Multi-sensor human action recognition with particular application to tennis event-based indexing

Damien Connaghan

School of Electronic Engineering

Doctor of Philosophy

Supervisor

Prof. Noel E. O'Connor

September, 2012

Acknowledgements

There are a number of people to whom I am greatly indebted for their help, inspiration and patience. First of all I would like to thank my supervisor, Prof. Noel E. O'Connor, for his guidance and advice over the course of my PhD.

Special thanks to Clarity and Science Foundation Ireland, who provided the funding for this research. I would also like to express my gratitude to my colleagues who I collaborated with and especially to Ciarán Ó Conaire, Philip Kelly and Alan Smeaton. I would also like to thank everyone else in Clarity and Tyndall whom I've worked with and sought advice from over the years. Finally, to my Fiancée, Debra Heeney and my family for their support.

Abstract

The ability to automatically classify human actions and activities using visual sensors or by analysing body worn sensor data has been an active research area for many years. Only recently with advancements in both fields and the ubiquitous nature of low cost sensors in our everyday lives has automatic human action recognition become a reality. While traditional sports coaching systems rely on manual indexing of events from a single modality, such as visual or inertial sensors, this thesis investigates the possibility of capturing and automatically indexing events from multimodal sensor streams. In this work, we detail a novel approach to infer human actions by fusing multimodal sensors to improve recognition accuracy. State of the art visual action recognition approaches are also investigated. Firstly we apply these action recognition detectors to basic human actions in a non-sporting context. We then perform action recognition to infer tennis events in a tennis court instrumented with cameras and inertial sensing infrastructure. The system proposed in this thesis can use either visual or inertial sensors to automatically recognise the main tennis events during play. A complete event retrieval system is also presented to allow coaches to build advanced queries, which existing sports coaching solutions cannot facilitate, without an inordinate amount of manual indexing. The event retrieval interface is evaluated against a leading commercial sports coaching tool in terms of both usability and efficiency.

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of PhD is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed:

ID No.:

Date:

List of Publications Journal

- **D Connaghan**, K Moran and N. E. O'Connor. An Automatic Visual Analysis System for Tennis. In the Journal of Sports Engineering and Technology, (in press).

Peer reviewed international conference/workshop

- **D Connaghan** and N. E. O'Connor. Toward Next Generation Coaching Tools for Court Based Racquet Sports. In ACM Multimedia, Nara, Japan, 29th October - 2nd November 2012.

- **D Connaghan**, M Gaffney, P Kelly, M Walsh, C O'Mathuna and N. E. O'Connor. Multi-sensor classification of tennis strokes. In IEEE Sensors, Limerick, Ireland, October 2011.

- **D Connaghan**, P Kelly and N. E. O'Connor. Game, shot and match: Event-based indexing of tennis. In 9th International Workshop on Content-Based Multimedia Indexing, Madrid, Spain, June 2011.

- **D Connaghan**, S Hughes, G May, P Kelly, C Ó Conaire, D O'Gorman, A Smeaton, N Moyna and N. E. O'Connor. A sensing platform for physiological and contextual feedback to tennis athletes. In BSN 2009 - Body Sensor Networks, Berkeley, U.S.A., June 2009.

- C Ó Conaire, **D Connaghan**, P Kelly and N. E. O'Connor. Combining inertial and visual sensing for human action recognition in tennis. In ARTEMIS 2010 - 1st ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams, Florence, Italy, October 2010.

- L Conroy, C Ó Conaire, S Coyle, G Healy, P Kelly, **D Connaghan**, N.E. O'Connor, A Smeaton, B Caulfield, P Nixon. TennisSense: a

multi-sensory approach to performance analysis in tennis. In 27th International Society of Biomechanics in Sports Conference, Limerick, Ireland, August 2009.

- C Ó Conaire, **D Connaghan**, P Kelly, N.E. O'Connor. TennisSense: a platform for extracting semantic information from multi-camera tennis data. In DSP 2009 - 16th International Conference on Digital Signal Processing, Santorini, Greece, July 2009.

Peer reviewed national conference

- **D Connaghan** and N. E. O'Connor. An intuitive user interface for visual sports coaching. In iHCI 2010 - 4th Irish Human Computer Interaction Conference, Dublin, Ireland, September 2010.

- **D Connaghan**, P Kelly and N. E. O'Connor. Recognition of tennis strokes using key postures. In ISSC 2010 - 21st IET Irish Signals and Systems Conference, Cork, Ireland, June 2010.

Contents

List of Figures

Part I

Background

Chapter 1

Introduction

Recent advances in computer vision and human motion detection offer great potential to many fields today, such as military operations, assisted living applications, medical applications, sport and leisure applications to name but a few. As mass production and more efficient approaches drive down the cost of visual and inertial sensors, the amount of data available from such sensory channels increases. The potential to fuse data from multiple sensors such as low cost cameras and inertial sensors attached to the human body potentially allows engineers to infer new knowledge not available from any single source.

Tennis is one of the most popular court based racquet sports in the world because of the relative simplicity of the rules and the small amount of equipment needed. A major aim of tennis coaching is to provide feedback to athletes to improve performance. While there are many aspects of performance that can be enhanced (e.g. physiology, biomechanics, psychology), recent sub-discipline of sport science have emerged (Performance Analysis and notational analysis) which aim to objectively record performance so that key elements of that performance can be quantified in a valid and consistent manner [79] [78]. An extremely common method is to video record a (ten-

nis) match and identify all of the key events (strokes), e.g. [123]; [80]; [13]. Subsequently, the coach would review this information to perhaps: (i) quantify the patterns of play, and/or (ii) identify if positive/negative outcomes are associated with a particular technique or tactic. The video recordings themselves do not necessarily directly quantify aspects of performance (e.g. measure technique or tactic) they simply provide the coach with an accurate and objective record of events, in comparison to self-recall which is inaccurate and biased [78] [55]. The coach reviews the recordings to use their experience and expert knowledge to infer technical, tactical or mental strengths and weaknesses related to performance.

While there are a variety of uses of such video-based playback systems, a central requirement for them is the identification and indexing of key events. To date this has invariably been completed through a manual process, where each action/event in the recording is tagged by the user. This however is a very time consuming process. A solution to this would be the production on an automated system that could record tennis matches and automatically index the match into key tennis events. Coaches could then review and quantify instances of indexed events as a visual coaching aid. To the best of our knowledge only one system (Hawke-Eye Coaching) has been developed which automatically indexes events. However a major limitation of this system is that it cannot index specific strokes played. Automatic indexing has not been achieved in any other sport to the best of our knowledge. The main technological advancement has been in verification of referee decisions, which has been very popular in professional tournaments. In assessing how best to present information to guide the coaching process in tennis, [132] argue that a combination of both visual and verbal strategies can be effective if used correctly. In fact, empirical evidence has suggested that in tennis, the use of videotaped replay and loop-film technique has merit and can be

given consideration for use in instructional settings [114].

The subsequent sections in this chapter explore the state of the art in existing sports coaching tools and where sensors are widely used in the field of sports coaching software. Existing problems with sports coaching tools are explored before presenting the research objectives and contributions of this thesis which aim to enhance existing sports coaching software.

1.1 Existing Sports Coaching Tools

Video technology has long been used as the perfect feedback modality to record training sessions and use these recordings to improve an athletes performance. The following sections look at the different groups of coaching systems used today.

1.1.1 Commercial Sports Coaching Solutions

There is a lucrative market today for combining technology and sport, whereby there are endless off the shelf solutions aimed at improving a coach's knowledge of how to get the best out of an athlete. Many of the leading sports coaching products (SiliconCoach[1] , Dartfish[2] , Quintic[3] , Simi[4] , Templo[5]), do not focus on one particular sport, such as football or tennis, but aim to satisfy the objectives of sports coaches across multiple sports. Another important aspect of all the leading sports coaching software tools available today is that they all focus on a single modality, namely video. Professional video analysis software certainly does offer benefits to a coach, as vision analysis is a superb tool for portraying feedback to an athlete. The following subsections look at the various categories of sports coaching soft-

[1] http://www.siliconcoach.com/
[2] http://www.dartfish.com/
[3] http://www.quintic.com
[4] http://www.simi.com
[5] http://www.mar-systems.co.uk/

ware offerings today, in terms of analysis, video and instrumented coaching environments.

1.1.2 Analysis Sports Coaching Software

Post-match analysis software such as Avenir[6] , All Stats[7] and Football Technologies[8] , provide coaches with an abundance of useful data which can be captured during a match or post-match and then organised to help a coach make game winning decisions by studying various statistics on players, tactics used or plays to avoid. All major sports are swamped with analysis software packages, which provide interfaces to manually input data and in most cases these tools are vital to a sports coach. Existing analysis tools, however, are not technologically advanced and in the cases of all vendors, manual data input is necessary to generate meaningful archives.

1.1.3 Video Sports Coaching

Coaches mainly use sight and sound to sense an athlete's movement pattern, while touch can also be used in certain instances. They form a mental image of the skill and match this against the athlete's performance. Access to camcorders and laptops with commercial software over the past number of years has broadened coaches' awareness of the potential for technically enhancing coaching [66].

Hailes et al. [66] state that advances in video software technologies are making instant video feedback more commonplace. Visual coaching tools such as Vidback[9] are examples of commercial offerings today which are widely used in sport, while Dartfish Simulcam and Dartfish StroMotion[10]

[6] http://www.avenirsports.ie
[7] http://www.allstats.com
[8] http://www.footballtechnologies.com/
[9] http://www.simi.com/en/products/vidback/index.html

provide the coach with visual editing techniques only possible with high end production systems in the past. It should be noted that all these video coaching tools are portable by nature and each vendor assumes that the coach has access to a camcorder to capture video streams and a laptop to ingest the media files into the respective video coaching applications. These applications can exist in an online or an offline manner.

1.1.4 Instrumented Sports Coaching Environments

One of biggest technological breakthroughs in the sporting domain has been the success story of Hawk-Eye[11] , where high quality cameras have been deployed by governing bodies of sports to identify ball position and player position to a high degree of accuracy. Hawk-Eye has been deployed into several professional sporting organisations to date such as tennis, cricket and snooker, while investigations are currently underway to bring this technology to soccer and Gaelic games.

Hawk-Eye have also commercialised products for instrumented coaching environments in tennis and cricket. Hawk-Eye Tennis Coaching System provides a number of interesting features which include:

- How quickly a receiver reacts to a first serve or second serve;
- Any spin put on a ball during a serve or ground stroke;
- The location of a bounce during a stroke;
- Where the receiver returns a serve (first or second serve);
- Where those returns land (first and second serve);
- First serve percentage, points won on first and second serve;

[10] http://www.dartfish.com
[11] http://www.hawkeyeinnovations.co.uk

- How high the ball is when it passes over the net during a stroke;

However, one major drawback to this coaching system is that it is very expensive and this system requires on going court maintenance from expert system administrators making this system unaffordable to non-professional coaches. This system also lacks the ability to recognise the stroke type played by an athlete.

ProZone[12] provides a soccer analysis framework by instrumenting a football pitch with high quality cameras. ProZone provides semi-automatic frameworks for indexing sports video. The ProZone system (which, due to the cost of installation and game analysis costs, is generally only used by professional teams) uses eight cameras placed around a stadium to track players during a match. Each player (and official) is tracked via a semi-automatic process. The system employs a computer vision algorithm in which players are segmented by background subtraction from a static background image, followed by thresholding and connected component analysis. Each player is then tracked automatically. However, where there is a conflict, such as several possible players within close proximity or no possible choice of player, then the tracks are manually annotated. In addition to tracking players, events such as free kicks, corners and passes are all manually annotated. When a player touches the ball, this is recorded by clicking the correct event from a displayed list to indicate the event type. This ball data, combined with the position coordinates derived from the player locations, creates a ball trajectory data set. Both the player, official and ball data can then be employed to calculate tactical and statistical data. This style of instrumented visual analysis system is widely employed in soccer stadiums and is very expensive and its semi-automated operations results in a large amount of user input.

[12]http://www.prozonesports.com

1.1.5 Limitations of Existing Solutions

Approaches for visual event indexing for tennis have been previously reported [47] [127], however there has been no coaching system which is facilitated by automatic stroke recognition in tennis or any other court sports such as badminton or squash. Issues with existing instrumented sports coaching tools such as Hawke-Eye Coaching include its inability to automatically recognise which player has executed a stroke during play (due to change of ends) and most importantly its inability to recognise the stroke type played, which is essential for enabling coaches to generate high-level tactical queries. None of the existing solutions can automatically track a player over an entire match and this means that a coach will need to spend time manually annotating video after or during a capture session. Another drawback of existing instrumented coaching systems is the high cost associated with all these systems. With commercial video coaching tools such as Dartfish, coaches need to spend endless hours manually editing video after a capture, which is extremely time consuming and is a detrimental burden on coaching teams.

1.2 Research Objectives

There are several research objectives associated with building a next generation coaching system for racquet sports and tennis in particular. Our initial objective is to understand coaching requirements by working with coaches over the course of several seasons to fully appreciate how employing sensors in their everyday coaching environment will increase and improve their workflow and productivity.

Another objective is to enhance both the athletes' and coaches' education and training through capturing data on the athletes' performance and actions from multiple sensors located on body and via an instrumented court.

Very little research exists where the data from wireless inertial measuring units and visual sensors have been captured simultaneously for the purposes of sports coaching in any domain.

After the multi-sensor data has been pre-processed, a key research objective is to process and analyse the captured data to obtain information that is beneficial to coaches. Therefore it is pertinent that a concrete tennis ontology be created which clearly defines the semantics and rules of a standard game of tennis. This tennis ontology is then the foundation by which we can automatically index key tennis events which are of interest to coaches. When the semantic model is defined, variations between events detected in a capture session and the ideal actions as described by a professional coach can be identified and potentially visualised for feedback. This will be achieved by mining the ideal actions and comparing these to the captured data. This objective will reduce the time a coach has to spend browsing through archives and offers great potential if properly merged with existing research in video abstraction in the future.

Archiving the data over long periods of time will provide the coach and athlete with a vital tool for tracking trends and by applying machine learning algorithms to perform trend analysis, a further research objective will be to highlight information to the coach which would only have been possible previously through endless hours of manual archiving.

1.3 Research Contributions

Addressing the problem of using multiple low cost sensors to capture human actions in a wide area space has provided this research with several novel contributions. The first major contribution provides a novel approach for fusing multiple inertial sensors to recognise different human actions. A major contribution is the introduction of an early fusion approach, which fuses

multiple inertial sensors to provide an accurate human action classification system, which to the best of our knowledge is a first of its kind for human action recognition.

This early fusion event detection system is then applied to a wide area space to capture and recognise competitive tennis and automatically index a game of tennis into key tennis events. Therefore, the second contribution presents novel approaches to identify tennis events through early fusion of inertial sensors. Using a single inertial sensor (attached to a players forearm), this thesis proposes a set of algorithms to detect key tennis events including stroke recognition, rallies, games, and change of end events. Additionally, this thesis proposes a novel approach to detect the same tennis events (stroke recognition, rallies, games, and change of end events) using a fixed video camera infrastructure. This level of event detection in competitive tennis has never been achieved before and whether inertial sensors or visual sensors are used event detection is fully automatic.

Having all this information from different sensors creates a significant research problem in attempting to visualise this data in a manner such that a coach or athlete can use it to maximise performance. To overcome this problem, a third contribution is presented, in which a novel content management and retrieval system (Match Point) is presented which allows users to ingest the sensor information for a particular match and provides a user interface that allows users to query the system to find interesting queries. This system can automatically index the key events in tennis using either visual or inertial sensors, or a combination of both. The query engine which this system provides allows users to run queries which are simply not possible without automatic indexing of key tennis events. Such a complete system for event retrieval in competitive tennis has never been developed before.

