cameras used in this system, and we have used both their HD HERO and HD HERO2. Both are capable of recording video at 30 frames per second with a resolution of 1920x1080 pixels. HERO2 differs from HERO in lens sharpness (twice as sharp) and connectivity possibilities. Regarding connectivity, the HERO2 is meant to be able to stream video over Wi-Fi with an added Wi-Fi accessory. This accessory is not yet available, and has thus not been used in the present work. The cameras are very flexible in terms of positioning, and there are a host of different mounting accessories available from GoPro. Specifically in our tests we have used both tripods for camera positions on the level of the field, and suction-based wall mounts for camera positions from a more elevated perspective.

After the end of a match the cameras are collected and connected to a computer through an USB-interface for transferring the video files from the cameras. For our tests we used a single desktop computer, running Windows 7, to encode the videos.

Magnus Stenhaug and Roger B. Hansen, students at the University of Tromsø who are also part of the iAD-research group, have developed System #2 in large parts. We have used videos sequences from this system for evaluation of our prototype.

4.2 Android application (*Vuvuzela*)

The android application we built for the tablet handles both event registration and video playback. We have based the implementation on the decisions made through the design process that we described in Chapter 3. This section will explain the implementation of the application by describing both the data-model we have developed, and the different user-interfaces that are presented.

4.2.1 Android framework and programming patterns

In contrast to a free-style application architecture, the Android application architecture is framework-based application architecture. A free-style

---

[8] GoPro Hero Camera comparison - http://gopro.com/product-comparison-hd-hero2-hd-hero-cameras/

application written in Java starts from a class with a *main()* method and the developer is free to do pretty much whatever she or he wants. As opposite to this, a framework-based application is built on an existing framework, and the developer extends certain classes or implements interfaces provided by the framework to build an application; the application cannot run without the framework it was built upon. An example is Java web applications where the developer implements *Servlet*[9] interface or extends one of its subclasses.

The Android framework follows the Model-View-View-Model (MVVM) architectural pattern in how it handles relationships between Graphical User Interface (GUI) and a logic supporting the GUI. MVVM is similar to other patterns such as the Model-View-Controller (MVC) pattern. The main goal of these patterns is to clearly separate the GUI and the logic behind that handles the data model, the data processing and the communication.

The two main Android classes that are used in an application are the *View*-class (base class for all GUI-elements) and the *Activity*-class (provides logic behind the GUI). As a developer you typically subclass the *Activity*-class to handle logically separated parts of the application. In our case we have one *Activity* handling the registration of events, and another handling the viewing of events/videos. An *Activity* is active and running only when its GUI is in the foreground. As soon as another *Activity* comes in front of the current one, the current one stops running even if it was in the middle of doing something. Therefore, to handle longer running operations, such as file downloading and other network communication, the framework also provides a *Service*-class. We have extended the Service-class to handle communication with external systems. Additionally, the Android-framework provides a publish/subscribe model for inter-process communication. The *BroadcastReceiver*-class serves as the subscriber, and the *Intent*-class serves as the publisher. These are, similarly to the *Activity* and *Service*-class, extended to suit the needs application.

4.2.2 Overview

The application we built consists of two main activities; one for event registration and one for viewing previously stored events with corresponding videos. In addition we have a preference interface for changing application settings. Figure 4.4 shows an overview of the application.

---

[9] Servlets - http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/Servlets.html

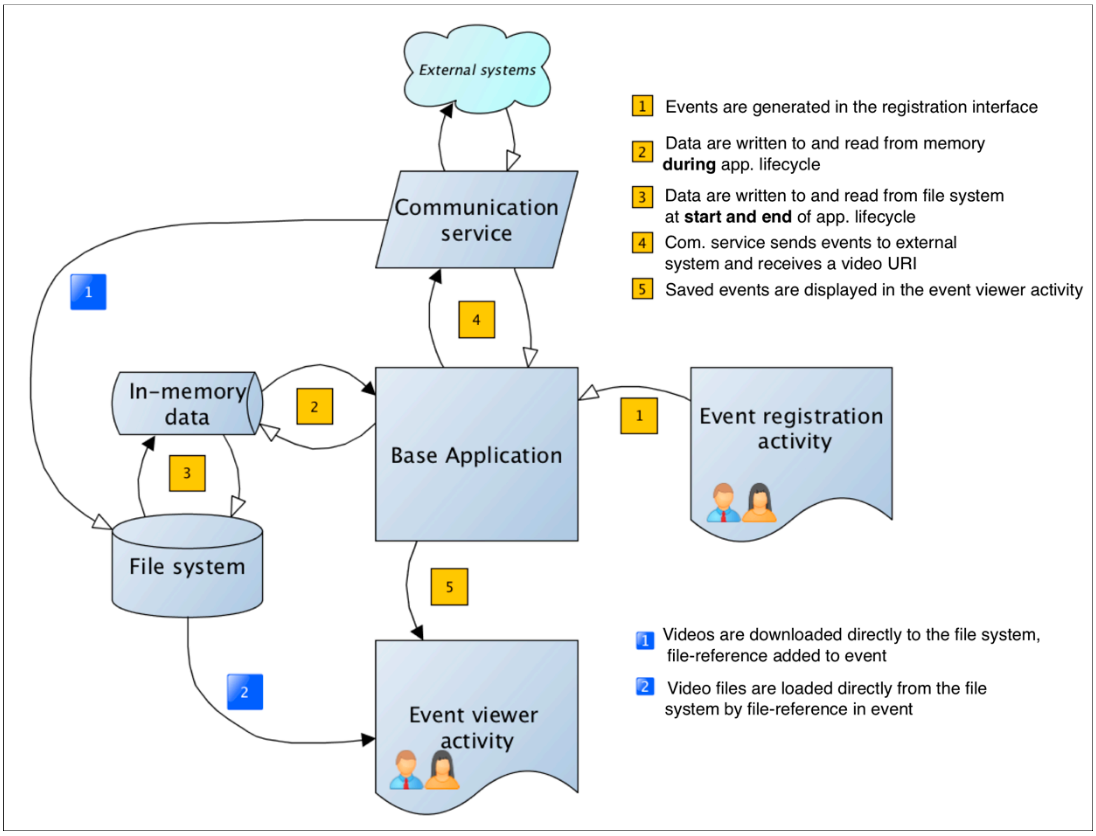

Figure 4.4: Vuvuzela Android application implementation overview

The figure illustrates how an event is handled throughout the application. The user generates events through the event registration activity (1). During the lifecycle of the application all events are kept in memory (2). When the application starts, it loads previous events from the file system into memory, and when the application shuts down all loaded data are saved to the file system (3). The communication service requests videos from the external video provider systems corresponding to the new events that are tagged (4). When the user starts the event viewer activity, the events are passed to the activity and displayed (5).

An important note is that no actual video-data is passed through the in-memory data-store that the base application handles. The videos are saved by the communication service directly to the file-system, and the event-viewer activity loads videos from the file-system. All events that have corresponding videos contain file references to those videos. We implemented it like this in order to prevent passing huge amounts of data through the inter-process communication channels.

4.2.3 Data model and storage

The internal data model of the Android application is described in Figure 4.5. It shows the different entities we have defined their properties, and how they relate to each other.

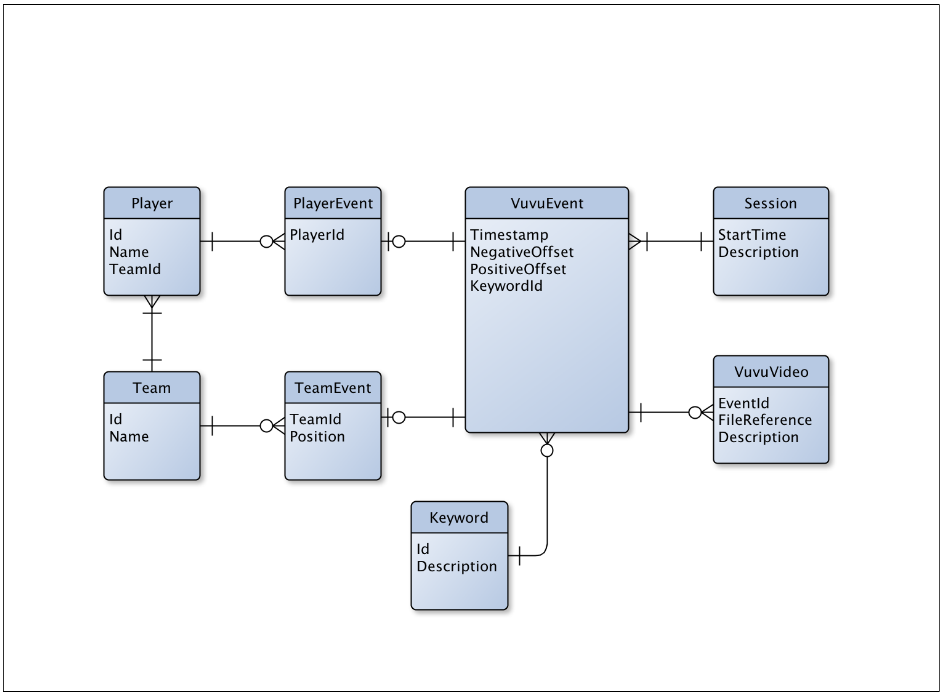

Figure 4.5: Data model in Vuvuzela Android application

We have defined an abstract *VuvuEvent*-class, which both the *PlayerEvent* and *TeamEvent*-classes inherit from. The *NegativeOffset* and *PositiveOffset* properties define how long before and after, in seconds from the timestamp, the event lasted. This is used to tell the video provider system the length of a requested video. In addition to the properties illustrated in Figure 4.5, we defined some abstract methods in the VuvuEvent-class that the inheriting classes must overload. These methods handle converting to and from string representation in formats such as JSON[10] and XML[11]. Additionally a *PlayerEvent* corresponds to a *Player*, and a *TeamEvent* corresponds to a *Team*.

The *VuvuVideo*-class contains a reference to the event, a reference to the actual video file on the device, and a description-field that the user can add to describe the video. This class is instantiated when a video has been downloaded from an external system.

The *Session*-class contains a description and a start time. The description is typically used to tell during which match the session relates to, and the start time makes it possible to sort by date and time. A session works as a single entity containing many events.

The *Player* and the *Team*-classes are similar except that a Player contains a reference to the Team she or he is part of.

For data storage on the device, we chose to save all data in files in the JSON format. We save all entity types in separate files, for instance players in a player-file, sessions in a session-file and so on. Events are saved per session. This means that every session has an event-file reference that saves the name of the file containing all the events created for that session.

4.2.4 Event registration interface

The event registration interface was designed through the design process described in Section 3.1, where several contributors made an impact on the prototyped version.

An early idea was to have buttons representing keywords that describe an event, and have the buttons correspond to a player in the interface. If the user pressed the button, an event would be instantiated and saved with the corresponding timestamp, player id and keyword. A design proposition of the event registration interface is shown in Figure 4.6. After some iterations of the design process we identified the need to focus more on the whole team in a match situation, and that an even simpler interface would be better suited.

---

[10] JavaScript Object Notation – http://www.json.org

[11] Extensible Markup Language - http://www.w3.org/XML/

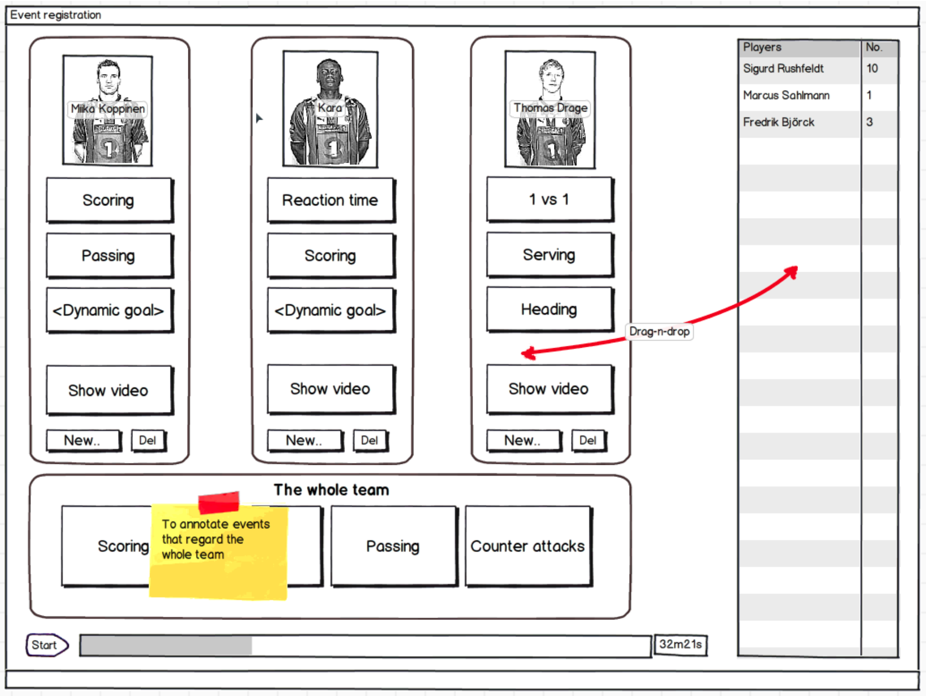

Figure 4.6: Early event registration interface design

Through the design discussions with TIL, we decided to divide the events into two main categories; events related to situations where the team is attacking, and events related to situations where the team is defending. Relevant keywords for describing events differ between the two categories, and this needed to be represented in the interface as well. Additionally, we wanted to give the user the ability to register events for players as well. This is made possible by defining up to three players as the players to be focused on during a session, and was taken into consideration when the interface was built. The version of the interface that we have used in the prototype can be seen in Figure 4.7. Below the figure is a short user scenario to explain how it works.

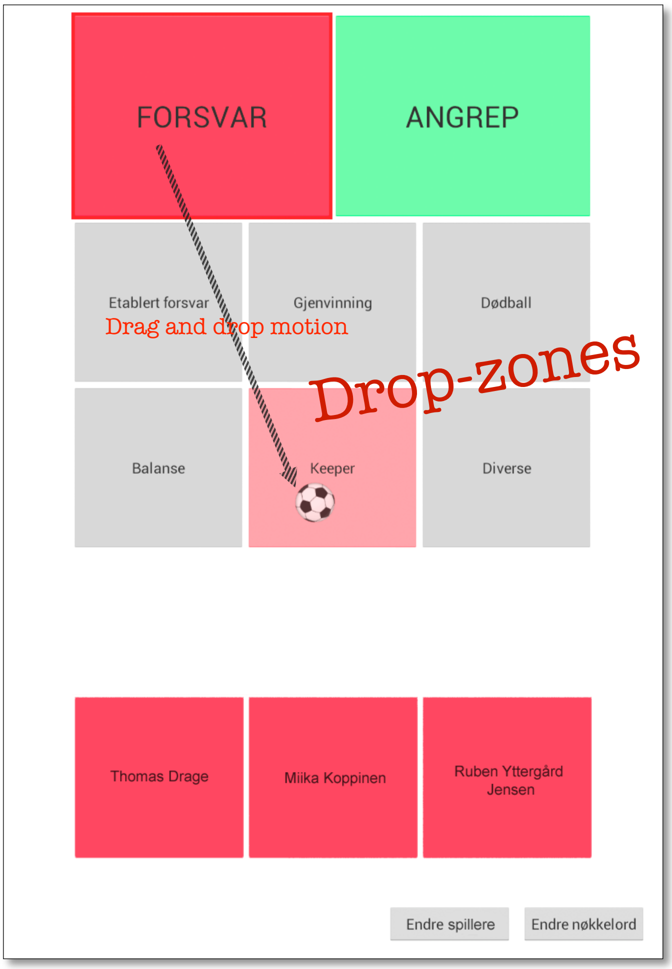

Figure 4.7: Event registration interface. Red text and arrow is not part of actual screenshot.