1.4 Structure of the Thesis

Chapter 2 explores the technical background on visual and inertial sensing before looking at literature for machine learning. This thesis uses state of the art approaches to recognise motion using multi-sensor streams and the first two sections in this chapter help to unlock the theory which is essential to realising the full potential of these sensors. The final section in this chapter looks at the state of the art machine learning techniques, which are used for classification in this work.

Chapter 3 describes the challenges which need to be overcome to perform human action recognition using video. The chapter then outlines the computer vision techniques which are used specifically for human action recognition using video. The final section in the chapter details the approach used in this thesis for detecting human actions from visual sensors. Two approaches to extract human features from visual sensors (Contour Features and Histogram of Oriented Gradient of Motion History Image) are evaluated and the chapter closes by discussing the experiment results.

In **Chapter 4**, we explore the challenges which arise when using inertial sensors to recognise human actions. There are various approaches which can be used to detect human actions and activities and we explore the main approaches used to date in the literature. We then discuss details of the techniques used in this thesis to recognise human actions using inertial sensors. Experiments are conducted to evaluate recognition accuracy. After evaluating individual inertial sensors, we introduce the state of the art in multisensor fusion relevant to the work reported in this thesis. Various fusion approaches are then explored with a view to improving human action recognition accuracy. The first fusion technique explores the approach used to fuse the different sensors within an inertial sensor. Finally, experiments are conducted which use both early and late fusion to fuse the data from

both visual and inertial sensors. As the results prove, sensor fusion can significantly improve recognition of human actions.

Chapter 6 takes the visual and inertial sensing software and applies this infrastructure to the sport of tennis. Throughout the course of this research different techniques have been investigated, such as using visual sensors to recognise tennis strokes from contour features and using multiple inertial sensors to classify tennis strokes. This chapter explores these different techniques and evaluates their potential in practice.

Chapter 7 describes the novel content management and retrieval system which integrates the results of the various analysis tools. This retrieval system allows coaches to generate advanced queries which would not previously have been possible without an inordinate undertaking of manual annotation on the part of a tennis coach or tennis expert. An evaluation study determines if such a system can significantly improve coaching techniques and also investigates how automatic event indexing tools compare to existing state of the art commercial coaching tools.

Chapter 8 concludes this thesis. It briefly reviews the research contributions and discusses directions for future work.

Chapter 2

Technical Background

2.1 Introduction

This chapter contains an overview of the background literature necessary to understand the methods leveraged in the core work in this thesis. Firstly foreground extraction and adaptive background models are explored, which is necessary for visual event detection in tennis. We then explore the current state of the art in inertial sensing. Event detection using visual or inertial sensing is greatly enhanced by employing the best machine learning approaches available and we introduce the machine learning techniques which will later be employed to detect events in tennis.

2.2 Visual Feature Extraction

This research performs event based indexing of tennis videos and the following subsections explore the computer vision approaches which are vital to achieving satisfactory visual event detectors. In any scene not all pixels are of interest and it is the role of visual feature extraction to eliminate non-interesting pixels (background pixels) from the scene.

Object recognition aims to detect interesting objects in a scene, such as

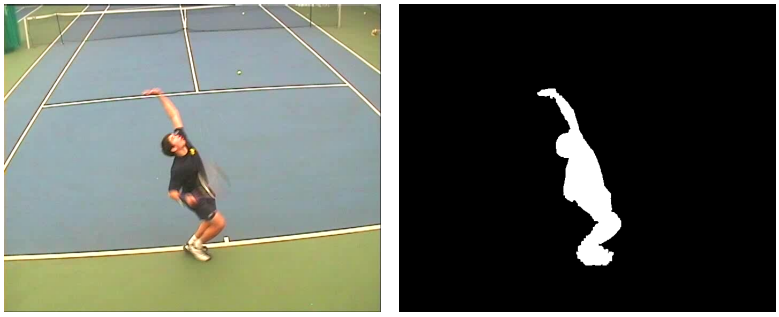
a moving car, a person or a potentially unusual detection. Of course, any video may have many different foreground objects visible, so it is a matter of judgement as to what objects are of interest to the user and should therefore be detected.

Feature extraction is the process of finding the main characteristics of a data segment that can accurately represent the original data [95]. It is the transformation of large volumes of raw data into a condensed set of features, known as the feature vector. In image segmentation, visual saliency is a process to estimate interesting objects from an image without prior knowledge of the image content [32]. In this thesis however, prior knowledge of the interesting objects i.e. humans is known and therefore the process of foreground extraction sufficient for detecting interesting objects. The following section describes this process in more detail.

2.2.1 Foreground Extraction

Foreground extraction is a well known technique for recognising moving foreground regions in computer vision as used in [36] [90] [19] and many more. This technique assumes a static camera is used and that image features, such as colour intensity or edge gradient information of foreground objects differ to that of the background.

In this context, a foreground object can quite often become a background object over time and therefore neither can be well defined. An example may be where a moving vehicle will be considered a foreground object, but when it is stationary for a period of time it will become a background object. A naive approach to background subtraction is to detect pixels belonging to foreground objects by determining if the difference between pixels in the current frame, f_i , and the corresponding pixels in an image of the scenes static background, b_i , are above a user defined threshold t. A pixel, (x, y),



(a) Original image (b) Foreground image

Figure 2.1: Shows the original image and the resulting image after the athlete is extracted as foreground.

is declared as foreground if

$$|fi(x,y) - bi(x,y)| > t$$

otherwise it is declared as background. There are various techniques applied to obtain the background model, b_i. For example [96] obtains an image of the scene without any foreground objects and uses this as b_i. Figure 2.1 illustrates an example where an athlete's silhouette is extracted from the foreground and displayed in a binary foreground image.

There are a number of reasons why this technique is naive. It lacks robustness to changing illumination, multi-modal backgrounds, and its assumption that b_i can be obtained at one instant in time. More advanced techniques generate their background model over time by observing the scene during a training period.

For example, in [83], a background variation model is built when the scene contains no moving objects. The background model is generated by extracting the minimum and maximum intensity values of each pixel, along with the maximum intensity difference between two consecutive frames. All these values are calculated during the training period when no foreground objects are present. The values are captured over several seconds and the

background model is updated at regular intervals for sections of the scene that the system recognises as being part of the background model. Every pixel in turn is classified as part of the background or foreground by analysing its values against the background model. Given the minimum (M), maximum (N) and the largest absolute difference between frames (D), pixel x from image I is a foreground pixel if:

$$|M(x) - I(x)| > D(x)$$

or

$$|N(x) - I(x)| > D(x)$$

Simply thresholding the image will typically not be sufficient to extract a clear foreground region as there will still be noise present, due to illumination changes, for example. In [83], the authors use a region specific noise removal process to remove noisy regions. Once thresholding is complete, a first pass of erosion is performed to the foreground regions to extract one-pixel noise areas. After the first pass is complete, a fast binary connected component operator is used to remove the minor regions of noise and finally erosion and dilation extracts the remaining noise which is larger than those already removed (morphological operations are explained in Section 2.2.5).

2.2.2 Thresholding

In the previous section on foreground extraction, the process of thresholding as a means to extracting a suitable foreground was introduced. The background model is used to compute the difference or distance between the background and any salient objects in the current scene. The technique used to detect any differences which appear different to the background is

called *thresholding*. Because of its intuitive properties and simplicity of implementation, image thresholding is commonplace in applications of image segmentation [62].

Finding an optimal threshold can be a delicate matter and will have a significant effect on the system's performance. If the threshold is set high it will result in a high number of missed recognitions, on the other hand if the threshold is set too low, the system will be too sensitive and there will be too many detections. A static threshold can be used in scenes where there are few changes in the properties of the image. By dynamically adapting the threshold to cater for different scenarios, these limitations can be addressed. There is a significant amount of research which has given great impetus to adaptive thresholding [30] [152]. The authors in [30] evaluate 40+ different thresholding approaches, six categories of thresholding algorithm are identified, each using a different measure to determine the optimal threshold. The measures used in each of the thresholding categories were: (i) histogram shape information, (ii) measurement space clustering, (iii) histogram entropy information, (iv) image attribute information, (v) spatial information and (vi) local characteristics.

Ridler and Calvard [152] use an iterative clustering approach to threshold selection. The mean image intensity serves as an initial estimate. Pixels are classified as foreground and background using this threshold and the threshold is iteratively re-estimated as the average of the two class means. In the conclusion of thresholding tests using synthetic data [6], it is noted that the methods of Ridler and Calvard, as well as Otsu's method [157], fail when the number of background pixels is more than 10 times greater than the number of foreground pixels.

Bouguet et al. [92] use different threshold constants for different regions of the image. The rationale behind this is that different types of regions

have different colour statistics. This region-based threshold constant is more flexible than the simplest method of using only one set of thresholds for the whole image. Also, compared to the method of having thresholds for each pixel, the approach is more practical and accurate. In [92] the authors choose a number of regions within the scene and as this number is not too high, it allows the threshold for each region to be set manually.

The previous approaches work well in the presented environments, however, this research will take place in an indoor environment, where changes in the lighting occur, so it is paramount that a successful background model should adapt to background changes. Therefore adaptive background mixture models have been identified as a solution to extracting a human subject as foreground. The following section briefly describes this approach.

2.2.3 Adaptive Background Mixture Models

There are various adaptive methods, including alpha-blending [83], Kalman filtering, Gaussian mixture models and averaging images over time. Averaging and alpha-blending are quick and simplistic but they are not efficient in scenes where there are multiple moving objects. The authors in [43] and [42] determine the background model to be the mean or the median of the previous m frames. Although quite fast, this approach consumes a large amount of memory.

A simple method of adaptive background estimation is to average the images over time. This creates a background candidate which reflects the current static scene except where motion occurs. This approach works effectively where foreground objects are in continuous motion and the background is visible a significant portion of the time. However it does not work well in situations where there are many moving objects and particularly if these objects move slowly.

In recent times, statistical methods have been employed to extract changing foregrounds from the background. Stauffer and Grimson [148] pioneered the adaptive background mixture model for real-time tracking, where a mixture of Gaussians modeled each pixel and the model was updated by an online approximation. The advantages of this approach were that the technique could operate in a scenario where there were multiple background objects by using a multi-valued background model. This approach is also immune to shadow, noise and change of lighting.

There are of course more advanced approaches to adapt the background model such as Kalman filtering, but a basic adaptive background model produces excellent results in an indoor environment where a human subject is constantly moving and there are very few foreground objects. There are also many contributing factors which can create a noisy background such as moving objects, moving trees, lighting switches etc, but in an indoor environment such as an indoor tennis court, lighting change is the only significant challenge. The probability density function for the Gaussian Mixture Model (GMM) used in this work is given by:

$$P(x) = \sum_{i=1}^{K} w_i \mathcal{N}(x|\mu_i, \Sigma_i), \tag{2.1}$$

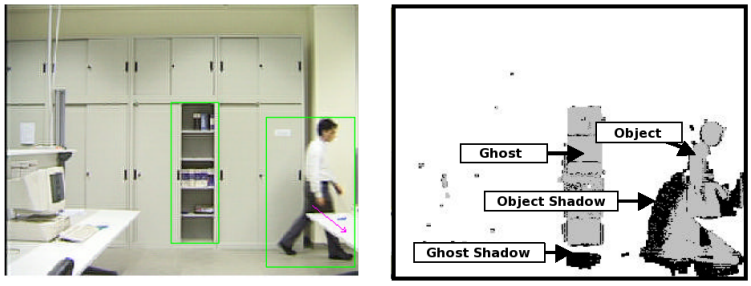
where $\mathcal{N}(x|\mu_i, \Sigma_i)$ is the probability density of a D-dimensional multivariate Gaussian distribution with mean $\mu_i \in \mathbf{R}^D$ and covariance $\Sigma_i \in \mathbf{R}^{D\times D}$:

$$\mathcal{N}(x|\mu, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right). \tag{2.2}$$

The weights w_i are each in the interval $[0, 1]$ and sum to unity: $\sum_{i=1}^{K} w_i = 1$. Given that we used a 3 dimensional matrix (RGB), $D = 3$.

2.2.4 Shadow Removal

Foreground objects often cast shadows which appear different from the modeled background, see Figure 2.2. If shadows are not dealt with appropriately



(a) (b)

Figure 2.2: Background Image; Background Edge Model;

they will create a noisy foreground object which will result in a distortion of the colour histogram, merging of objects and deformed foreground objects. There are many solutions proposed for shadow removal. Several solutions analyse the HSV color space, taking advantage of the fact that a shadow cast on a background does not significantly change its hue [41].

Other solutions take into account the saturation levels and also knowledge that any shadow will normally reduce the saturation of the pixels. In [74], a pixel is grouped into either foreground, background, shadow or highlight, by analysing the distortion of the brightness and the distortion of the chrominance of the difference. In an indoor tennis court however, we can deal with slow lighting modifications by slowly adapting the values of the Gaussians. Shadows can then be detected by detecting pixels which do change over a number of frames.

2.2.5 Morphological Opening and Closing Operators

Foreground regions like those generated by adaptive background mixture models are never complete silhouettes and always contain small holes which need to be removed. Morphological opening and closing operators are the most common approach to remove these holes and attempt to transform the blurry foreground object into a complete silhouette. Dilation (opening)

and erosion (closing) are two basic morphological operators, which can yield major improvements when properly applied in a feature extraction process. The feature extraction process used in this thesis applies both of these techniques to obtain the best foreground extraction results and this is achieved by finding a suitable configuration of adaptive parameters which best suit each application.

Dilation combines two sets of vectors using vector addition. The result of the dilation operation on an image is an expansion of the boundaries of all foreground objects. This results in an increased size of the areas of foreground objects and also any holes within foreground objects become smaller. In contrast to dilation, erosion is the other morphological operation which combines two sets using vector subtraction. The output of this operation is to erode the boundaries of the foreground objects, making the overall objects smaller and holes within objects expand. It is common to use both morphological opening and closing together to achieve a more complete foreground object. The authors in [85] state that if the closing of an image f is denoted as $C(f)$, and the opening of the same image is represented by $O(f)$, then a typical combination for this approach can be:

Proper opening:

$$Min(f, C(O(C(f)))) \tag{2.3}$$

Proper closing:

$$Max(f, O(C(O(f)))) \tag{2.4}$$

2.2.6 Motion History Images & Motion Energy Images

Davis and Bobick [45] describe a view-specific template of an action, where the action is represented over time. A clear foreground silhouette needs to be extracted in order for this approach to work. The basic idea used in [45] is to build a vector-image which can then be measured in terms of similarity

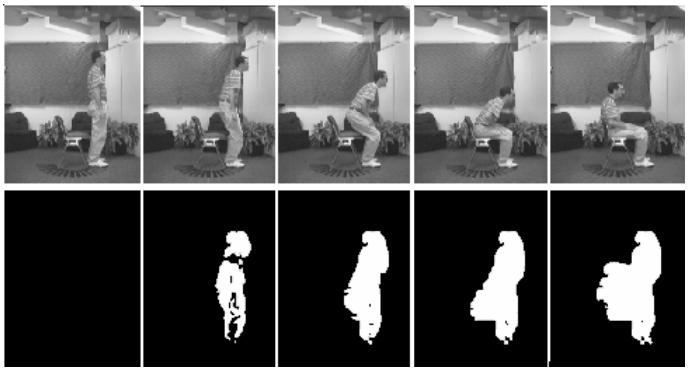


Figure 2.3: Motion Energy Image [45]

against recorded representations of given actions, called temporal templates. They use two slightly different techniques to represent an action (1) a motion energy image and (2) a motion history image and both are explained below.