User scenario – generate events

The user touches Angrep (Norwegian for Attack) or Forsvar (Defence), and then keeps the finger on the screen to start a **drag-and-drop** motion. Below the two main buttons at the top we have different **drop-zones**, which correspond to keywords that the user can alter in the preference menu (accessed in the lower right of the screen). When the user drops the ball in one of the **drop-zones** a *TeamEvent* is created with the team position (Attack or Defence), keyword and timestamp. The user can also initiate a drag from either of the player-buttons in the lower part of the screen. The **drop-zones** will change corresponding to pre-defined keywords for each player. Similarly to the *TeamEvent*-instance that is created when dragging from one of the team position-buttons, a *PlayerEvent* will be created when dragging from any of the player-buttons. A **drop-zone** changes colour when the drag-cursor enters its borders, and changes back to default if the cursor leaves, or is dropped within, the zone. This helps the user to understand where the cursor is at any time, and which keyword is registered for the event. If a drag ends outside any of the drop-zones, no event is registered.

4.2.5 Event viewer interface

The Event viewer interface mainly consists of two lists; one list of columns displaying the sessions that are stored on the device and a second list of columns to display the events that correspond to the session that is selected. Figure 4.8. shows a screenshot of the interface. The list to the left (three columns) displays the sessions that are present, and the list to the right (four columns) displays the events that were tagged during the selected session.

Figure 4.8: Session and event viewer interface

This interface can be used to browse through events and sessions, as well as editing the items. To do this the user has to make a long-click motion on any of the items, which brings up a contextual menu that provides choices such as deletion and editing of the selected item. If an event-item is clicked, the user is presented a list of videos that corresponds to the event. By browsing the file system of the device it is also possible to add videos that are on the device to an event. This functionality was needed to be able to add videos that Camera system #2 generates, since they are loaded onto the device after the match.

The videos are shown through the video player that is embedded in the Android framework. Videos that relate to the selected event are displayed in a list like the sessions-list and the events-list. When a video is selected, the user can choose to either delete it or add it to a playlist. The video playlists can be accessed from a separate interface that uses the same layout. Figure 4.9 illustrates the video list interface.

Figure 4.9: The interface displaying a list of videos corresponding to an event.

4.2.6 Communication service

To handle communication with external systems, such as the two camera systems, we have developed a communication service. It runs as a background worker that does not interfere with the main GUI-thread, and takes care of both posting events through http-requests and downloads video files provided by the external system. All communication is done through the http-protocol. When the communication service receives an event for posting to a camera system it calls the *toXML*() method of the received event. The resulting XML-object will have slightly differing properties depending on which of the camera systems it posts to. Camera system #1 uses the name of the player as an identifier for pairing with ZXY-data, while Camera system #2 uses the keyword and player or team to describe the resulting video. This is further explained in the next section.

When posting to Camera system #1, a URI pointing to a video of the event is returned. This URI is put into a download queue, and the reference to the downloaded video file is added to the event (see Figure 4.4). Posting to Camera system #2 is slightly different. Instead of posting one event at the time, the whole session is posted with an XML-object containing all the events that belong to the session.

4.3 Camera systems

As explained earlier, we have two different camera systems that provide video to our prototype. System #1 is developed in cooperation with Simen Sægrov in the iAD-group at the University in Oslo. System #2 is part of the Muithu-system described in section 2.4.4, and developed by Magnus Stenhaug, a student at the University of Tromsø. The next sections provide an overview of the implementation of the two systems.

4.3.1 Static camera system with ZXY-integration (System #1)

System #1 consists of two components; one web-server component that handles requests from our android prototype, and one component that generate videos based on the data contained in the requests. A request contains a timestamp when the event took place, the wanted length of the video, and the names of the players involved. The player names are matched with their names in the zxy-database, which provides the system with the positional coordinates for the players at the time of the event.

To map the coordinates on the field to images from a camera, an open source library called OpenCV is employed. OpenCV creates a transformation matrix that can be used to convert from zxy-coordinates to pixel-coordinates in an image. To get correct transformation matrices for each camera, we have to manually identify key places on the field and map the zxy-coordinate to the pixel-coordinate of image from the camera. This process is illustrated in Figure 4.10.

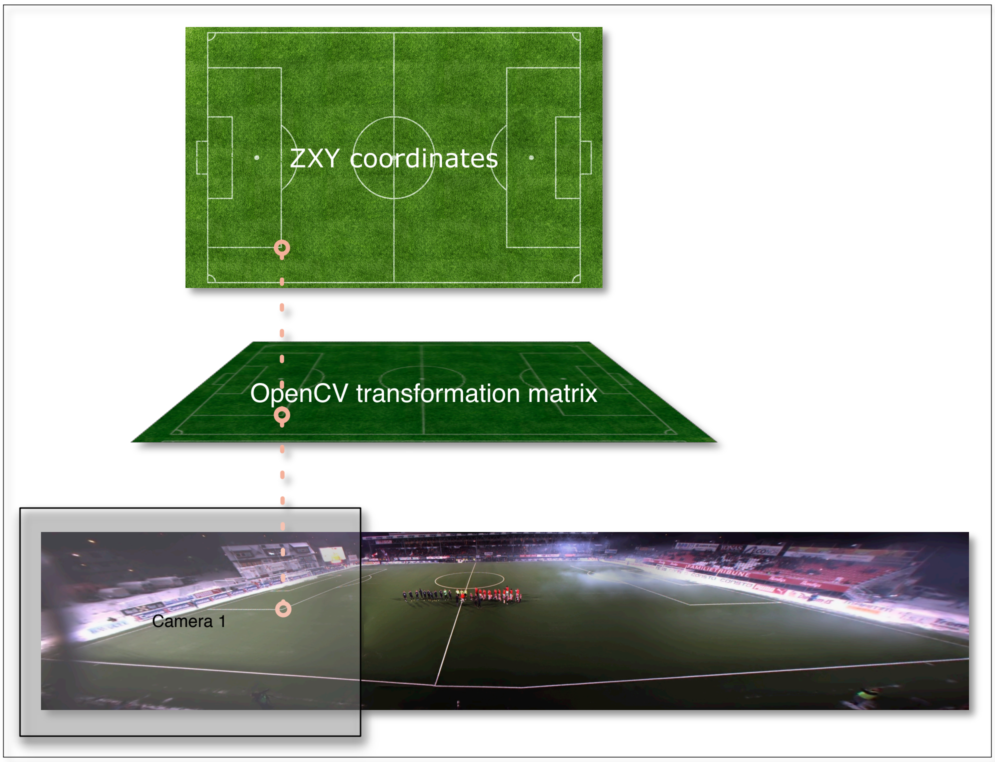

Figure 4.10: Illustration of ZXY-coordinate to pixel mapping

The system enables us to request both the optimal camera for a single player, and the optimal camera for a group of players. In the case of a single player, the system encodes a video sequence from the camera that covers the position of the player. If the player moves into the area covered by a neighbouring camera, the video sequence switches to that camera. In the case of a group of players, the system will choose the camera where the largest number of players in the group is covered. For instance, if Camera 1 covers three of the four players in requested group, the system will return the footage for Camera 1. With the provided functionality the web-server component can serve our Android application with both videos of events with a group of players (defence, offence) and events that relate to a single player in a *near-real-time* manner. We say *near-real-time* because *real-time* would be streaming the video directly, which is not part of the requirements of our system.

To handle time-synchronization the system employs a time code server that uses a method based on the well-known algorithm described [22]. In contrast to original method, the time code server synchronizes a media clock that is run on top of the system clock, since we cannot assume to have sufficient privileges to adjust system time.

4.3.2 Position flexible camera system (System #2)

System #2 handles requests from our Android application as a batch of events. The application posts all events generated in a session to the system through http-post with an XML content describing the events. An event that is posted to the system contains a timestamp and a description. The description includes the keyword for the event, and the object that was involved. The object could either be a player or a part of the team. A video sequence of 15 seconds in length backwards from the timestamp of every event is created from each of the camera feeds that were recorded during the match. This means that if there were three cameras recording the match, each event will have three corresponding videos (each 15 seconds long) potentially covering the event.