Motion Energy Images

For illustration, we analyse the human action of someone sitting, as shown in Figure 2.3. The row above shows the raw key frames of the action. The bottom row shows how the foreground silhouette of the human subject would be aggregated over the raw frames above. The idea is that the aggregated foreground can be used to describe the motion in that region. These aggregated foreground images are known as *motion-energy images* (MEI). Let I$(x; y; t)$ be an image sequence, and let D$(x; y; t)$ be a binary image sequence describing regions of motion of n images. The foreground MEI E $(x; y; t)$ is defined as

$$E_T(x; y; t) = \bigcup_{i=0}^{n} D(x; y; t - i)$$

A duration variable τ is used to define the temporal properties of an action. The authors in [45] use a backward seeking algorithm which can dynamically search over a range of T.

Motion History Images

$$H_\tau(x,y,t) = \begin{cases} \tau & \text{if } D(x,y,t) = 1 \\ \max(0, H_\tau(x,y,t-1) - 1) & \text{otherwise.} \end{cases}$$

Figure 2.4: Formula for Motion History Image

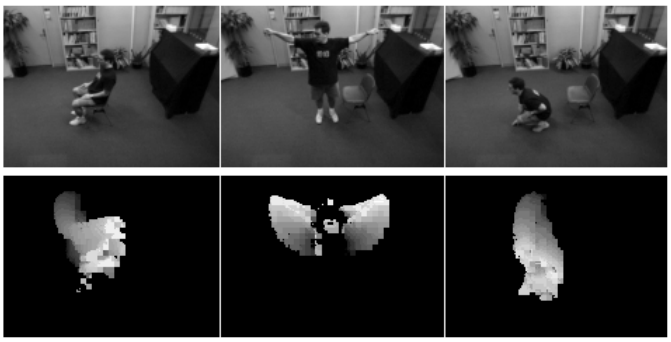


Figure 2.5: Motion History Image [45]

The sister process of MEI is motion history images (**MHI**) which describe when (rather than how) movement occurs within the action sequence. In an **MHI** H_T , pixel intensity is a function of the temporal history of motion at that point.

The output is a scalar-valued image where the pixels which have moved most recently have a brighter intensity. An example of an MHI can be seen in Figure 2.5, taken from [45]. The formula for Motion History Images is given in Figure 2.4. It should be noted that an MEI can be generated by thresholding the MHI above zero. Although MHIs do not account for optic flow (the direction of image motion [18]), temporal clues are obtained by identifying the brightness of the foreground object.

In this example the MHI describes the flow of motion: whereby the downward movements of the arm is older than the upward movement of the arms. Where the movements are quite basic and there is no occlusion, MHIs will give a good visual representation of the action being performed. As motions become more complex, the optical flow is more challenging to

identify, but is still represented (only not as well as basic actions). The reason for MHIs popularity is down to its simple nature and low computational cost compared to optical flow, for example.

2.2.7 Histogram of Oriented Gradients

Another common approach used to extract features for human action recognition is to compute a histogram of oriented gradients (HOG). HOG is a commonly used technique to recognise objects in images. HOG are inspired by Scale-Invariant Feature Transform (SIFT) descriptors [108]. This concept is built around the idea that local shape information is often well described by the distribution of intensity gradients or edge directions even without precise information about the location of the edges themselves.

Algorithm Overview

The image is divided into small sub-images called cells. Cells can be represented as rectangular (R-HOG) or circular (C-HOG). A histogram of edge orientations are accumulated within each cell. The combined histogram entries for all the cells are used as the feature vector describing the object. To provide better illumination invariance (lighting, shadows, etc.) the cells are often normalised across larger regions incorporating multiple cells called blocks.

To find a set of descriptive features, a unique HOG can encode local gradients and the first step in this process finds the gradient for each pixel. The training image (see Figure 2.6) is then decimated into a series of sub-windows, known as "blocks". These blocks can vary in size from 8 to 64 pixels, they can have various length to width ratios. All blocks are then divided into quadrants and the HOG for each quadrant is calculated.

HOG are capable of detecting gradients or edges that characterise the

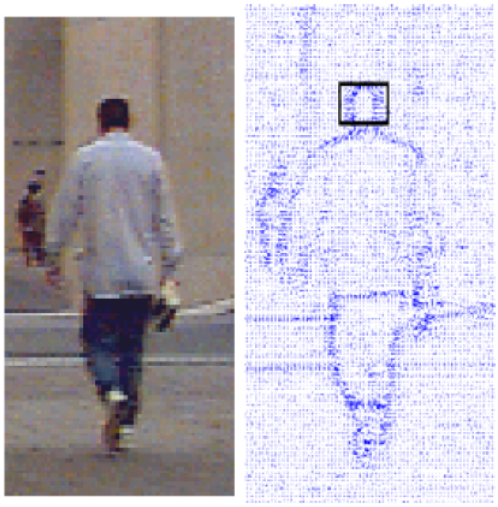


Figure 2.6: A pedestrian image (training) and the resulting gradient [22]

local shapes. It is also an effective approach to tune the spatial and orientation sampling densities for differing applications. In human detection, the authors of [44] use coarse spatial sampling and fine orientation sampling to prove that HOG descriptors significantly outperform existing feature sets for human detection.

For recognising shapes, approaches which use the HOG approach are considered accurate for visual classification. One such example which uses HOG is reported in [65], where the algorithm has been employed with color features and shape modelling. This approach can reliably detect vehicles even when multiple occlusion is present in the scene. However to make this approach suitable for recognition of rear view human postures, several parameters were adjusted in order to achieve maximum recognition accuracy. The theory behind HOG is that local object appearance and shape is well characterised by the presence of local intensity gradients or edge directions, even without the knowledge of where the edge positions occur.

2.2.8 Applications

Automatic recognition of human movements and actions is a vibrant research field due to the ubiquitous nature of video recordings in our everyday lives, as well as the inherent complexity of the task. This field produces numerous and often difficult challenges such as determining the kinematic structure and movement of a self-occluded three dimensional entity from video frames. This complexity provides lots of interesting challenges from an academic perspective. From an application perspective, computer vision provides a non invasive solution, making it an attractive approach [117]. Applications in this area can be grouped into any of the following groups: surveillance, control, or analysis. The following three sections describe these three groups.

Surveillance Applications

Security and surveillance applications have traditionally used networked video cameras, which are monitored by a human surveillance team. Any irregular activities are then acted upon in real-time by security. However with the reduction in cost of camera networks, more and more surveillance cameras are being used everyday, which in turn requires more human resources to monitor areas. This problem has naturally lead to a surge of activity in the area of vision-based solutions which can replace or assist the need for a human operator. Automatic recognition of unusual human activities in surveillance cameras is one area which has seen a surge of activity in recent times [139].

Other security applications include those which automatically monitor and understand potentially crowded locations such as train stations or airports. Example applications include crowd counting [58], crowd flow [103] and congestion analysis [83]. With the increased awareness of security, today's surveillance applications analyse actions, activities and behaviors both

individual people and also for groups of people. Example applications in automatic surveillance include anti-social behavior [97], shopping behavior analysis and person detection [134].

Control Applications

These applications estimate human motion or human poses to manipulate an output. This could be used as an interface to a game, e.g. EyeToy[1] , or more generally for human computer interfaces. It could also be used within the entertainment industry where personal avatars can be based on the captured semantics, such as shape and motion to create a more realistic product. Microsoft Kinect[2] , which is a motion sensing input device and is used for entertainment gaming is an example of this type of application.

Analysis Applications

Motion capture may also be used to analyse and optimise the performance of athletes [46] [50] or to automatically diagnose patients [130]. More recent applications in this domain allow for video annotations along with content based retrieval and the motor industry has also embraced visual research capabilities to address challenges such as automatic airbag activation, sleep detection, impeding pedestrian hazards or lane following [98] [100].

2.3 Semantic Video Indexing

The amount of multimedia content has increased rapidly in recent years due to lower cost capturing equiptment and the widespread use of video over the internet . There are many applications for multimedia today such as live braodcasts of various programmes, advertising, movies, or even picture

[1] http://www.eyetoy.com
[2] http://www.xbox.com/en-US/kinect

sharing. With this has come advancements in the technology for media capture, storage and transmission, where newer, more sophisticated and cheaper capture devices are constantly expanding the growth of multimedia. As content generation and dissemination increases, more advanced tools to filter, search and retrieve this content in an efficient manner also becomes more important. The authors in [120] state that the existing tools for information retriveal in text databased cannot be adopted for video and the lack of retrieval tools for efficient access and data mining could render most of this data useless. In most cases many of the existing video retrieval systems do not process the audio content and intead focus on the video semantics.

Semantic video indexing is the first stage toward automated video browsing, retrieval and personalisation. Semantic video indexing allows the end users to retrieve videos based on their interest and preference with regards to the content within the video. It is essentially the process which attaches concept terms to video segments. The authors in [120] state that the difficulty lies in the mapping between low-level video representation and high-level semantics. To overcome this problem they therefore formulate the multimedia content access problem as a multimedia pattern recognition problem. Automatic temporal video segmentation methods typically involve computing pixel-level and/or histogram-based difference measures for each pair of consecutive frames in the video. These methods then use shot boundary detection methods to detect shot boundary positions [171], [54]. More advanced temporal segmentation methods use low-level image features such as edges [174], focus of expansion points [7] and image motion [129].

2.4 Inertial Sensing

Measuring human motion involves sensing the movement in a three dimensional space and is often a complex task. The Six degrees of freedom (6DoF)

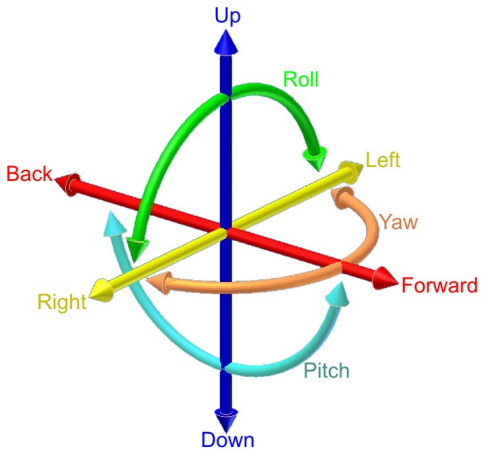


Figure 2.7: Six degrees of freedom (6DoF) of a rigid body in a 3D space.

refers to this freedom of movement of a rigid body in three-dimensional space. Specifically, the body is free to move forward/backward, up/down, left/right (translation in three perpendicular axes) combined with rotation about three perpendicular axes, often termed pitch, yaw, and roll. Figure 2.7 illustrates the 6DoF and inertial sensors are used to measure movement in this space.

The amount of academic literature which deals with wearable inertial sensing in the area of human action recognition has only began to grow in recent years largely due to the relatively recent drop in cost of inertial sensors. While Chapter 4 describes the latest research in using inertial sensors for human action recognition, the following section introduces what is meant by inertial sensors and the applications to which they have been applied. Stubberud et al. [150] explain that inertial sensing can be found in guidance, navigation, and control systems for small space vehicles, such as satellites, which use inertial measurement sensors to calculate accurate positioning, velocity and angular information about the vehicle.

Inertial sensors are commonly used for motion recognition today and this is largely due to their ability to overcome challenges such as line of sight and

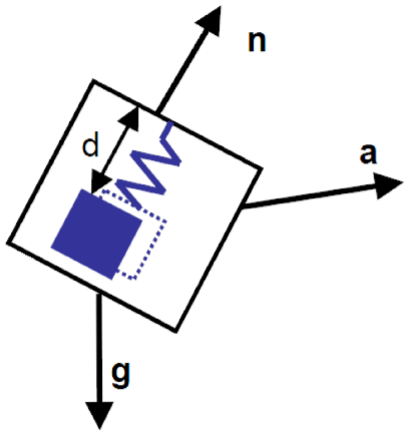


Figure 2.8: Single Axis Accelerometer [109].

mobility issues which are associated with other sensors. Different technologies which employ inertial sensors include aircrafts, cars, smartphones and gaming consoles. In a typical use with humans, inertial sensors are strapped to a body segment to measure angular rotation or acceleration [135].

Thankfully, recent developments in these sensors have made this technology inexpensive, which helps to drive forward advancements and usage. The following section discusses literature on inertial sensing devices and how they capture real world data. The first subsection looks at how accelerometers are designed and how acceleration is measured. The remaining subsections look at gyroscopes and magnetometers and then wireless inertial measuring units respectively.

2.4.1 Accelerometers

The authors in [109] describe an accelerometer as a mass, suspended by a spring in a void chamber (Figure 2.8). The mass moves in one direction and its sensitivity is measured by the accelerometer. The dislodgement of the mass can be used to calculate the acceleration experienced by the mass

along that particular sensitivity axis.

A more common style of accelerometer used today is the tri-axial, where three single axis accelerometers are mounted together. To find the three gains and offsets and how each axis is oriented in relation to how they are housed, equation 2.5 is used. The output S_{yA} can be associated to the initial acceleration and gravitational force by the following equation:

$$S_{yA} = S_a - S_g \tag{2.5}$$

In this thesis, accelerometer signals are obtained using tri-axial accelerometers where the output is obtained by equation 2.5, after calibration.

2.4.2 Gyroscopes

Also in [109], the authors describe several different approaches for making angular rate sensors (gyroscopes) such as laser gyroscopes, spinning rotor gyroscopes and vibrating mass gyroscopes to name but a few. The standard spinning rotor gyroscope and laser gyroscopes are primarily used for navigation. These traditional styles of gyroscope are unsuitable for detecting human motions because of their large size and high expense [142]. A more suitable gyroscope for human motion is the small and inexpensive vibrating mass gyroscope.

A 2D perspective of a standard gyroscope is shown in Figure 2.9. A mass is moved in the direction denoted by r_{act} .The dislodgement experienced by the mass can be calculated by examining the direction at right angles to the movement direction. When the unit experiences a rotation where the angular speed is at right angles to the plane, coriolis force will be experienced in the direction at right angles to the angular speed. The amount of coriolis force f_C is denoted by:

$$f_C = 2m.v.w$$

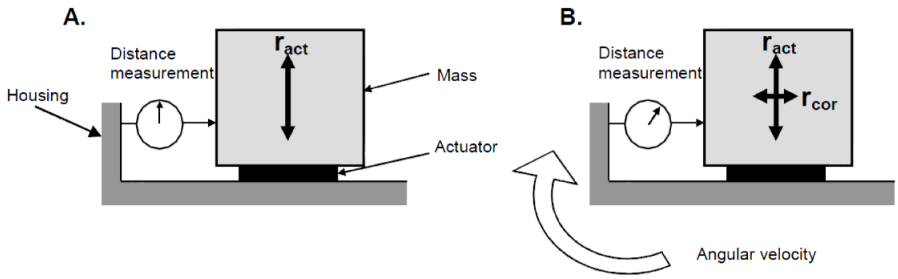


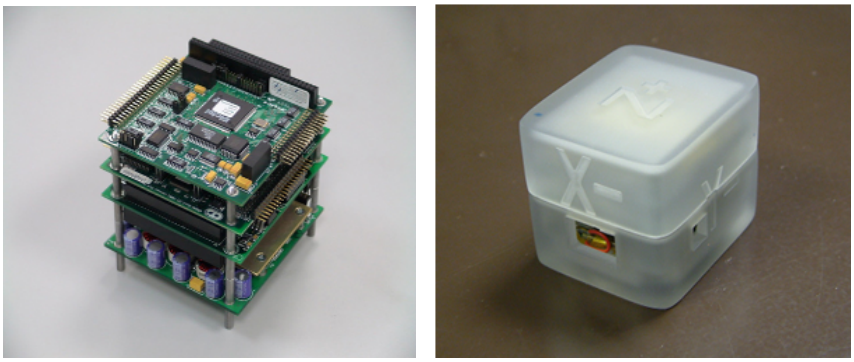
Figure 2.9: A.: A gyroscope is made up of a mass, which vibrates in the path illustrated by $\mathbf{r}_{act}$. B: Rotation of the gyroscope will cause the vibration of the mass and also a dislodgement. The angular speed can then be inferred from the coriolis force generated. [109]

in that m represents the mass, v is the speed of the mass at that moment and w is the angular speed. Therefore the dislodgement due to coriolis force is relative to the angular speed and is used to measure angular speed.

2.4.3 Magnetometers

A magnetometer is a device which can be used to measure the properties of the natural magnetic field or an artificially generated magnetic field. One possible field of detection is the Earths magnetic field. Magnetometers are generally used to measure a compass heading information and can be regularly used to offset integration drift in gyroscopes, which is a common problem with gyroscopes [15].

Magnetometers cannot sufficiently measure 3-D orientation alone, though they do record slight responses to changes in 3-D orientation. As this thesis will discuss in detail in Chapter 4, the data captured by magnetometers can be of significant value when fused with other sensors such as accelerometers and gyroscopes.



(a) (b)

Figure 2.10: One of the WIMU designs used in this research

2.4.4 Wireless Inertial Measuring Units

An Wireless Inertial Measurement Unit (WIMU) is a sensor unit which has tri-axial accelerometer and a tri-axial gyroscope. A calibrated WIMU can measure 3-D angular speed and 3-D acceleration.

Where the orientation and initial position of the WIMU is known, these signals can be used to determine kinematic movement of the WIMU. The orientation can be calculated by recording the original orientation and then measuring the difference in the current orientation which may be obtained from the gyroscope. Figure 2.10(a), taken from [57], shows a modern inertial measuring unit, which is suitable for mounting on a human body given its small size.

Figure 2.10(a) shows the internals of the WIMU which has been designed specifically for attaching to the body of tennis athletes and was used in this work. This device was developed by our colleagues in Tyndall. The assembled hardware of the WIMU is composed of a NAP150 Board (power supply, battery management and integral dual-axis 70g/37g accelerometer), an IMU layer (9x motion sensors), a 1Mbps Nordic Radio with ATmega128 layer (wireless communication and processing) and a prismatic lithium polymer battery (1230mAh). The IMU layer consists of a tri-axis accelerometer with

a range of ± 10g, the gyroscope range is ±1200/s and the magnetometer can measure ±6gauss.

Calibration of a wireless inertial measurement unit (WIMU) is the process of finding the parameters which allow us to convert the values recorded by the WIMU into meaningful physical units. The reason such a process is required is because a sensor's readings can deviate from node to node, due to manufacturing inconsistencies. Sensor readings can even deviate within the same device as a result of temperature conditioning and power supply voltage. For these reasons it is necessary to calibrate the data generated by a WIMU and there are many methods available to do this. The sensors used in this thesis are pre-calibrated.

2.4.5 Applications

In this section we discuss applications for activity recognition systems in wearable or mobile settings. We begin with applications for healthcare and assisted living, which represent an important class of applications. In addition to this there are three other application groups discussed, industrial application uses, applications for the entertainment and gaming industry and applications within the sporting domain. Finally, this section briefly introduces the other applications areas which commonly use inertial sensors.

Healthcare and Assisted Living

Currently, activity recognition and context-aware computing is often motivated to enable new health-related applications and technologies for the aging. Longer life expectancy is increasing the percentage of the elderly population in societies all over the world and posing difficulties to existing healthcare systems. It is hoped that technology can help overcome these challenges, for example by helping the elderly to be more independent and

also to reduce the need for human assistants to the elderly.

One such system aims to prevent age-related diseases or serious medical conditions before they occur. These disease prevention sensors use data gathered over long periods of monitoring to detect changes or irregular patterns in a subject's daily activities which may indicate early signs of diseases such as Alzheimers. Automatic recognition of subtle changes in behavior is an active area of research and inertial sensors which accumulate data and have the ability to give summaries of daily activities [33] or applications which aggregate data from physiological parameters [104] [6] are currently providing huge benefits to physicians and carers to determine the health of a patient.

Another application of inertial sensing in the area of health and well being is the use of context information to champion a more active and healthy lifestyle. It may also be used to facilitate disabled or elderly patients in executing daily activities. The authors in [112] use variations in a mobile phone signal to recognise if a user is active and can provide summaries of the levels of activities in which a subject is engaged as motivational feedback. In [39] a similar approach is described which uses inertial sensors for activity recognition so that when a flagged activity is performed by a human subject, a feedback system displays virtual rewards on a mobile phone screen. Another approach uses a combination of data captured from localisation and activity sensors on wearable devices to suggest physical activities, e.g. by noting that the user has enough time to walk to the next bus stop instead of waiting at the current one [5].

Ground breaking improvements in wireless technology have helped to drive forward body worn sensor devices, which can be used in everyday lives. Motion capture sensors which can be body worn are mainly used for healthcare monitoring [101] [126]. The most commonly used sensors worn for the

detection of motor movements in the healthcare domain are accelerometers and gyroscopes [91]. These sensors can be attached to any body part or even attached to sports equipment and collect data on movements performed by athletes. Some approaches use portable virtual training applications, which assist training and provide the subjects with useful information like instant notification of mistakes made during a session at anytime and in any location [84].

Industrial Applications

In an industrial environment, activity-aware applications can potentially support workers in their duties, reduce unnecessary mistakes and improve safety in the workplace, for example. Wearable computing systems which support workers in communications, accessing necessary information, or data collection, have been commercially available since the early 1990s from organisations such as Xybernaut[3] . As was highlighted in an overview in [145], the first organisations to adopt this wearable technology [such as the shipping, airline and telecommunications industries] were those who used complicated and expensive systems.

There is currently on going progress to harness advancements of industrial systems, which make better use of multimodal sensors, for example, by detecting context information such as the current location or the activity being performed by a worker. An example of this is given by [110], who give an overview of technical uses for body worn sensors in motor manufacturing assembly lines, aircraft maintenance, hospitals and emergency response units. In the context of each of these applications wearable sensors and activity recognition is used to give hands free interactions to data and assist in training of new workers, provide a summary of activities performed by

[3] http://www.xybernaut.com/default.aspx

the workers. In [160], Ward et al. fuse information from body worn microphones and accelerometers to recognise individual worker activities in a wood factory such as sawing or hammering.

Entertainment and Games

Body worn activity recognition systems can be found in many entertainment devices today, such as games consoles and mobile smartphones, where these devices provide a range of customised applications for entertainment and gaming. Inertial sensors have been used to recognise human activities in contexts, such as performing arts, where sensors are attached to dancers to augment their performances with interactive multimedia content that correlates with their movements. Gowing et al. [64] propose a system for augmented reality-based evaluations of Salsa dancer performances. They present a novel technique to achieve dancer step segmentation jointly using audio signals captured by the on floor piezoelectric sensors and signals from the WIMU devices which are attached to the dancers.

Other systems in performing arts are described in [9] [52] [12], which use inertial sensors attached to the human body to collect data, which is later classified using machine learning systems and the data is then used to visualise the dancer's motions. It should be noted that application designers in this domain have been early adaptors of body worn inertial sensing technology and this is also expedited by the fact that entertainment applications are non-critical in comparison to healthcare applications. The gaming industry has been a leader in harnessing the power of inertial sensors. The authors in [175] describe a motion sensor attached to the body and other objects used to control computer games. In [72], Heinz uses inertial sensors attached to the human body to detect moves which then control computer games. Inertial sensors are of course widely used in computer games console

systems, such as the popular Nintendo Wii[4] .

Sporting & Leisure Applications

Inertial sensors have also found significant use in sporting applications. In [53] the authors developed a system to recognise basic activities and also sporting activities such as playing soccer, riding a bicycle and exercise routines such as rowing. They used Neural Network classifiers to recognise the different activities. Long et al. [107] compute the amount of energy expenditure which is used to perform certain daily activities and also sports activities such as table tennis, football, volley ball etc. Motion detection in the context of martial arts activities can be recognised by placing a tri-axial accelerometer on the torso to capture the bodily acceleration for recognising rapid movement human activities. The research described in [71] uses both gyroscopes and accelerometers attached to the body to recognise actions in Wing Tun to increase interaction in video games of martial arts, with a view to using similar systems for martial arts education.

In [143], the authors present an on-body wireless sensor system for measuring activities during snowboarding in real-time and the apparatus is shown in Figure 2.11. They use inertial sensors to measure force, along with an intelligent communication setup in a wireless network to capture and analyse a snowboarders posture and motion on the snowboard. In [60], the authors develop signal processing algorithms to measure the angular rotations of wrist during golf swings, while in [71] it is described how to use body-worn sensors, accelerometer and gyroscopes in particular, to record the actions made by humans in martial arts. The data acquired is then used to find the quality of the moves and level of expertise of the person making those moves. In [8], the authors model the golf swing as a double pendulum

[4]http://wii.com

Figure 2.11: Wireless inertial sensing setup to sense a snowboarder's motions during action [143].

system and use inertial sensors placed along the body and golf club to determine how closely the movements of the body follow predetermined motion rules.

Commercial systems which used inertial sensors in the sporting area include Nike+[5] , which monitors an athlete's sporting activities. A device is placed inside the shoe, which can keep record of running and jogging exercises and the information can be aggregated over time to give an activity history. This sensor can be integrated with auxiliary devices such a smartphone and can be used for training purposes or to collaborate and interact with other users. Polar[6] also develop numerous products which can record an athlete's training and performance and even integrate into a team sports scenario. One of the most popular inertial sensor applications today is the various running tracking applications for smartphones such as Apple's iPhone[7] where ideas which originated in the activity and context awareness research community are now being harnessed by companies and independent software developers and are accessible to anyone with a smartphone.

[5]http://nikeplus.nike.com/plus/
[6]http://www.polarusa.com/us-en
[7]http://www.apple.com/iphone/

Other Application Areas

Inertial sensors have been employed in other areas to achieve activity recognition. For example [116] uses body worn inertial sensors to recognise soldier activities. The inertial sensors can log soldier activities during a mission which can be used to automatically generate action reports or assist in covert operations during a mission. The work described in [137] exploits the potential of activity recognition with inertial sensors to target mobile advertising.

2.5 Sensor Fusion

The vestibular system in the inner ear of humans and animals provides inertial information which is necessary for navigation, body posture and orientation. Humans cannot perform efficient head stabilisation and visual tasks without this system. The information captured by the vestibular system is used to execute visual movements such as tracking and gaze holding [49]. It is well known that human vision and the vestibular system fuse neural signals at a very early processing stage [16]. The inertial information improves the accuracy of the vision system and the visual cues aid the spatial orientation.

Huge benefits can also be gained in the field of computer science by fusing multiple sensor streams and later in this thesis we will introduce our approach for fusing multiple sensor streams to achieve higher classification accuracy than can be achieved by any one sensor stream alone.

Traditionally, inertial sensors have been used for navigation and also for guidance of defence systems. Positioning, velocity and altitude are measured using precise accelerometer and gyroscope sensors, which are combined with localisation technologies such as Global Positioning Systems (GPS) and radars to name but a few. With all these informative signals generating

masses of information, intelligent methods of fusing salient points of information are required. Every sensor has strong and weak aspects and no sensor is perfect at measuring a particular force. Sensor fusion is the process of fusing the multisensory information to obtain interesting information, where data is combined from various indirect and noisy instruments.

Sensor fusion has many different applications such as automatic target recognition (e.g., missile guidance), automatic vehicle guidance, combat surveillance and automated threat detection systems, such as identification-friend-foe-neutral (IFFN) systems [67]. Non military applications include observing manufacturing processes, robotics [87], and also within medical devices. The principle techniques used to combine the data is formed from more established fields such as digital signal processing, statistical estimation and artificial intelligence [68] [111]. In 1985, the Joint Directors of Laboratories (JDL) formed the Data Fusion Group and the JDL published a model which divided the various processes concerned with data fusion into six levels [163]. This widely used model is still used today and provides valuable guidelines for data fusion. Other popular approaches to fuse sensors are Bayesian Fusion [113] or Kalman Filtering [23], where either of these approaches can be used to combine data from various indirect and noisy instruments.

The purpose of sensor fusion is to use beneficial features of one sensor to overcome the limits of features in another sensor. An example would be where magnetometers are employed to eliminate integration drift associated with gyroscopes. In this particular scenario, iron and magnetic equipment will interfere with local magnetic fields and effect the orientation estimation. Errors in the gyroscope drift will have different patterns than those found in the local magnetic field and with this knowledge in mind, gyroscopic drift can be reduced. There are two approaches to fusion: early and late. After each sensor has completed its capture, the difference between early and late fusion

lies in the approach each uses to merge the results of the different sensors together. Later in this thesis we will introduce these fusion techniques.

2.6 Machine Learning

Data mining is a methodology which can infer new knowledge by analysing trends or patterns from raw data. These patterns can lead to potentially useful new knowledge that is not always obvious without intelligent computational analysis [164]. Machine learning is a technology which enables data mining. In principle it requires computer programs which are trained to find patterns in data. The following sections describe the main principles of machine learning. First, we introduce Bayes' theory, which is a fundamental equation for statistical learning. This is then followed by an introduction to classification and regression and the main differences between generative and discriminative models are then introduced. Then we introduce supervised and unsupervised approaches to machine learning before introducing the main concepts of the two classification methods used in this theses (Instance Based Learning and Bayesian Networks).

Bayes' Theorem

Bayes' theorem is widely used to find probabilities in machine learning and is essential to Bayesian Networks. In mathematical terms, Bayes' theory denotes the relationship of the probabilities of A and B, $P(A)$ and $P(B)$, and the conditional probabilities of A given B $P(A|B)$ and B given A $P(B|A)$. In its most common form, it is:

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

2.6.1 Regression & Classification

Regression and classification are two machine learning approaches to create models of prediction from computer readable data. The objective of classification is to learn a mapping from the feature space, X, to a label space, C. For example, where feature space $X \in R^d$ and the label space $C = \{0, 1\}$, the function $f : R^d \rightarrow C$ can be used to classify each example in X to its most probable discrete value in C. It is this mapping (or function), f, which is the classifier. The objective is to minimise the generalisation error.

The goal of regression is to learn a mapping from the input space, X, to the output space, C. This mapping, f, is called a function. For example, we might have, $X \in R^d$ and $C = R$, then regression will use the function $f : R^d \rightarrow R$, to determine which indiscrete output a given example belongs to. The major difference between regression and classification lies in the dependent variables used. Regression uses numerical dependent variables only, whereas with classification, the dependent variables are clear-cut. Classification has a fixed amount of unordered variables, while regression has indiscrete values or discrete but ordered values.