As the system includes video from several cameras placed around the field that are started separately, an important problem to solve is the synchronization of the different video feeds. This is handled by using a mobile phone to annotate the time each camera is started. The timestamp for the start-up of each camera is loaded into the system, which then synchronizes with the clock of the machine that runs the system. Time synchronization is further discussed in the evaluation of our prototype.

4.4 Summary

This chapter has explained how the Vuvuzela prototype was implemented, and how we designed the different user interfaces in the Android application. We have also given an overview of the two camera systems that have been in use on Alfheim Stadium during our tests, and how they communicate with our prototype application.

Chapter 5 Case study and experiments

This chapter explains how we have evaluated our prototype through case studies and experiments, using material from three Tippeliga matches at Alfheim Stadium.

5.1 Introduction

Case studies can bring us to an understanding of complex issues or objects by emphasizing detailed contextual analysis of a limited number of events and their relationships [23]. In spite of some criticism (for instance that a small number of cases can not establish generality of findings; or can bias the results), case studies continue to be used with success because of its applicability to real-life issues and contemporary human situations.

Because of the user-centred and real-life nature of our system, we decided that a case study of how the system could prove useful for the coaches at TIL was the best way to evaluate what we have implemented. We have conducted tests of our annotation system with match analyst Truls Jensen at TIL during matches on Alfheim Stadium, and evaluated its usefulness through interviews where we reviewed the events and corresponding videos that was generated. Additionally, we have done experiments with regards to performance and proof-of-concept of the camera system that integrates ZXY-data to video feeds (Camera system #1).

5.2 Live tests at Alfheim Stadium

To evaluate the usefulness of our prototype we have conducted tests of the system during football matches on Alfheim Stadium, the home ground of Tromsø Idrettslag. The tests were done with Truls Jensen as the user of the system.

5.2.1 The role of the test-user

Jensen's role during a match is to analyze the events of the match from an elevated point of vision, as opposed to the coaches who are situated at the coaching bench at the ground level of the field. He is positioned in the upper section of the stadium above the stalls in order to get an overview of the field. To annotate events and interesting trends he uses a piece of paper with the outline of a football field printed on it, and a pencil to draw and write on the paper. While this is what Jensen uses, it is not the only tools in use at TIL for analysis. As noted earlier they have other systems for post-match analysis. However, to identify trends and give feedback to the coaches (by internal radio contact) and players during a match, the pen-and-paper approach is what is in use at Alfheim today. Figure 5.1 is shows Jensen during a match with his current tool set.

Figure 5.1: Truls Jensen with his current analysis tools

5.2.2 Annotation of events through Vuvuzela

Our prototype was tested during two Tippeliga matches in April 2012. The first match was between TIL and Sandnes Ulf, and the second between TIL and Hønefoss. We were situated together with Jensen in the upper sections of Alfheim Stadium. He was given a short introduction to the user interface and how to annotate events before the matches. Jensen defined his own keywords for events in the offense and defence parts of the team. These keywords are shown in Table 5.1.

Keyword no.	Defence Keywords	Offense Keywords
Keyword #1	Established defence	Established offense
Keyword #2	Ball recovery	Breakdown
Keyword #3	Set piece	Set piece
Keyword #4	Team balance	Ball recovery
Keyword #5	Keeper	Goal
Keyword #6	Miscellaneous	Miscellaneous

Table 5.1: Keywords defined by Truls Jensen during case study

During the matches Jensen annotated events with the defined keywords. All annotated events were saved on the device for evaluation. It should be noted that a limited number of event keywords were used in the tests. A somewhat more extensive set is possible but not practical because too many keywords will demand more attention when using the device than advisable during a hectic match. The interface that was used during the case study is shown in the step-by-step description in Figure 5.2. An event is generated at the moment the user lifts the finger from the screen, as seen in Step 3.

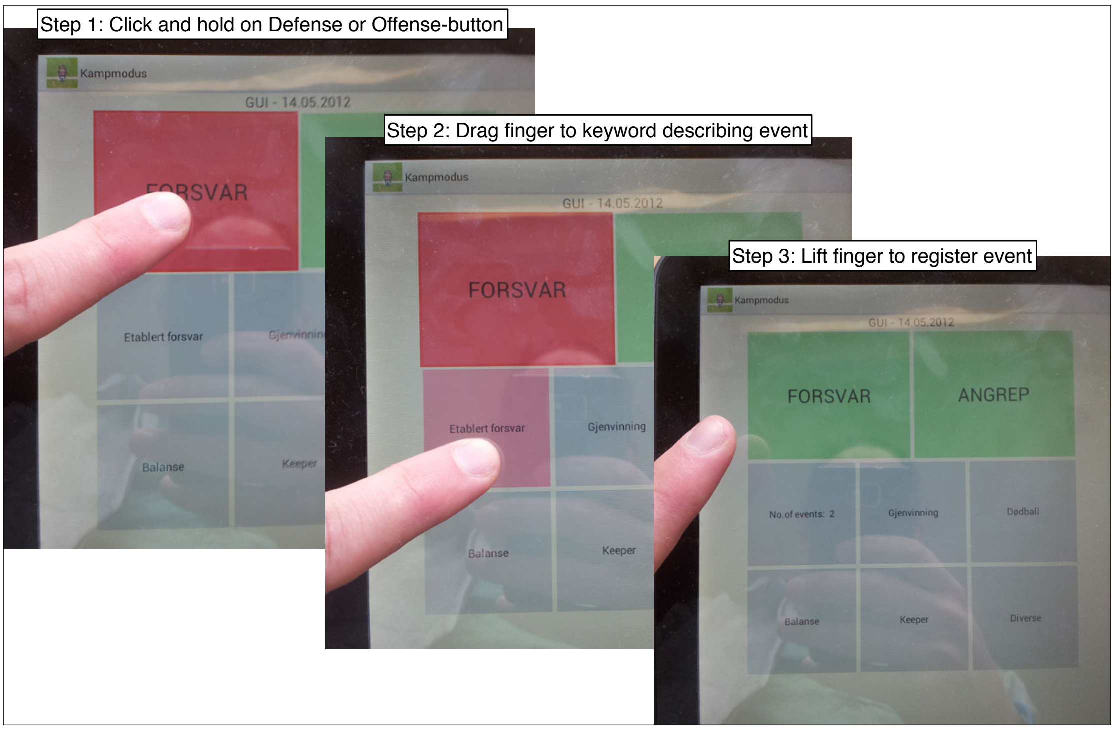

Figure 5.2: Drag and drop interface for event registration, as used in the case study

5.2.3 Video recording of the matches

To record the two matches in our case study we employed camera system #2. In addition to some cameras around the field on ground level, we mounted two cameras on an elevated platform at the opposite side of the field of the test-user's position. Figure 5.3 illustrates the position of the cameras and the test-user in relation to the field.

Figure 5.3: Positions of cameras and user during test-matches at Alfheim Stadium[12]

For evaluation purposes we only used video recorded by the two cameras illustrated in the figure, since their point of view closely simulates the point of view of the user although on the opposite side of the stadium.

After each match, the cameras were collected and the video files were transferred to a machine running the software that extracts video sequences corresponding to the annotated events, after which the resulting video sequences were copied to the tablet application for further review together with Truls Jensen.

5.3 Experiments with Camera system #1

Camera system #1 is deployed on Alfheim for recording of matches and testing integration with ZXY-data as a part of the ongoing cooperation between the iAD-group and TIL. Because of re-calibration and further development on the

[12] Overview image of Alfheim Stadium retrieved from http://maps.google.com

recording software, we were not able to use the system during the matches covered in our case studies with Truls Jensen. However, the last Tippeliga match of the previous season (against Rosenborg) was recorded using this system, and by integrating the video data with ZXY-positional data from the same match we were able to test how this system performs in cooperation with our prototype.