2.6.2 Generative & Discriminative Models

There are two differing models commonly used in machine learning, generative and discriminative models. Essentially a generative model learns the joint probability distribution $p(a, b)$ and a discriminative model learns the conditional probability distribution $p(b|a)$, which is the probability of b given a. At its most basic, if we have the given data in the form (a, b), (1,0), (1,0), (2,0), (2, 1). Table 2.1 gives the calculations for $p(a, b)$ and Table 2.2 gives the probability distribution of $p(b, a)$.

Discriminative algorithms provide a classification method for naturally distributing a given example a into the class b. The distribution $p(y|x)$ pro-

	y=0	**y=1**
x=1	1/2	0
x=2	1/4	1/4

Table 2.1: $p(a, b)$

	y=0	**y=1**
x=1	1	0
x=2	1/2	1/2

Table 2.2: $p(b, a)$

vides a logical distribution to classify an example x in class y and algorithms which use this model are called discriminative algorithms. Generative algorithms model the data as $p(x, y)$.

Overall, generative models allow for degrees of ambiguity, uncertainty and generalisations. In addition, they tend to be efficient in handling large amounts of data, and are hence most conducive to modeling time-series data [1]. Popular schemes include Naive Bayes, Gaussian (Mixtures), Hidden Markov Models, Bayesian Networks, to name but a few. Common discriminative approaches include K-Nearest Neighbor, Support Vector Machines, Neural Networks, to name but a few. While these techniques are different, they share a common characteristic in that, towards finding the exact decision hypothesis that minimizes classification errors on the training data, each aims to predict the class label directly based on the feature representation [164].

The decision criteria for selecting either a generative or discriminative supervised approach has been a constant source of debate in the field of machine learning, resulting in a variety of studies on the subject being published in the literature. For example, the authors in [144] argue that where there is a low amount of training data, a generative model is most suitable, since using a discriminative approach may lead to problems with overfitting. The authors in [170] also claim that generative models are most applicable

when there is a lot of ambiguity and not enough data to train against. The authors in [144] suggest that discriminative models lack the sophistication of generative models, and can be problematic since they will need manual configuring (e.g. penalty functions, regularisation, and kernel functions), and that the relationship between variables are not well defined, i.e. they are "black-boxes".

2.6.3 Supervised & Unsupervised

There are two main methods used in Machine Learning, supervised learning and unsupervised learning. Unsupervised learning concerns the identification of obscure structures from unmarked data. Given that the instances fed to the system are unmarked, there is no evaluation metric which the learner can use as a signal of how precise it is at distinguishing different instances.

In the supervised approach, machine learning is achieved by annotating the data, which in turn helps to infer a function or a pattern within the data. A training set is used in supervised learning which contains a series of instances or examples which will have two properties, an input vector and an intended output value (class value). After the training data is analysed, the supervised learner infers a function, which is also called a classifier. It is this inferred function which then predicts the most likely output class for any previously unseen input instance.

In the supervised machine learning approach, there are three main areas. Binary classifiers classify examples from a defined set into two classes, on the basis that the examples of each class share some common properties. An example of binary classification would be for enforcing quality control within a manufacturing process; i.e. to decide if a product is in satisfactory condition to be sold or if it should be marked as defective. The second supervised learning approach is known as multinomial classification and this

is the process of classifying examples (or instances) into two or more groups. The third area is regression, which is explained above. In the next section some commonly used concepts in machine learning are introduced. Then, the remainder of this section discusses the two supervised machine learning approaches which are used in this thesis, Bayesian Networks and Instance Based Learning.

2.6.4 Concepts, Attributes and Instances

In Machine learning, it can be said that an input is made up of attributes (features), instances (examples) and concepts (classifier). In general terms, what we are attempting to infer from the learning process is an intelligible description of what the data represents as a concept which can be understood by the classifier and used to detect similar trends. In supervised learning, which is used in this work, the learner is presented with a set of examples (training set) from which it is expected find a way to learn how to classify unseen examples.

The information the learner is supplied with is called the instance. Each instance is a separate independent example of the concept to be learned. The input to a supervised classifier is a set of instances and these instances are contained within a training dataset. The characteristics of each instance is determined by the attribute values which the instance contains. These specific values are a measurement of some aspect of the concept to be learned.

2.6.5 Instance Based Learning

Instance based learning (IBL) is a classification method that compares new problem instances with training instances, rather than explicitly performing generalisations. IBL algorithms, which are a form of lazy learning, are derived from the nearest neighbour classifier [40]. In general, there is no

preprocessing of training sets and to classify a new instance, all training instances are compared with each test instance.

With IBL, training instances are unmodified and a distance function is used to calculate which training instance is closest to the unknown test instance. Essentially when the closest training instance has been detected, the classifier predicts that the unknown test instance belongs to the same class as the closest training instance. IBL can classify instances which contain either numerical or nominal attributes. In this work, all the attributes are numerical.

The Distance Function

Different distance functions may be used such as Euclidian and Manhattan distance depending on the specific application. In this work, Euclidian distance is applied and can be denoted as:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_k - y_k)^2}$$

where k is the number of attributes. Since we are dealing with numerical attributes the distance between two attributes is essentially the numerical difference between the two.

Efficient Nearest Neighbour

The most common type of IBL is the Nearest Neighbour method, where the distances between an unknown instance and each instance in the training set is calculated. However training sets may have a large number of instances and comparing each training instance to the test instance can be computationally expensive and time consuming. This method is proportional to the number of training instances, as the quantity of linear comparisons will grow as the number of training instances grow. Thankfully, there are more

efficient methodologies for finding nearest neighbours and the most common of these is called kD-tree, where k represents the number of attributes. This approach models the instances as a binary tree that divides the input space with a hyperplane and then splits each partition again, recursively. A split is only made vertically or horizontally.

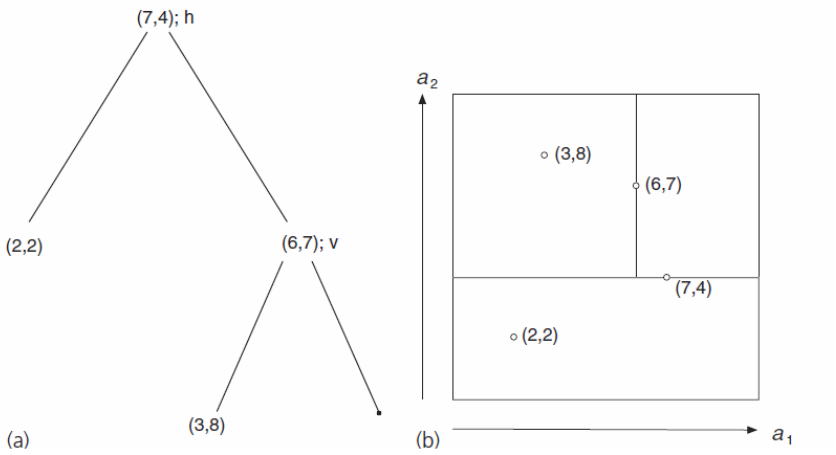


Figure 2.12: A simple kD-tree with four training instances: (a) the tree and (b) the instances and splits on a 2D plane. [164]

Figure 2.12(a) gives a simple example with k = 2, and Figure 2.12(b) illustrates the training instances on a two-dimensional plane, the hyperplanes that make up the tree are also visible here. The hyperplanes are not decision boundaries, the decision criteria is made on a nearest neighbour basis and is explained below. In this simple example, the first split is horizontal (h), through the point (7,4) this is the tree's root. The left branch is not split any further. The branch on the right side is then split vertically (v) at point (6,7). There is no point on the left child, but the right child has the point (3,8). As can be seen from this example, every region holds a single point or no points.

This approach speeds up Nearest Neighbour calculations as it can locate

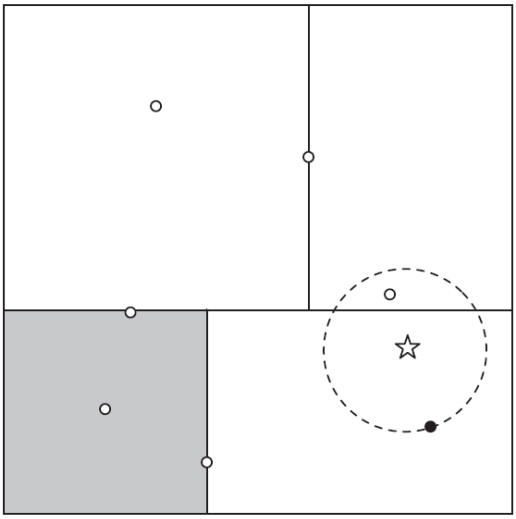


Figure 2.13: How to find the nearest neighbour using the kD-tree

the nearest neighbor of a given target point by following the binary tree from its root to find the region containing the target. Figure 2.13 shows a space similar to that found in Figure 2.12 with several additional instances and an extra boundary. In Figure 2.13 the target is denoted as a star and is not one of the instances in the tree. The black node is the leaf node in the tree which is inside the same region as the target. Having located the leaf node inside the same region as the target, we need to establish if there are any other nodes in another region which are closer to the target. Therefore a node which is closer will be inside the dashed circle as shown in Figure 2.13.

The shaded region in Figure 2.13 is the black nodes sibling, but the circle does not intersect it, so this sibling does not contain a nearest neighbour. This approach then backs up to the parent node in the tree and checks its siblings (which is everything above the horizontal line in Figure 2.13). Here the circle intersects the region so it needs to be explored. To explore this space, we find both daughters of the current node and calculate if either of these are located within the circle (the right point intersects the circle) and if any do then this node which becomes the nearest neighbour. In practice

this approach is far more efficient at examining all points to find the nearest neighbour.

This type of IBL operates very effectively and every attribute has the same level of influence on the final decision. However noisy exemplars can cause incorrect decisions and one way to counteract this is to adopt the k-nearest neighbour strategy, where a small number of k nearest neighbours are located and used to form a majority vote of what class the test instance correctly belongs to. In our research we implement Weka's Instance Based Learning classifier, IB1, which is a variation of K nearest neighbour.

2.6.6 Bayesian Networks

The second classification system which is used in this thesis is the widely used Bayesian Network. A Bayesian network will graph the model of probable relations in a set of probabilistic relationships within a set of variables. As of late, Bayesian Networks have become one of the most used classification techniques for inferring new knowledge. The techniques used are still advancing as researchers strive to find more accurate ways to infer new knowledge [70]. The principle of Bayesian Networks is to use graphical models to represent knowledge. Algorithms which are optimised for searching these graphs to find patterns are then utilised. The following section introduces the principles of a graphical model.

Probabilistic Graphical Models

This approach is based on the concept of declarative representation. Using this technique we build a computerised model of the system which we would like to reason. This model translates our knowledge of how our system operates into a computerised form. This computer readable form can then be interpreted by various algorithms that can provide answers to many ques-

tions. For example, in a medical diagnosis application, there are many different possible diseases that a patient could have, many different symptoms or diagnostic examinations, personal characteristics which may be symptomatic of specific diseases and other circumstances to factor in. These domains can be characterised by a set of *random variables*, where the value of each variable defines a property of the domain. Then when we are provided with some observations about some of these variables, our goal is to reason probabilistically about the values of other variables. To achieve this using principled probabilistic reasoning, we build a *joint distribution* over the possible outcomes of a set of specific random variables. A reasoning algorithm can take this model as well as some observations relating to a specific patient and answer questions relating to the specific patients problems.

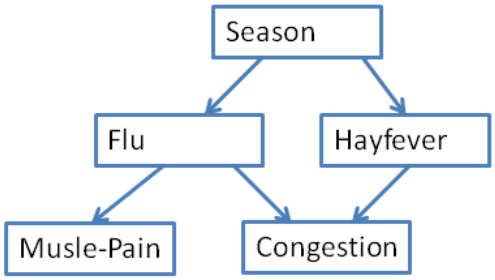


Figure 2.14: A sample Bayesian Network

To efficiently encode a complex distribution over a high dimensional space, probabilistic graphical models use a graph based structure. The graph in Figure 2.14 shows where the nodes represent the variables and the arrows represent the direct probabilistic connections. In this graph, it can be learned that there is no direct relationship between congestion and season, the two interact with flu or hayfever. One usage of this graph is that it can be used to represent a state of independencies that hold in the distribution. For example the statement:

$$P(Congestion|Flu, Hayfever, Season) = P(Congestion|Flu, Hayfever) \tag{2.6}$$

If we are interested in the distribution of the patent having congestion and we can tell from the graph that she has Flu, or that she has hayfever, then the value of season is not relevant (as can be inferred from Figure 2.14)

The second usage of the graph is that it defines a tree like structure for efficiently presenting a high dimensional distribution. In this sense, we can avoid having to calculate the probability of each relationship between each variable as some variables are unrelated to others. This helps to breakdown the distribution of the full joint distribution into a smaller product and this in turn increases the efficiency of computational calculations.

In essence, a Bayesian Network is learnt by defining two components, (1) a function for evaluation of a specific network which is based on the data and (2) a method of searching through the networks. Learning Bayesian networks involves lots of counting. A search will have a number of network structures, and to obtain the conditional probability tables the data needs to be scanned to obtain the counts needed. However counts can be stored efficiently using the all dimensions (AD) tree , which is an analogous approach to the kd-trees search, discussed in Section 2.6.5.

A Bayesian Network is extremely useful at classifying instances where the training set is quite sparse, as is the case with event detection in tennis, where the number of training instances will only run into several thousand at the most. For this reason and because Bayesian Networks have been widely employed in visual classification systems which have a similar scope to this research, Bayesian Networks have been chosen as one of the machine learning approaches used in this research.

2.7 Conclusion

This chapter outlines the main research components which are related to this work and used at varying levels throughout the following chapters in this thesis. Visual feature extraction is a large field and the different approaches described in Section 2.2 are all used to recognise human actions and activities in this thesis. There are of course different approaches, which can be used to detect a human subject in a visual scene, such as optical flow, but as has already been pointed out, foreground extraction is well suited to indoor environments where the background does not change often and there is few foreground objects.

One problem however is that artificial lighting creates shadows and these will generally become part of the foreground if the threshold is set low. In order to combat this problem a shadow removal module is applied to each frame in order to create a perfect silhouette of the human subject in the scene. Without a robust shadow removal technique, it will become difficult to classify the humans activities from the features extracted. Foreground extraction is not a perfect approach and to obtain a single foreground blob (which represents the human subject), we use two vital computer vision techniques called erosion and dilation, which help to remove noise from foreground objects.

Inertial sensors were introduced in Section 2.4 and an explanation of what types of sensors are generally found in an inertial node is given. In this work, accelerometers, magnetometers and gyroscopes are used and a description of each is given in this section. The final part of this section explains what is meant by a wireless inertial measuring units (WIMUs) and the technical specifications of one of the WIMU devices which is used to capture human motion is explained.

Finally, this chapter introduced the main concepts of machine learning,

a state of the art technique used for artificial intelligence. It is beyond the scope of this thesis to cover the complete theory of machine learning, instead we briefly introduced the main concepts of the two classifiers used in this work (Bayesian Networks and Instance Based Learning), each of which is used to recognise human actions using visual or/and inertial sensors in the following chapters.

Part II

Human Action Recognition

Chapter 3

Human Action Recognition with Video

3.1 Introduction

Human action recognition (HAR) is one of the most active research areas in computer vision today. The growing relevance is spearheaded by a wide range of potential applications in numerous areas such as automatic video event indexing, video surveillance and visual analysis of an athlete's performance for example. The earliest recorded investigations into human motion analysis date back as far as the 1850s, when contemporary photographers E. J. Marey and E. Muybridge photographed moving subjects and reported interesting and artistic aspects relevant to human and animal locomotion [136] (Figure 3.1).

Johansson's pioneering moving light display (MLD) experiment is one of the earliest works for studying and analysing human motions in the field of neuroscience. In essence the problem can be summarised as, given a succession of images with a person or persons performing an action, can a computer system be developed which will automatically recognise what

action is being performed? The question is quite simple but in practice the solution to this problem is extremely difficult.

It should be noted that human actions and human activities are differentiated in the context of this thesis. When we mention human actions we are referring to single motion patterns which are performed by a single person and are usually executed within a short duration of a few seconds. Examples of human actions include sitting, running, jumping or walking (which is shown in Figure 3.2). However, human activities refer to multiple complex actions which can be performed by one or more persons who are interacting with each other to achieve a certain task. These tasks generally take a much longer duration to complete. Examples of human activities could include team based analysis, where a football team score a goal or a group of people robbing a bank (Figure 3.3). There are significant difficulties in automatically detecting and understanding semantics from human activities. However, action recognition as targeted in this thesis could prove to be a useful starting point.

In this chapter we examine existing challenges and literature for using visual sensors in HAR. We then evaluate two common approaches for recognising human actions from video. Both approaches are evaluated on a near-field camera and also on an aerial view outdoor camera, whereby we then conclude which approach is the most suitable for HAR in this work.

3.2 Research Challenges

Automatic detection of humans in a visual scene and recognition of human activities and actions thereafter is currently a very active research field due to the many challenges which make this problem difficult to solve, as outlined in the following.

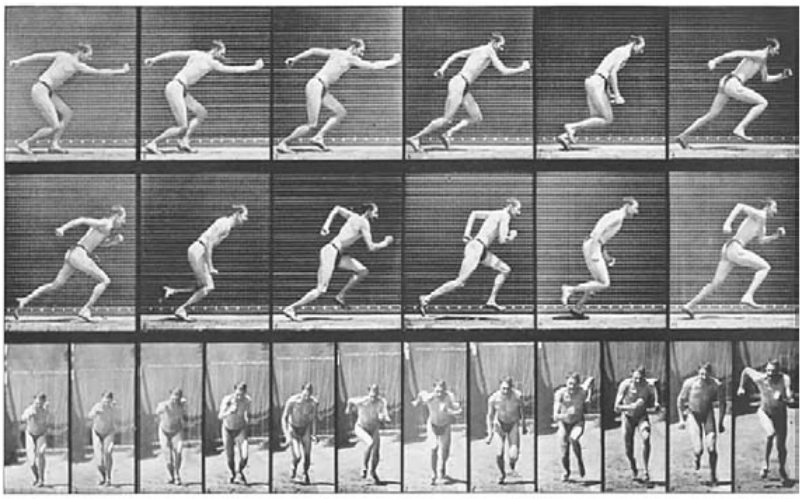


Figure 3.1: The Human Figure in Motion, Muybridge 1878.

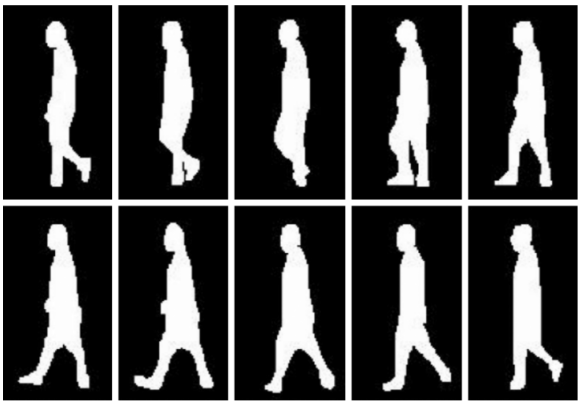


Figure 3.2: Example of a simple human action captured over a sequence of images, in this case the subject is walking [155].

Figure 3.3: Indoor video view. This video shows an acted back robbery [59]. (a) Subject walks into the bank. (b) Bank robber is recognised to be an outsider. Bank robber enters the secured safe. (c) A person walks out of the bank. (d) Bank robber leaves the bank.

- **Wide variety of articulated poses** By pose, we mean a specific position that the human body can take and a human action is a consecutive list of human poses. A human subject can perform a large number of different poses and will even perform multiple poses simultaneously. Example poses include running, walking, sitting or boxing and examples of simultaneous poses include walking while waving, or pointing an arm in a particular direction whilst sitting down.

- **Variable appearance and clothing** No two humans will dress the same and the visual appearance of two people based on height and build will rarely have similarities, which makes the job of training a model to detect generic human actions more difficult.

- **Complex Backgrounds** Identifying a human subject in a changing or outdoor background is often difficult. Changes in the background make an automatic human extraction process more challenging as moving background objects can be confused with the human subject.

- **Unconstrained Illumination** Capturing video indoors or outdoors can be challenged by changing illumination. In either case poor lighting may be present, or the scene may undergo significant lighting change.

- **Occlusion, different scales** Self occlusion is a significant problem in wide area human action recognition with video where humans are free to move around substantially in the scene and the camera will not capture their full body.

Given the potential applications, the scientific problems which exist, the performance and cost of current hardware and the focus on security issues today, huge efforts have intensified to push forward research on human motion detection. The intensified drive forward is obvious by observing the increasing number of conference papers and journals which are linked to this field, especially in the surveillance area [118].

The video based human action recognition system which is detailed later in this chapter, is initially employed in an indoor surveillance environment to detect human actions and thereafter in an outdoor aerial view scenario. Section 3.5 describes the two approaches used for human action recognition, (1) contour features and (2) Histogram of Gradient Orient Of Motion History Images (MHIHOG). Experiments in Section 3.6 indicate how well each method performs at recognising basic human actions. This comprehensive set of experiments helps to underline which approach is the most suitable for recognising human actions under the different scenarios with which this work is concerned. There are of course, many different approaches in the literature for detection of human actions using visual sensors and some of the more relevant approaches are described in the following section.

3.3 Related Work

In [124], the authors recognise human actions by using a collection of spatio-temporal events which are generated by image sequences and localised at points that are significant in space and time. Spatio-temporal salient points

are extracted by calculating the variance in the data of pixel neighbours in both space and time. A measure of the distance between both sets of spatio-temporal salient points is calculated using a method which is based on Chamfer distance [24]. Liu et al. [105] propose to generate a semantic bag of video words using sample videos with Pointwise Mutual Information and diffusion maps. Spatio-temporal features are extracted from the actions and after feature quantization the actions are represented by a semantic bag of words. The training videos are converted to a bag of semantic words and a Support Vector Machine (SVM) is used to build a classification model of the training videos. An input action then under goes the same transformation process and the unseen video is converted to a histogram of semantic words. The classifier decides to which action the unseen video is most likely to belong.

Motion History Images, which were described in the previous chapter, have been commonly used to detect basic human actions in the past, however they struggle to accurately represent complex human motions. The authors in [21] use a method for representing motion in successively layered silhouettes that directly encode system time in what is called the timed Motion History Image (tMHI). This representation can be used to both determine the current pose of the object and to segment and measure the motions induced by the object in a video scene. These segmented regions are not "motion blobs", but instead motion regions naturally connected to the moving parts of the object of interest. The method is used to recognise waving and overhead clapping motions to control a music synthesis program. [77] et al. propose a novel method which calculates the histogram of oriented gradient (HOG) of a motion history image (MHI). Their algorithm first generates a MHI with differential images, essentially the result of frame differencing over each image which captures the human action. The second step com-

putes the HOG of the MHI and then a SVM is used to train a classifier with the HOG features. This step does not require the human to be extracted as a silhouette, which increases the overall performance.

Human actions can be represented as a series of postures over time in a 2D scene and a commonly used approach for representing posture is to use its boundary shape [73]. A comprehensive review of current approaches used to detect human actions using one or more cameras can be found in [73]. Since each border point in a digital image is similar to its neighbor point, it is inefficient to use the whole human contour to describe a human posture. There are, of course dimensionality reduction techniques such as Principal Component Analysis [151], which can reduce the redundancy, but these approaches are computationally expensive due to matrix operations. In contrast to high dimensionality, simple information like the X/Y variance of the human posture do not provide enough information to give enough information to recognise a large number of basic human actions. However *Contour Features*, overcome these issues and have been shown to be accurate in detecting human actions.

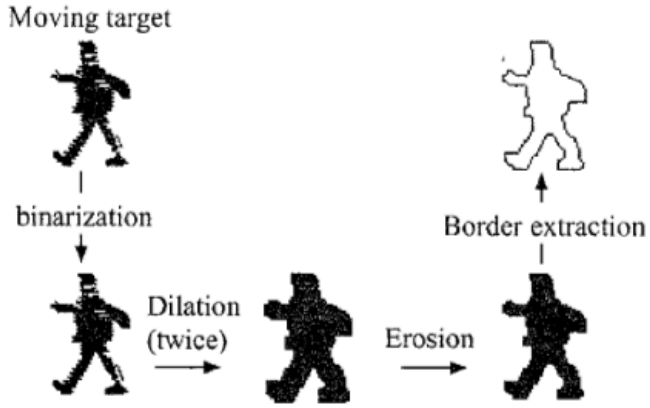
Previous approaches which use contour features include Fujiyoshi et al. [56], who use a process for analysing the motion of a human target in a video stream. Moving targets are detected and their boundaries extracted by extracting the human as foreground, using foreground extraction. From the foreground images a skeleton of a human is formed as shown in Figure 3.4, taken from [56]. Two features of motion are identified from the skeleton, the posture and the repetitive movements of the skeleton. Both cues give clues to human actions such as walking or running. This method has proven useful and it is not necessary to build a priori human model when employing this method. In addition, the computational cost is low, and it is an appropriate solution for practical deployments. One issue with this approach is that the

human actions need to be relatively simplistic.

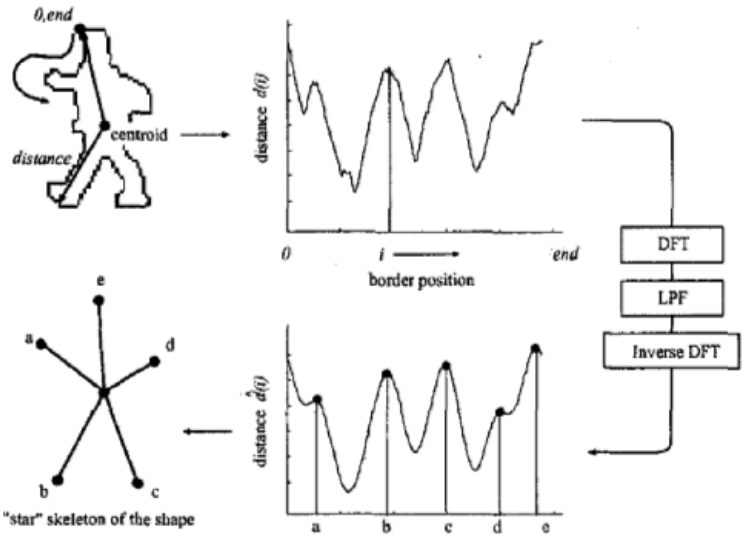
The authors in [27] also use image skeletonisation to recognise basic human actions from a near view video. This action recognition method extracts features from the human motions using star skeleton for recognition and these features are then modeled using Hidden Markov Models. Each human action is represented by a sequence of temporal images, which are transformed to an image feature vector using star skeletons from each image. Each feature vector of the sequence is allocated a symbol which matches a codeword in the code book using Vector Quantization [86]. Then the time-sequential images are converted to a symbol sequence. To train the system, the model parameters of the HMM of each category are optimised to give the best representation of the training symbol sequences for all categories of the human actions to be recognised. For human action recognition, the model which best matches the observed symbol sequence is selected as the recognised category.

3.3.1 Discussion

Contour features have been widely used to recognise human actions as the previous section illustrates. The advantages of using this approach is that it is computationally inexpensive, it gives a useful representation of the human silhouette and different vector ranges can be easily implemented to reduce or expand feature dimensionality size. In this thesis, we evaluate both MHIHOGs and Contour Features. MHIHOGs use Motion History Images which utilises motion shape information of a video to recognise actions. The advantage of MHI is in its simplicity and low computational cost compared to the optical flow method, for example. Moreover HOG are known to be a very accurate technique for representing movement in video [44] [22] [77].



(a) The outermost boundary pixels are identified by calculating the distance from the center of the object to the edge.



(b) In a preprocessing step, morphological erosion and dilation is applied and then the border is extracted

Figure 3.4: Image Skeletonisation [56].

3.4 Human Motion Segmentation

Motion segmentation is a process which detects pixel regions which correspond to moving objects. These mobile objects may potentially be targets in images or video, which can be used for later processes such as activity recognition and object tracking [76]. Human motion detection is the process of recognising people in a captured scene and extracting the human subject's motions over time. The foundation of an effective human motion analysis system requires accurate pose estimation and action recognition systems and these processes are heavily reliant on an efficient human detection system [73]. It is difficult to obtain high-level human action recognition without successful motion segmentation. The two main approaches for motion segmentation are foreground extraction and optical flow. In this work, the former approach is adopted. Both of these approaches have been introduced in the previous chapter.

Foreground extraction is a commonly used technique for motion segmentation and is very effective in scenarios where there is a relatively static background. In this research we use foreground extraction for motion segmentation because it can generate efficient segmentation results in scenes where the background is fairly static but the lighting changes over time, as is the case in our application. Furthermore we chose foreground extraction, over optical flow because the latter is vulnerable to image noise, colour and changing lighting and has a significant computational cost.

Frame Differencing

The visual datasets used in this work contain a series of human actions performed by various people in both an indoor and an outdoor environment. In both datasets, each human subject stands idle before each action is performed. After empirical analysis it was concluded that no action in this

dataset exceeds exceed 4 seconds, so we can simply extract 4 seconds of video every time we detect the beginning of a new action. This fixed window length can easily be adjusted for different datasets in future work. Using foreground extraction we detect the beginning of a new action by analysing the video and recognising this idle stance through low motion activity. When the subject moves and this motion is above a fixed threshold (20% of the human silhouette), we record this time as the beginning of a new action. Different threshold sizes were tested but 20% is the most accurate for detecting human movement in this dataset and again this threshold may be adjusted for different datasets in future work.

To assess this action boundary detector, we developed an evaluation experiment which measures the accuracy of this approach. For this experiment we applied our detection algorithm to detect all the actions of five human subjects. We applied our frame differencing algorithm to each subject in turn and where the beginning of a new action was detected to within 1 second of the actual start of an action it was deemed to be the correct. Accuracy to within 1 second is sufficient for extracting actions features. This is because in this dataset no action is shorter than two. Each subject performed seven different actions and each action was executed ten times giving a total of 70 actions per human subject. The overall precision and recall for detecting the beginning of a new action was .89 and .93, which is deemed to be sufficiently accurate.

3.5 Feature Recognition Methodologies

The two approaches used to detect human actions in this thesis are *Histogram of Oriented Gradient of Motion History Images (MHIHOG)* and *Contour Features*. In the next two sections we explain how each approach is implemented to detect human actions. As is the case when training any machine

Figure 3.5: Indoor camera view

learning system, providing the classifier with features which give a strong generic representation of the class in question is vital to obtaining sufficiently accurate results. Several preprocessing steps are necessary to obtain visual features which will help the classifier in the training and testing stage.