We have tested this system in conjunction with our annotation interface by simulating a live match environment, using the actual recordings from the Rosenborg match from November 2011. The simulation was performed by manually synchronizing the time when we started the viewing of the match with the pre-recorded data that was captured by our cameras, and the ZXY-system, during the match, and then annotated the simulated real-time match with the Vuvuzela prototype. This process is illustrated in Figure 5.4.

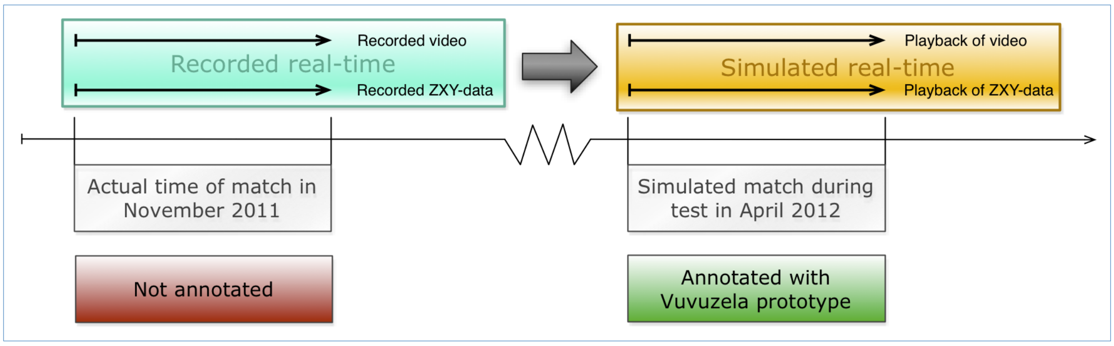

Figure 5.4: Simulation of real-time match annotation

The Android application prototype was calibrated to connect to the server running the video provider system through a Wi-Fi network, and to post each event when it was annotated. The annotation was done by us, and because of our limited knowledge of football analysis we did not try to actually analyze the events of the match. However, we annotated notable events to see how well the system performed regarding the time it took to encode and provide the requested videos. We also wanted to see if the system could provide precise video footage showing the players involved in the event. These tests were done in cooperation with Simen Sægrov from UiO who has developed the part of the system that integrates ZXY-data with the recorded video.

5.4 Summary

This chapter starts with a brief, general discussion on case studies and continues with describing cases to test the usefulness of our prototype when applied to recordings from three Tippeliga matches at Alfheim Stadium. First, match analyst Truls Jensen annotated two 2012 matches on the Vuvuzela mobile device in real time, using event keywords defined by himself. Second, we used recordings with Camera system # 1 and ZXY data from a 2011 match to simulate a case to test our annotation system. The experiment was done by playing back the recorded match and then annotating the match in simulated real-time with the Vuvuzela prototype.

Chapter 6 Evaluation

In this chapter we will discuss both the case study we conducted with TIL and evaluate the data we gathered from the experiment done with the static camera system.

6.1 Case study evaluation

To evaluate our prototype through the conducted case study, we arranged an interview-session with Truls Jensen. As noted earlier, Jensen tested the system as a user during the matches we covered in the cases described in Chapter 5. As a basis for the interview we asked him to go systematically through the videos of the events he annotated during the matches. By examining how many of the videos that were precise in describing the annotated event, we obtained a metric for how accurate the user managed to annotate events. Then we had detailed and thorough discussions with Jensen regarding the usefulness of the prototype, and how it could be applied to identify emerging trends during a match. Finally, we did a comparison of our system with Interplay-Sports, using recordings from the first match of the case study.

6.1.1 Video relevance

During the first match, TIL vs. Sandnes Ulf, 48 events were annotated and during the second match, TIL vs. Hønefoss, 29 events were annotated. The camera systems used for the case studies were configured to provide videos that end at the time when the event was tagged, and start 15 seconds prior to the timestamp. This way, the user must wait until an interesting happening has ended before tagging it. Since the first match was Jensen's initial experience with the actual prototype, a portion of the annotated events was tagged at the wrong time, resulting in a video that only covered the event partially. This effect is illustrated in Figure 6.1.

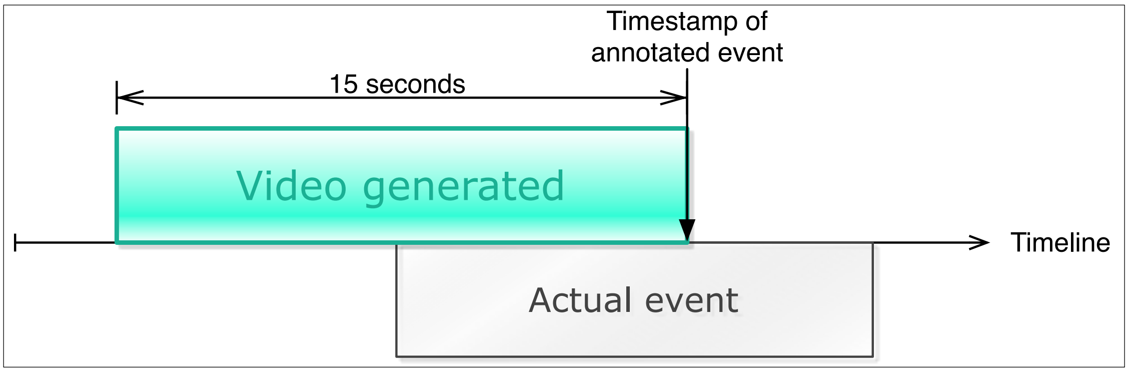

Figure 6.1: Mismatch between an annotated event and the actual event

The reason for configuring the video provider system like this is that an event often looks interesting at first glance but the outcome might prove that it is not worth focusing on. For instance, a set piece corner kick often generates an interesting situation, but if the player that takes the corner kicks it directly out of bounds there is no reason to annotate the event. This differs from the way many other analysis systems handles annotation, where every corner may be annotated and used for statistical purposes, regardless of the outcome.

Since we used a two-camera setup, the resulting set of video files must be the double of the number of annotated events, which are 96 files from the first match and 58 from the second match. With the exception of the events where the action moves between the areas covered by each camera, most of the tagged events were only covered by one of the cameras. Figure 6.2 illustrates this by showing a snapshot of an event that was tagged during the match.

Figure 6.2: A snapshot from the two cameras showing the same event

The event was annotated as "Offense Breakdown", which in layman's terms means that the team recovered the ball from the opposition and went quickly into an offensive move to try to catch the other team off guard. This is also called a counter-attack. More specifically, we see one of the attacking players on TIL bringing the ball towards the opposition's goal (marked with a red ring on the snapshot from the north-facing camera), while their defence is off balance. The outcome of the situation was that the player with the ball made a shot attempt that ended off-target (outside the goal posts). According to Jensen, this situation is correctly tagged as interesting because the player should have passed the ball to the teammate on his left instead of finishing. He also argued that this would be an effective video to show to the involved players, because it clearly shows the situation. What is also apparent in Figure 6.2, is that the southward facing video did not cover the situation, and thus did not need to be saved.

We went through all the videos with Jensen and identified the ones that were either falsely tagged as interesting, or did not cover the actual situation because of the camera position, or were tagged at the wrong time, resulting in a video that is skewed from the actual event as exemplified in Figure 6.1. Figure 6.3 shows the distribution of the videos in terms of relevance, from each match. The total number of annotated events for the first match was 48, while the total number for the second match was 29. As noted above, the number of video files must be the double because of the double cameras.