3.5.1 Histogram of Oriented Gradient of Motion History Image

This approach is based on Motion History/Energy Images, which was discussed in Section 2.2.6 and is a commonly used approach to detect human actions by representing human motion using temporal templates. Our preliminary research into Motion History Images (MHI) has concluded that this approach is most productive in representing basic human actions. More complex actions tend to obstruct information related to the beginning of the movement, which makes this approach unsuitable for complex human actions such as boxing or running, where there is repetitive limb movements. An example of this problem can be in rapid movement actions such as jogging or boxing where the rapid movement of the arm forwards and backwards obscures some of the visual detail captured by later frames. Another challenge related to using traditional MHIs is how to best represent the information

from the resulting images. In [45], the authors used Hu Moments [75] to represent basic actions and these are known to give good representation in a scale invariant and view invariant manner.

To mitigate against these drawbacks, we implemented a method based on MHI and Histogram of Oriented Gradients (HOG), called MHIHOG. The method was first described in [77], however our approach is a variation of that detailed by the authors in [77] and has several differences. One significant difference is that we normalise the HOG features so that the features from human subjects of different sizes can be compared. The second difference is that we train our features on an Instance Based Learning classifier to compare which gives the most accuracy in classifying the HOG features. The overview diagram of our proposed method is shown in Figure 3.6. Our approach also obtains a HOG of the foreground MHI, unlike the method detailed in [77].

To generate a motion history image, our approach processes four images per second. Investigations were conducted for different sampling frequencies but our tests concluded that due to the diversity in the actions in the dataset, we obtain better classification results with a low sampling rate. When MHIs are generated for each action a HOG is used to extract features from the motion history image. The first step in this method creates a MHI with frame differencing which represents the action. The second part computes the HOG of the MHI. In our method, we adopt the approach introduced by [77] and train a classifier to learn the actions using the HOG features of the MHI.

3.5.2 Contour Features

Contour features are commonly used to identify basic human actions from a 2-D scene [35] [56]. The concept of contour features is to connect from

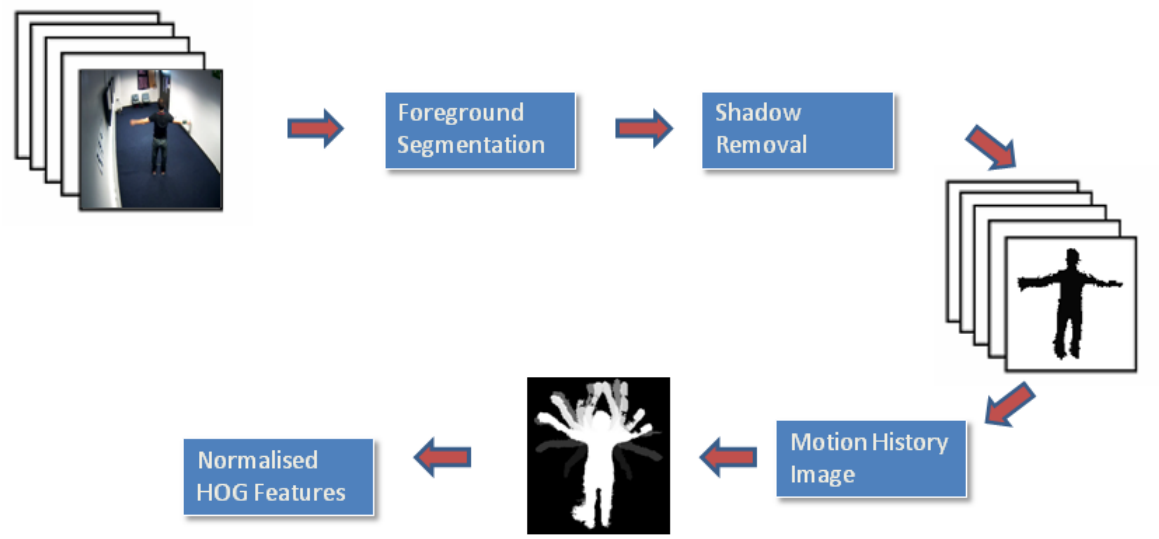


Figure 3.6: Overview of the process of extracting Histogram of Oriented Gradient of Motion History Images

the center to the peripheral extremities of a human contour as illustrated in Figure 3.7. After segmenting each action into four second video clips using frame differencing, we segment the video into 120 frames in total (30fps). For each frame, we extract the human subject using foreground extraction and the resulting human silhouette is then sliced into pie segments (16 in this case). Then the distance from the centroid of the human foreground to the furthest foreground pixel along the pie line is calculated in a clockwise or counter-clockwise order. The end result is that for each image we obtain is 16 features which give a good representation of the human posture. Contour features are calculated for each image in the action sequence.

3.5.3 Action Classification

Using an Instance Based Learning, a model can be generated for each of the actions that will be recognised. Instance based learning classifiers have proved suitable for our needs and the time taken to generate classification

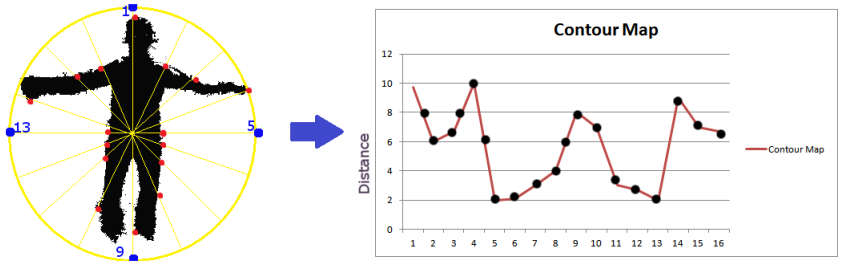


Figure 3.7: Extracting contour features from the human silhouette

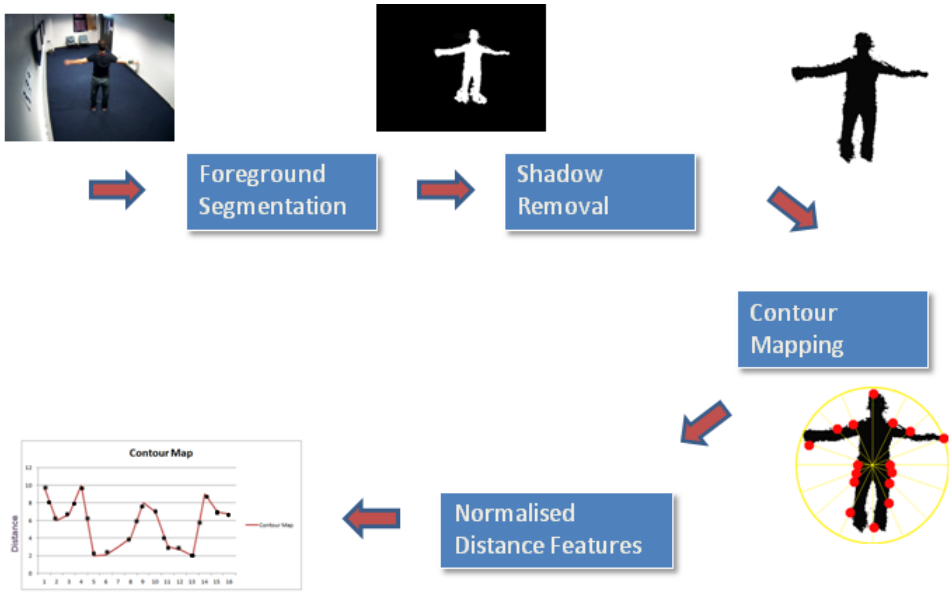


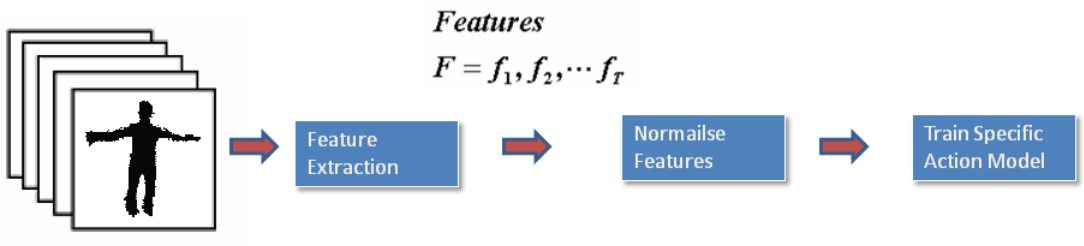
Figure 3.8: Overview of extracting Contour Features

models is only a few seconds so we do not perform any linear or nonlinear dimension reduction techniques. In both feature recognition approaches we normalise the features to compensate for height variances between different human subjects. The normalisation step scales the features so that the variance is equal to 1. By normalising the features the classifier will be more accurate at finding similarities between the actions performed by people of different sizes. Each human subject is wearing different colours of clothing but this does not cause a conflict when using foreground extraction, which is in contrast to optical flow, which can sometimes be adversely affected by different colours of clothing.

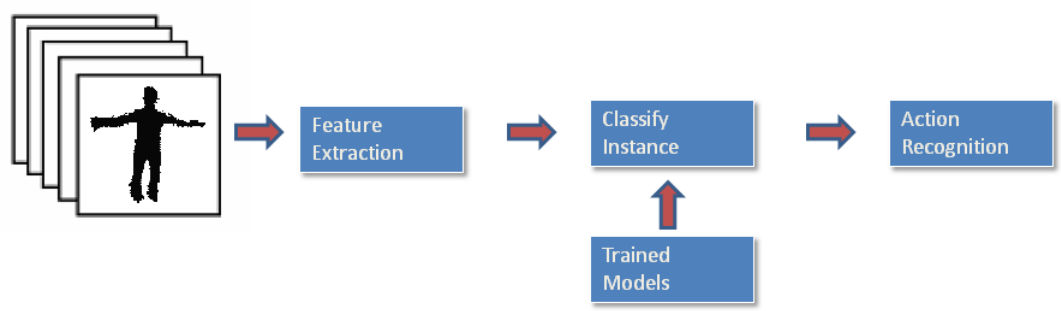
Once the training model is learned as per the method in Figure 3.9(a), an unknown input action is tested against the training model. Figure 3.9(b) shows the workflow used to test each input action against the training model. The classifier then predicts which action from the training set is most similar to the unknown input action.

3.6 Indoor Action Recognition Experiments

In this section we provide evaluation results of each Human Action Recognition (HAR) approach. The first experiment is to determine the accuracy of each method in detecting human actions from a near-field rear view camera in an indoor environment. This rear view angle is similar to the baseline camera angles used in Section 6 to recognise tennis events. The indoor data capture contains seven distinct human actions (crawl, walk, jacks, jump, boxing, wave, sit on floor) and details of this dataset can be found in the following section. Experiments were conducted on a computer equipped Intel Core 2 Duo Processor and 3GB of Random Access Memory. The computer operates on Windows 7 and the software packages used were Matlab R2009a and Weka 2.7.



(a) Training Workflow



(b) Testing Workflow

Figure 3.9: Action Recognition Workflow

3.6.1 Indoor Human Action Recognition Dataset

The dataset used in this experiment was captured with a single camera (640×480). At any time there is only one person in the scene. For the indoor human action dataset, seven actions are performed ten times, by ten people. The indoor actions performed were crawl, forward walk, jack, jump, boxing, single wave, sit on floor. The camera captures each indoor action from a rear view camera as shown in Figure 3.10.

Each person performed each action ten times and the subject kept their arms by their sides when not performing an action. The human subjects wore different clothing, were of different gender and body size to provide sufficient variety. Figure 3.10 illustrates how each action was executed by a variety of human subjects.

A single Wireless Inertial Measuring Unit (WIMU) was also attached

(a) Boxing

(b) Crawl

(c) Jack

(d) Pick Up

(e) Sit On Floor

(f) Walk

(g) Wave

Figure 3.10: Human actions captured on an indoor camera

to a human subject's right forearm so that the same dataset could be used for HAR based on inertial sensors, containing a tri-axis accelerometer, gyroscope and magnetometer. It should be noted that seven different actions were selected as this is more actions than any reported in existing works. Most visual action recognition datasets do have nine actions but nine actions would give an unfair bias when comparing the accuracy of action recognition between visual sensors and inertial sensors. This bias would be because action recognition using inertial sensors is not as well developed as visual sensors. The specifications of the WIMU used in this experiment can be found in Section 2.4 and the data from the inertial sensors is examined in Section 4.

3.6.2 Results

The first experiments in this section use the "leave one sequence out" cross validation approach to assess the accuracy of the classifiers. This approach means that one sample was omitted from the dataset and the remainder of the dataset was used to train the classifier. The omitted sample is then used to test the classifier and the result is recorded. Then the process repeats itself until every sample action has been tested. Once all the samples are tested, a confusion matrix showing the percentage accuracy of each action is generated, as shown in Table 3.1.

Using an Instance Based Learning classifier, the MHIHOGs provide a cross correlation accuracy of 92%. Table 3.2 shows the confusion matrix for contour features using leave one out cross correlation. For contour features the overall accuracy is 94%, which only contains a small number of incorrect classifications of the 480 samples tested. Therefore the conclusion of this experiment is that contour features are the most suitable for event detection of basic human actions from a single camera.

	Crawl	Walk	Jacks	Jump	Boxing	Wave	SitonFloor
Crawl	.96	.03			.01		
Walk		.95	.01				.4
Jack		.02	.90	.01		.07	
Jump		.01	.03	.88	.02	.04	.02
Boxing				.04	.94	.02	
Wave			.01	.01		.95	.03
SitonFloor		.04		.02	.01		.93

Table 3.1: Confusion matrix showing the accuracy of recognised actions for **MHIHOG** features on seven actions performed by 10 subjects in an indoor environment using "leave one sequence out" cross correlation and classified with Instance Based Learner (IB1)

	Crawl	Walk	Jacks	Jump	Boxing	Wave	SitonFloor
Crawl	.98			.02			
Walk		1					
Jack			.88	.04		.08	
Jump			.04	.92	.02	.02	
Boxing			.03	.03	.88	.06	
Wave			.02		.02	.96	
SitonFloor					.02		.98

Table 3.2: Confusion matrix showing the accuracy of recognised actions for **Contour** features on seven actions performed by 10 subjects in an indoor environment using "leave one sequence out" cross correlation and classified with Instance Based Learner (IB1)

Instance based Learning vs Bayesian Networks

When we use a Bayesian Network to classify the same features the MHIHOGs give an accuracy of 85% and the contour features give an accuracy of 87%, proving that a lazy learner algorithm such as IB1 [2] classifies visual human actions more accurately than a Bayesian Network.

Leave One Subject Out Classification

For this experiment, an IB1 classifier is trained on a random five human subjects and then tested on an unseen person, otherwise known as *leave one subject out*. This experiment assesses whether classifiers can be tested on unseen human subjects. A leave one subject out approach is used to train a model for each action, i.e. the training set contains no samples of the human subject being tested. The Leave one subject out approach is an efficient method to gauge if the classifier is accurate at classifying instances from unseen human subjects and therefore not biased.

	Crawl	Walk	Jacks	Jump	Boxing	Wave	SitonFloor
Crawl	.68			.22		.10	
Walk	.33	.56		.11			
Jack		.11	.78		.11		
Jump			.30	.70			
Boxing					.60	.40	
Wave			.33			.67	
SitonFloor			.20				.80

Table 3.3: Confusion matrix showing the accuracy of recognised actions for **MHIHOG** features on six actions performed by 10 subjects in an **indoor** environment using leave one subject out and classified with IB1

The overall accuracy obtained using MHIHOG is 68%. Furthermore, the confusion matrix illustrates that this approach does not perform well when the actions involve repetitive limb movements such as those found in walking or boxing. In contrast MHIHOG does perform well when the human actions are basic in nature such as sitting on the floor or performing jacks.