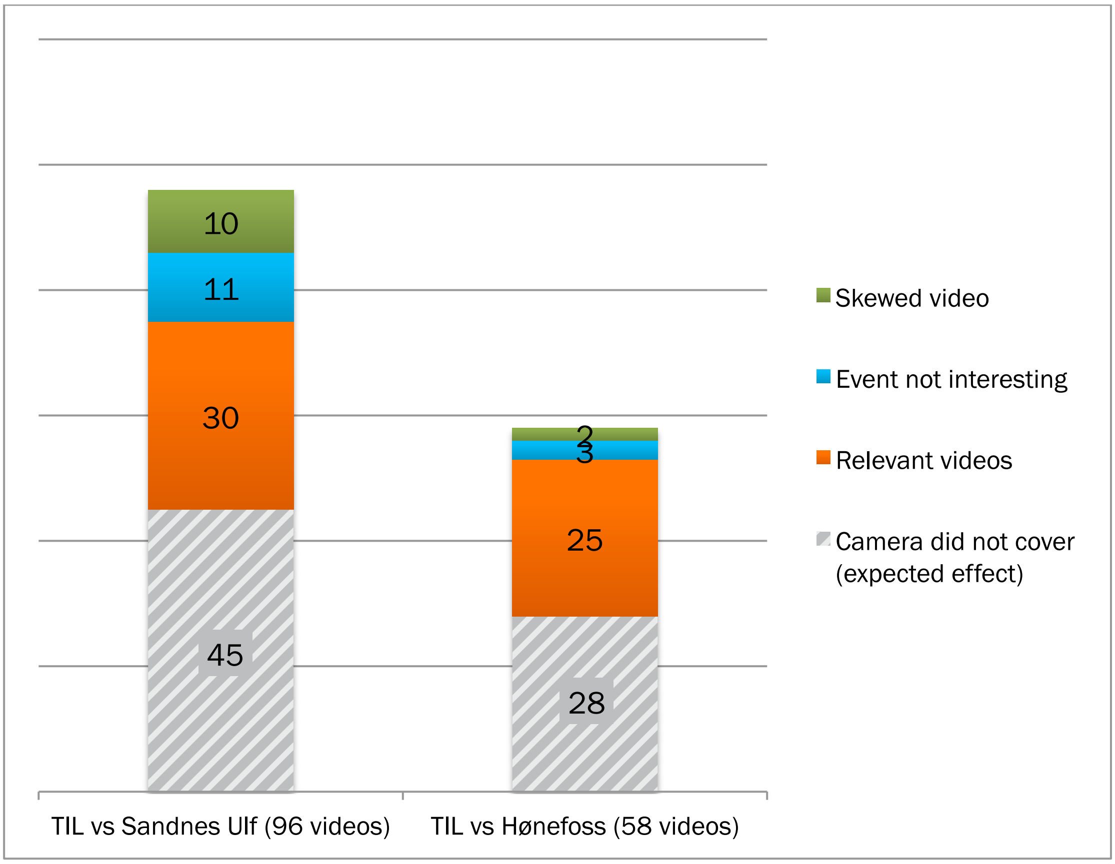

Figure 6.3: Distribution of videos per match in two test cases from April 2012. The first match to the left.

The figure shows that the level of faulty use (the categories 'Event not interesting' and 'Skewed video'), of the annotation interface was reduced from 21 in the first match to 5 in the second. This is an indication that the user of the system will need some training in order to reduce the number of irrelevant videos that are generated. We also see that almost half of the generated videos cover a part of the field where the event did not take place. This was not a surprise, but rather an expected outcome of having the cameras covering a half of the field each. In the few cases where both cameras cover the event, the event took place in the middle of the field where an overlap between the camera zones is present. In spite of the reduction of videos from the first to the second match related to the experience of the user, the data we have from these two cases indicate that the number of events will differ between matches. This is not unexpected, because all experience show that football matches (and, for that matter, sessions as defined here) vary much in the way the play develops. Therefore a study of several more matches has to be conducted to say more about the number of relevant videos that will result from employing our system.

6.1.2 Identifying emerging trends

Table 6.1 shows the distribution of keywords in the annotated events of the match versus Sandnes Ulf. Because of the relatively high number of events that were annotated as "Defence Miscellaneous", we examined the videos corresponding to the events in our session with Jensen.

Defence keywords	# annotations	Offense keywords	# annotations
Established defence	3	Established Attack	5
Ball recovery	0	Breakdown	7
Set piece	7	Set piece	7
Team balance	1	Ball recovery	0
Keeper	2	Goal	3
Miscellaneous	11	Miscellaneous	2

Table 6.1: Distribution of keywords in the first match of the case study

Eight of the eleven events were related to a single player from the opposing team. Jensen continuously pointed out during the match that the player in question was causing a problem for the TIL-defence, and annotated those events with the *miscellaneous* keyword. According to Jensen most matches has some kind of situation that occurs several times that is impossible to define prior to the match in terms of a keyword. When looking at the events from the second match, we identified a similar trend. However the trend in the second match related to a player on TIL that kept doing the same mistake. In both matches Jensen wrote down these trends to use in his half-time summary when talking to the coaches and players. We call these situations *emerging trends.* Through our talks with Jensen, he specified that a system that enable showing these emerging trends through video feedback would greatly improve upon the current situation of pen and paper.

6.1.3 Comparison to Interplay-Sports

During the first match of our case study TIL employed live annotation through Interplay-Sports. From an interview with Svein-Morten Johansen, the analyst at TIL who handles the system, we collected data on the time spent doing an analysis of the match. To perform live annotation with Interplay-Sports, the user is situated on the platform where the TV-production cameras are positioned. He connects a laptop computer to the camera that records an overview picture of the field, and uses the footage from that camera as a basis for the analysis. Prior to the match, the players chosen to represent the team are entered into the system. The set-up of the program and the technical part of configuring the camera connection takes one hour. Figure 6.4 shows Johansen's position during a match.

Figure 6.4: Position of Svein-Morten Johansen when live annotating through Interplay-Sports

After the match, Johansen spends around two hours to ready a match-presentation to show to the coaches and the players. He argues that after spending one-and-a-half years as a user of the system, there are still parts of the system he is not using to its full extent. Interplay-Sports is not directly comparable to our system, and the analysis made possible by it has a different focus than what we propose. However, in contrast to our system, Interplay-Sports is not capable of giving precise video feedback during or closely after the end of a match.

6.2 Experiments with Camera system #1

As described in section 5.3, we simulated a live match environment to test the performance of camera system #1. We annotated events through the mobile application, and the server running the video system generated videos based on the requests from the application. The events that we annotated was notable events that we as laymen could understand, such as a goal and a big scoring opportunity. To measure performance we recorded data on the time taken to encode each video, and the resulting size of the video files, which impacts the time it takes to transfer the video to the device through a wireless network.

Similarly to camera system #2, this system was also configured to generate videos backwards from the timestamp of the event (see figure 6.1).

6.2.1 Following a single player

To test if the system could follow a single player, we annotated the first goal of the match. The goal was scored by Sigurd Rushfeldt, a striker at TIL. The system was configured to generate two videos; one that zoomed in on the involved player and one that did not. Figure 6.5 shows a snapshot of the resulting videos. The red arrows point to Rushfeldt's position, and are added for clarification. The situation we see in the images is a few seconds prior to the scoring, which shows that the video is precise in displaying the event.

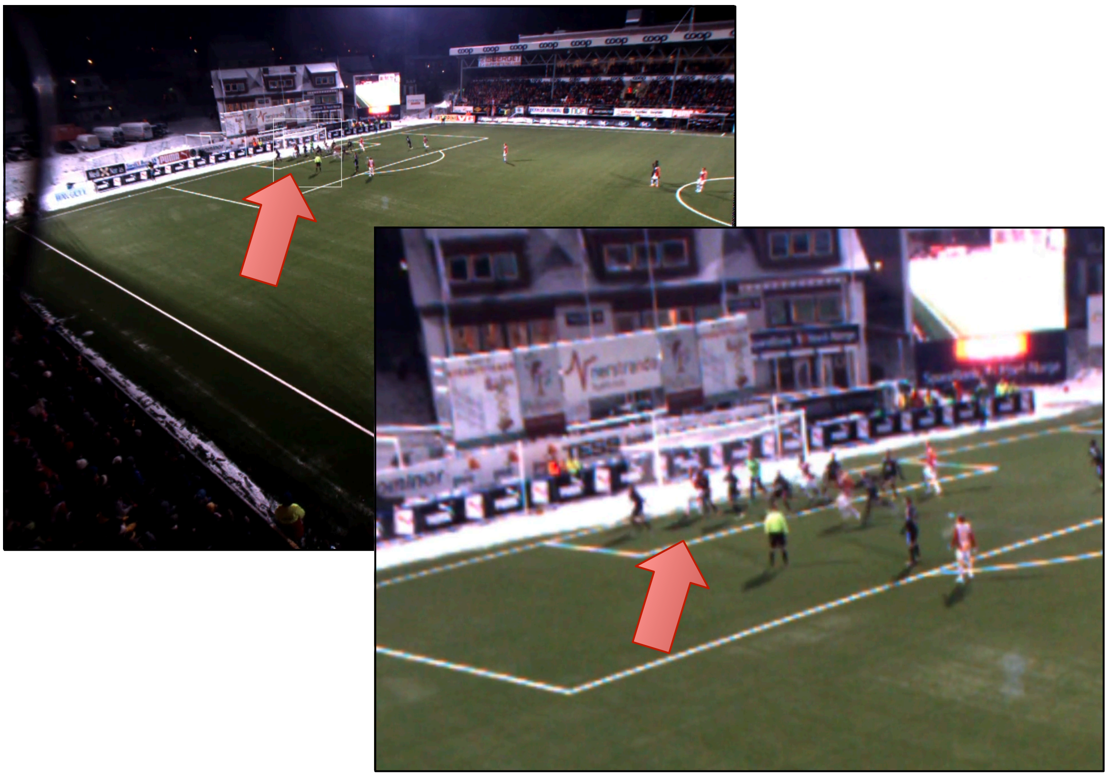

Figure 6.5: Tracking a single player in an event

6.2.2 Following multiple players

To test the system with a request involving multiple players, we annotated an established offense move by TIL. Four players were included in the request generated by our mobile application. The video generated by the system embeds white squares in the video stream to show the position of the players who were involved. This is used for testing and clarification purposes only, and would be removed in a real match situation. Figure 6.6 shows a snapshot of the resulting video.

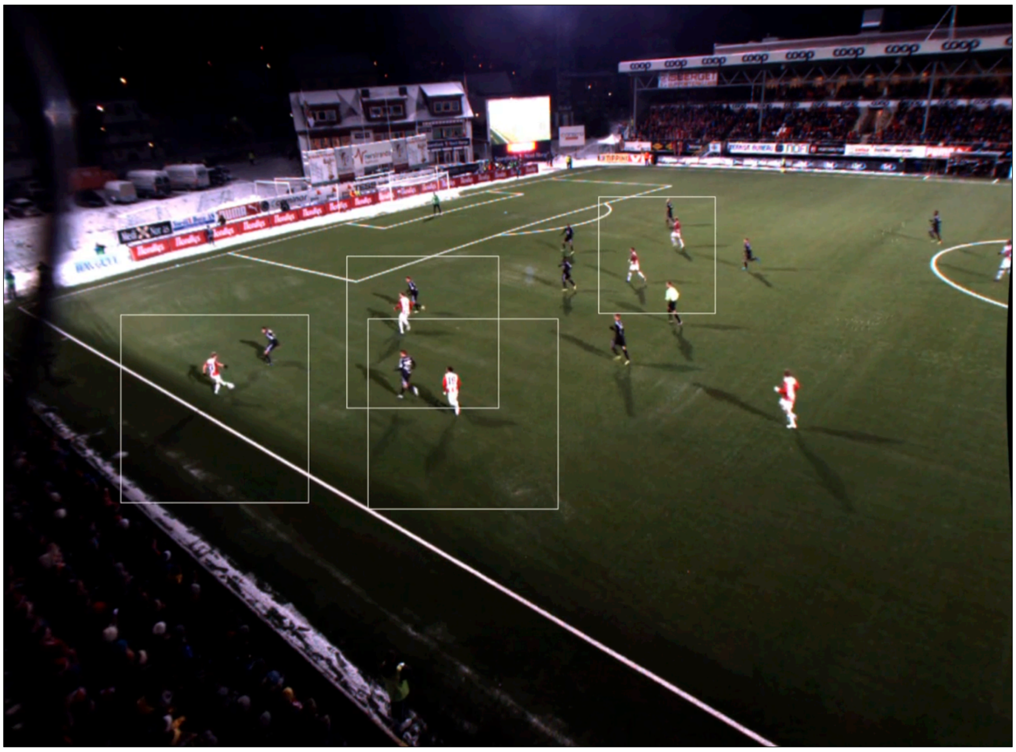

Figure 6.6: Tracking multiple players involved in an event

6.3.3 Encoding performance and file size

The videos we tested with were encoded using FFmpeg's h264-encoder[13]. The encoder is run with parameters regarding the speed of the process, which affects the size of the video files. Typically a file that is encoded with a "slow" setting will be smaller than a file that is encoded with a "fast" setting. We ran tests with a

[13] FFmpeg - http://ffmpeg.org/

preset called "slow" and a preset called "ultrafast", and compared the differences. The tests were run on both the zoomed in video from section 6.3.1, and the video from 6.3.2 that is not zoomed.

The system was run on a machine with a quad-core Intel CPU (Core i7 2600K)[14] that uses Intel's HyperThreading technology. Additionally the machine was fitted with 8GB RAM.

Table 6.2 displays the results of the tests we ran. The length of each video clip was 20 seconds.

	Zoomed video		Non-zoomed video	
	Secs. to encode video	Size of file	Secs. to encode video	Size of file
Ultrafast setting	1.67	14,3 MB	1.37	9,5 MB
Slow setting	25	11,9 MB	12.77	5,6 MB

Table 6.2: Time taken to encode videos, and the resulting file size

The results were that by using the "ultrafast" setting, the system can provide our tablet application with a requested video within two seconds, while the slower encoding process uses a lot more time. The sizes of the files differ from the non-zoomed video to the zoomed video. A reason for this is probably the fact that the non-zoomed video is static, which means that most of the pixels in an image will not change from one frame to the next. This in turn means that the encoding process can compress the images more effectively than in the case of the zoomed video, where a larger number of the pixels will change from frame to frame. It is important to note that these numbers does not include the time it takes for a video file to be transferred to the tablet via a wireless network. The router used for our experiments was running the 801.11G wireless standard, which has an average throughput of around 20Mb/s. Converted to bytes this is 2,5 MB/s, which means that a file of 15MB takes 6 seconds to transfer to the tablet.

---

[14] Intel Core i7 2600K CPU - http://ark.intel.com/products/52214

However, the important fact we read from the data, is that if the system uses around eight seconds to both encode (with the ultrafast setting) and transfer a video to the device, it is sufficient to serve its purpose in our context. We argue this because the videos are downloaded in the background while the user continues to annotate events, and because a playback of a given event is not expected immediately after the event has been annotated.

As noted earlier, it is impossible to know exactly how many events will be annotated during a match. However, we see from our case study that an upper limit could probably be set at around 50 events. This estimate is quite generous, as both matches from the case study generated less than 30 events. The raw video data generated by the each camera in our static camera system is around 500GB per match, which means that four cameras would generate roughly 2TB of data per ninety minutes. If we take the largest generated video from our system (Ultrafast setting, zoomed video) at 15MB times 50 events, we end up with a data size of about 750MB per match. While this is not a very precise prediction, it tells us that our system can reduce the amount of video data stored per match in a significant way.

6.3 Summary

This chapter has described the results of our case study with Truls Jensen, and our experiments with the static camera system (System #1). We argue that while the prototype used in the case study could not show videos in the half-time break, our system could potentially do that with videos provided through the static camera system. We do not propose our system as an alternative to their present analysis software, but as a supplement. Our most important finding is that through the case study and the interviews we have had with TIL, we show that the prototype we have built is a tool that could provide precise video feedback that is not available today.

Chapter 7 Conclusion

This chapter will summarize our achievements, conclude our work, and outline possible future work.

7.1 Achievements

This thesis describes and evaluates a system for live annotation of football matches, using a mobile tablet device and differently configured video systems. The problem definition we stated in Section 1.1 follows below:

This thesis shall build and evaluate a system for live notational analysis and video feedback of sporting events, with football as the specific domain. The focus will be on implementing a system that provides an annotation interface for registering events and gives precise and useful feedback through video corresponding to the annotated events. Another property of the system should be the ability to reduce the amount of video data that needs to be stored and transferred through use. The system will be deployed at Alfheim Stadium, the home ground of the Norwegian Premier League (Tippeligaen) football team Tromsø Idrettslag (TIL). The implemented prototype will subsequently be tested and evaluated by coaches from TIL.

We have proposed an architecture that uses three main components; an annotation- and a video playback component running on a mobile device, and a video recording component that provides precise video sequences to the same device. Our system was designed in close co-operation with the coaching team at TIL to ensure its usefulness. Because the users of the system are an important part of our thesis, we have included the coaches in both the design and the evaluation process. The expertise of the user plays an important role in defining the precision of our annotation system. We described the process of event annotation as a way of filtering out uninteresting events, thereby reducing the amount of stored and transferred video data, and enabling focus on events that can help improve the precision of the feedback.

To evaluate the system we have conducted a case study with the main player developer at TIL (Truls Jensen), who plays an important role in analyzing the performance of the team during matches. Both through usage of our *Vuvuzela*-prototype and through interview sessions where we examined the videos of the annotated events, we have shown that our system is well suited to handle the problem of giving precise feedback through video footage. Additionally we have examined the performance of the static camera system deployed at Alfheim Stadium to evaluate how well it can provide precise videos of events annotated during a match. We show that by generating relevant, and short, video sequences of interesting events, we can improve the precision of the feedback given to the team, compared to the pen-and-paper approach used by today.

The system we have built and evaluated is not proposed as an alternative to the system (Interplay-Sports) TIL uses for in-depth, post-match, analysis, but rather a supplement that can assist the coaches in identifying interesting events and trends during a match.

7.2 Related work

Our thesis is part of an ongoing project with TIL. One of the camera systems we have used in the evaluation process is part of the *Muithu*-system that is developed in parallel to our system by researchers in the iAD-group at the University of Tromsø; while the other camera system is developed in cooperation with our colleagues in the iAD-group at the University of Oslo. Both groups continue their research in the overarching project, and our thesis is a contribution to this larger body of work.

7.3 Concluding remarks

Our focus was to build a prototype that could evaluate the usefulness of the system architecture that we have proposed. We wanted to prove that it is possible to develop a system that can improve the current approach to live match analysis used by football clubs such as TIL. This is accomplished through the work done in cooperation with TIL, and the evaluation sessions we conducted with the involved coaches. We show that our approach provide precise, video-based feedback to both players and coaches.

7.4 Future work

The implementation of the *Vuvuzela*-system is not complete. While we show that our system can employ video from different systems, we have not been able to test a camera system that provides video during a live match because the infrastructure that is required is not yet fully built out at Alfheim. The test match we ran with Camera system #1 was simulated real-time, as described in our evaluation. To be able to do it in a live match situation, a network infrastructure will have to be deployed on the stadium to ensure that the system can transfer data between the mobile device and the video provider component.

References

[1] M. Hughes and I. M. Franks, Notational Analysis of Sport: Systems for Better Coaching and Performance in Sport. (2nd Edition), Routledge, 2004.

[2] I. M. Franks and G. Miller, "Eyewitness testimony in sport". *Journal of Sport Behaviour*, vol. 9, pp. 39-45, 1986.

[3] N. James, "Notational analysis in soccer: past, present and future," *International Journal of Performance Analysis in Sport*, vol. 6, pp. 67-81, 2006.

[4] E. Olsen and Ø. Larsen, "Use of match analysis by coaches," in *Science and Football III*, T. Reilly, J. Bangsbo and M. Hughes, Eds, E & FN Spon, London, pp. 209-220, 1997.

[5] A. Blaze, G. Atkinson, C. Harwood and A. Cale, "Prevalence and perceptions of performance analysis in the English Premier Association Football League," in *Performance Analysis in Sport V*, M. Hughes, Ed., Centre of Performance Analysis, University of Wales Institute, Cardiff, pp. 79-83, 2004.

[6] R. Groom and C. Cushion, "Coaches perceptions of the use of video analysis: A case study," *Insight*, vol. 7, pp. 56-58, 2004.

[7] D. E. Comer, D. Gries, M. C. Mulder, A. Tucker, A. J.Turner and P. R. Young, "Computing as a discipline," Commun.ACM, vol. 32, pp. 9{23, January 1989. [Online]. Available:http://doi.acm.org/10.1145/63238.63239

[8] D. Johansen, H. Johansen, T. Aarot, J. Hurley, A. Kvalnes, C. Gurrin, S. Zav, B. Olstad, E. Aaberg, T. Endestad, H. Riiser,C. Griwidz and P. Halvorsen, "DAVVI: a prototype for the next generation multimedia entertainment platform," in Proceedings of the seventeenth Association for Computation Machinery ACM international conference on Multimedia,ser. MM '09. ACM, 2009, pp. 989{990. [Online]. Available: http://doi.acm.org/10.1145/1631272.1631482

[9] E. Trucco and K. Plakas. "Video tracking: A concise survey," *IEEE Journal of Oceanic Engineering*, vol. 31(2), pp. 520 –529, 2006.

[10] C. Ben-Abdelkader, R. Cutler and L. Davis, "Person identification using automatic height and stride estimation," *Proceedings,16th International Conference on Pattern Recognition*, vol. 4, pp. 377 – 380, 2002.

[11] M. Everingham and A.Zisserman, "Automated person identification in video," in *Image and Video Retrieval*, P. Enser, Y. Kompatsiaris, N. O'Connor, A. Smeaton and A. Smeulders, Eds, Volume 3115 of Lecture Notes in Computer Science, Springer, Berlin and Heidelberg, pp. 1961–1961, 2004.

[12] K. Choi, B. Park, S. Lee and Y. Seo, "Tracking the ball and players from multiple football videos," *International Journal of Information Acquisition*, vol. 16:27, pp.1-8, 2006.

[13] BBC site: http://www.bbc.co.uk/sport/0/football/17228500

[14] ZXY Sport Tracking, http://www.zxy.no

[15] Ø. Holmstad, "Real-time annotations of video streams using staged processing," M.Sc.-thesis, University of Tromsø, Department of Computer Science, 59 pp., 2011.

[16] V. Di Salvo, A. Collins, B. McNeill and M. Cardinale, "Validation of Prozone®: A new video-based performance analysis system," *International Journal of Performance Analysis in Sport*, vol. 6 (1), pp. 108-119, 2006.

[17] Interplay Sports. The ultimate and mobile Video Analyze and Scouting software. http://www.interplay-sports.com

[18] D. Johansen, M. Stenhaug, R. B. Hansen, A. Christensen and P. M. Høgmo, "Muithu: Smaller footprint, but larger imprint?," Manuscript, Department of Computer Science, University of Tromsø.

[19] C. Abras, D. Maloney-Krichmar and J. Preece, "User-Centered Design," in. *Encyclopedia of Human-Computer Interaction*, W. Bainbridge, Ed., Thousand Oaks: Sage Publications, Draft, 2004, 14 pp., (in press).

[20] J. Preece, Y.Rogers and H. Sharp, *Interaction design: Beyond human-computer interaction*. John Wiley & Sons, New York, 2002.

[21] D.W. Swanepoel, J.W. Hall III and D. Koekemoer, "Vuvuzela- good for your team, bad for your ears," *South African Medical Journal*, vol. 100 (4), pp. 99-100, 2010.

[22] R. Gusella and S. Zatti, "The accuracy of the clock synchronization achieved by TEMPO in Berkeley UNIX 4.3BSD," *IEEE Transactions on Software Engineering*, vol. 15 (7), pp. 847–853, 1989.

[23] S. Soy, "The case study as a research method," Unpublished paper, University of Texas at Austin, 1997. Available: http://www.gslis.utexas.edu/~ssoy/usesusers/l391d1b.htm, updated 2006.

Appendix A

CD-ROM

The included CD-ROM contains the source code for the mobile Android application and sample videos from both camera systems.