For our second experiment we apply contour features to the same data set to determine which feature recognition methodology is the most accurate at classifying human actions.

	Crawl	Walk	Jacks	Jump	Boxing	Wave	SitonFloor
Crawl	.97					.03	
Walk	.20	.75		.03		.02	
Jack	.36		.50			.14	
Jump	.10			.85	.05		
Boxing	.01			.23	.74	.02	
Wave	.08			.02	.18	.72	
SitonFloor	.03				.03		.94

Table 3.4: Confusion matrix showing the accuracy of recognised actions for **Contour** features on seven actions performed by 10 subjects using LOSO and classified with IB1

The overall accuracy when using contour features is .78, which is a significant improvement over MHIHOG features. However contour features use 1920 features per action, which is significantly more than the 80 features which the MHIHOG generates. This means there is extra processing involved with contour features but we consider this cost effective when it comes with a 10% increase in accuracy. In conclusion, we have found contour features to be significantly more accurate for HAR from a near-field indoor camera.

3.7 Aerial & Wide View Action Classification

It is not always possible to have the camera in a convenient near view position for recognising human actions or activities. Wide area view and aerial view cameras have also been used to detect actions and in either of these views human action recognition is significantly more challenging. In Section 6, an aerial view camera is used to detect tennis events and to demonstrate that aerial view action recognition is possible for detecting more traditional human actions, this section now provides accuracy results for detecting hu-

Figure 3.11: Aerial view of a variety of human actions [26]

man actions from a single aerial view camera.

3.7.1 Related Work

It is common to film from a distance when it is not physically possible to set up a camera at close range, which is the case for many military and surveillance operations [156] [26] or field sport analysis [173] [51]. Human actions can be detected using a single view camera or using multiple views in a distributed environment [25].

Chen et al. [26] recognise human actions from cameras where the size of the human subject may be as small as forty pixels, which is common when the actions are recorded from afar. There are several issues with this technique. The first is that the image resolution is reduced and the quality of visual information is reduced. They report that it is challenging to detect "waving" from "walking" using a single motion classification technique and therefore detect consecutive poses as Histogram of Oriented Gradients (HOG) and actions are represented by a sequential number of Histogram of Oriented Optical Flow (HOOF). Supervised Principal Component Analysis (SPCA) is then employed to reduce the feature size of the histogram vectors.

As shown in Figure 3.11, the human subject's limb in this scene is only four pixels and the edges between the human and the background is not well defined. In [26], the authors report that these problems result in an optical flow which contains a lot of noise. To overcome this issue a novel descriptor is proposed in [26] which joins human poses and movement data within a

Figure 3.12: Walking action taken from our Aerial View Dataset

spatial-temporal volume. They use Histogram of Oriented Gradients (HOG) to represent the human poses. In the work presented here, we investigate the usefulness of our previously described approaches in this scenario. This is motivated by the goal to have a single approach that works across both scenarios.

3.7.2 Aerial View Human Action Dataset

To test our action recognition approach we created a dataset of human actions captured from an aerial view in an outdoor environment. The actions performed were boxing, clapping, one handed waves, two handed waves, jacks and walking. A fixed camera was positioned approximately nine metres above the street and captures the scene of the street below as shown in Figure 3.12. Eight people participated in the capture and each performed each action ten times giving a total of 480 actions.

3.7.3 Experiments

In our first experiment we extract MHIHOG features for each action in the aerial dataset and then obtain a measure of the similarity between the 480

actions using leave one out cross correlation. The features are trained with an IB1 classifier and as Table 3.5 illustrates the overall accuracy is very high at 91%. It is unsurprising that boxing gets incorrectly misclassified as clapping since both these actions appear almost identical from an aerial view. In the second experiment we detect the contour features of each human action in the dataset and use leave one instance out cross correlation to detect the accuracy of different actions. The accuracy for contour features using an IB1 classifier in the aerial outdoor environment is 92%. The confusion matrix in Table 3.6 shows that very few actions are misclassified and similarly to HAR using an indoor camera, contour features are more accurate at HAR from an outdoor/aerial view camera.

	Boxing	Clap	2 Hand Wave	Jack	Walk	1 Hand Wave
Boxing	.92	.06		.02		
Clap		1				
2 Hand Wave	.02		.92	.06		
Jack		.02		.98		
Walk					1	
1 Hand Wave						1

Table 3.5: Leave one out cross correlation showing the accuracy of recognised actions for **MHIHOG** features on six actions performed ten times, by eight human subjects in an **outdoor** environment and classified with IB1

	Boxing	Clap	2 Hand Wave	Jack	Walk	1 Hand Wave
Boxing	.98	.02				
Clapping		1				
2 Hand Wave	.02		.98			
Jack		.10		.90		
Walk		.01		.01	.97	.01
1 Hand Wave		.02				.98

Table 3.6: Leave one out cross correlation showing the accuracy of recognised actions for **Contour** features on six actions performed ten times, by eight human subjects in an **outdoor** environment and classified with IB1

3.8 Weizmann Dataset Experiments

Finally, in this set of experiments we apply both the MHIHOGs and Contour features to the popular Weizmann dataset [63]. This dataset is a collection of 90 low resolution human actions (180×144, deinterlaced 50 fps) and contains nine people performing the ten actions one time each. The actions in this dataset are bend, jump forward, run, side walk, skip, walk, single handed wave, double handed wave, jumping jacks and jump in place. Table 3.7 shows our results for MHIHOGs and Table 3.8 shows our results for contour features. Each confusion matrix summarises the recognition rates for "leave one sequence out" cross correlation.

Instance based learning was used to classify the actions in this experiment. MHIHOGs achieve a recognition accuracy of 69%, while contour features achieve a higher accuracy of 84%. Contour features may use every frame possible but our experiments have discovered that using a fixed window of 25 frames per action achieves the highest accuracy. In fact, when a window of 40 frames per action was used, the accuracy dropped to 75%. Therefore the highest overall accuracy was 84% and this was achieved using contour features, with a window size of 25 frames per action and classified with Instance based learning.

3.8.1 Discussion

The reason why MHIHOGs do not perform so well in this dataset is because there are a high number of basic human actions to be detected and these features appear very similar amongst many actions such as walking and running, where a lot of information is lost with motion history images since not every frame is used.

In [63], Gorelick et al. report accuracy results of 96%, though the reported overall processing time required to extract features of a 50 second

pre-segmented video (110×70) is 30 seconds on a Pentium 4, 3.0GHz. In our approach, the overall processing time takes on average 8 seconds to extract contour features from a 50 second pre-segmented video (180×144) on an Intel Core Duo T2600 2.16GHz, which is a significant improvement on [63].

	Bend	Jump	Run	Side	Skip	Walk	Wave1	Wave2	Jack	Jump
Bend	1									
Jump		.67		.22		.11				
Run			.45	.11	.33	.11				
Side		.11	.11	.78						
Skip		.11	.44		.11	.34				
Walk		.22			.44	.34				
Wave1							.89	.11		
Wave2	.11						.11	.78		
Jack	.11				.11			.11	.67	
Jump										1

Table 3.7: Confusion matrix showing the action recognition for **MHIHOG** features on the Weizmann dataset using leave one out cross correlation and classified with IB1

	Bend	Jump	Run	Side	Skip	Walk	Wave1	Wave2	Jack	Jump
Bend	.89			.11						
Jump		.78								.22
Run			.56	.11	.11	.22				
Side				1						
Skip			.33		.45					.22
Walk				.11		.89				
Wave1							1			
Wave2								1		
Jack									.78	.22
Jump										1

Table 3.8: Confusion matrix showing the action recognition for **Contour** features on the Weizmann dataset using leave one out cross correlation and classified with IB1

3.9 Discussion

Two approaches for detecting human motion were investigated and evaluated against different datasets, in different environments, of different sizes and different resolutions, where the number of test subjects varies in different tests. Applying each action recognition method to differing datasets evaluates how the action recognition methods perform in different scenarios. There was no apparent trade-off between accuracy and frame resolution size, though most cameras today do have high resolution. Results indicate that contour features are significantly more accurate at detecting human actions and the results are very encouraging. The confusion matrix for MHIHOG illustrates how this approach has difficulty detecting actions which involve high motion. The confusion matrix also proves that MHIHOGs are best suited to simple motions, in much the same manner as motion history images perform best when the actions are simplistic. The issue with motion history images is when the human performs complex actions the features obtained are not discriminative. This approach introduces a lot of noise and visual kinematic information is lost, as was the case when the subjects are boxing or walking in our experiments.

However, contour features overcome these issues as information from each frame is processed unlike motion history images, where motion can cause overlap. Contour features are therefore the most suitable approach for HAR from an indoor, outdoor and aerial view camera. Of course, no single visual action recognition approach can completely overcome recognition issues such as self occlusion or complex backgrounds with clutter and ongoing research is seeking solutions to these problems. However, body worn inertial sensors can potentially overcome the challenges with which visual sensors struggle. The next chapter introduces an approach to action recognition using inertial sensors and uses the same dataset to detect which sensor is most appropriate

for human action recognition.

Chapter 4

Human Action Recognition with Inertial Sensors

4.1 Introduction

In this chapter we introduce a novel approach to detect human actions using a single inertial sensor worn by a human subject. One of the main premises of wearable computing is to enable personal applications that can adapt and react to the current context of the user. The term context is generally broadly defined and can in principle encompass any kind of information that relates to the current situation of the user or the objects surrounding him or her [48]. This section covers an introduction to inertial sensors and this is followed by an overview of the research challenges in this field. We then introduce our own approach for detecting human actions with inertial sensors.

Early work in activity recognition with wearable sensors can be found in the early nineties, when advances in hardware technology made sensors and hardware light enough to enable mobile computerised systems, which could be attached to a human subject for long periods of time (e.g. as described in

[146]). Although these research prototypes were still relatively bulky and a long way from "vanishing into the background" as envisioned by Mark Weiser [161], they held the exciting promise of making the computer perceive human life from a first person perspective, thus enabling truly personal applications. Early work centered on traditional, text and keyboard based applications, and then gradually explored new methods of input and interaction. An example of this scenario would be using wearable cameras [138] [147] or microphones [34] and incorporating other context information such as the user's current location, the subject of a conversation or the identity of a conversation partner in order to provide the user with relevant information about his current situation in real time, or to store information for later retrieval [133].

Measuring the physical activity of a person through the use of objective technology has been a longstanding goal of the medical research community, and accelerometers have been used for this purpose for several decades [119], [165]. These early systems aimed to estimate global measures such as the total energy expenditure or the oxygen requirement of the subject while he or she was performing a number of different activities. Mobile systems incorporating inertial sensors that could separate and recognize specific physical activities emerged at the turn of the last decade, stimulated both by advances in hardware technology, machine learning methods, and by their expected usefulness for the new paradigm of context-aware computing [61] [131] [99].

Existing research in human wearable action recognition sensors spans many areas, where some researchers focus on activity recognition in daily living for healthcare [102], automatic recognition of activities in unlabeled data [115], semi-automatic or unsupervised learning of activities [167] [81], or combining various sensor modalities to increase recognition accuracy [149]

[158].

4.2 Related Work

There are many different methodologies for recognising human actions and activities from raw inertial sensor data. Given the nature of inertial sensors, sensor data usually undergoes a pre processing step. In most cases, high frequency noise in acceleration data needs to be detected and removed. Techniques such as low-pass median [88], Laplacian [17], and Gaussian filters [94] can be used to remove high-frequency noise. In certain cases, gravitational acceleration needs to be removed from accelerometer data to help analysis of meaningful dynamic acceleration. To achieve this, high pass filters can be employed to recognise body acceleration from gravitational acceleration [168]. The ability to represent raw data, while conserving relevant information is vital for efficient recognition systems. This step can have a major effect to the overall success and computation time of activity recognition systems.

Sensors can capture enormous amounts of information and therefore it is necessary to find abstractions of the raw data via relevant features. The feature vector contains important patterns for identifying separate actions [168] [35] and these vectors are then used for classification. One such method of feature extraction is Fourier Transforms, which have the ability to hold the primary information, while reducing the dimensionality of the sensor data [69]. Discrete-Fourier Transforms are a specific version of Fourier Transforms that use discrete input functions like sensor samples [166].

Time-domain features can be signal statistics and basic waveform characteristics which are directly derived from a portion of the data. Ward et al. [159] attach a triaxial accelerometer and a microphone to the human subject's wrist to recognise human actions. They report that people commonly

attach devices to their wrist on a daily basis, making this sensor placement acceptable. In this work, they extract the mean, variance and count the number of peaks in each axis as the features to represent each action and can classify regular working man actions such as hammering, drilling and sawing etc. to a high degree of accuracy. Extracting the variance from raw accelerometer data has proven to give a high recognition rate of human actions and is a widely used approach [153] [128] [106] [160] [169]. In [128] , the authors report that variance of accelerometer features perform well when the sensor is attached to the person's dominant wrist.

4.2.1 Discussion

Filtering high frequency noise is a common signal processing step and in this thesis, we use a standard filtering technique. Correct sensor placement is a decision that needs careful consideration and attaching multiple sensors to a person will result in the person being uncomfortable which may effect their performance. For this work, we wish to attach as few sensors as possible to the human subject and existing research has shown that sensor placement on the wrist obtains very good results and also that people do not mind having wrist attachments.

For feature extraction, our review has concluded that variance and standard deviation of raw sensor data can obtain high accuracy for detecting human actions. This approach is also fast as it does not require a significant amount of computational effort.

4.3 Action Recognition Methodology

This section details our approach for recognising human actions using a single inertial sensor. The sensor is attached to the human subject's wrist. This placement is suitable for discriminating actions involving upper body

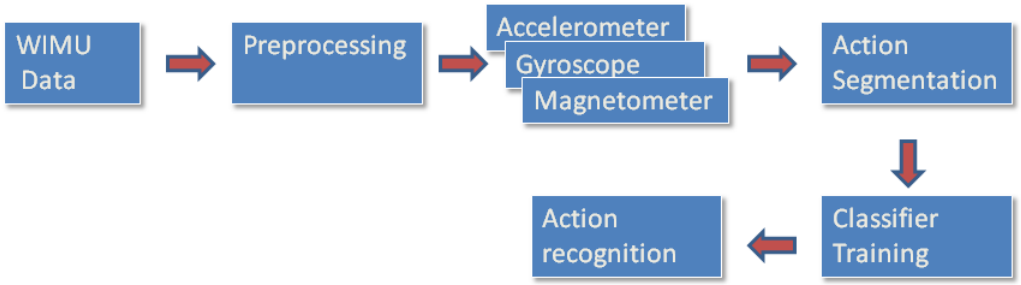
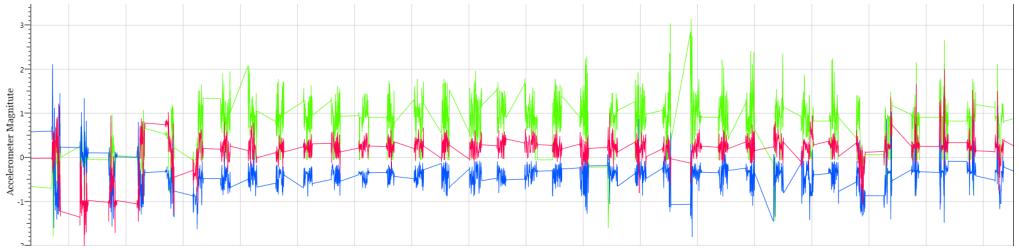


Figure 4.1: System Overview for Inertial Classification of Human Actions

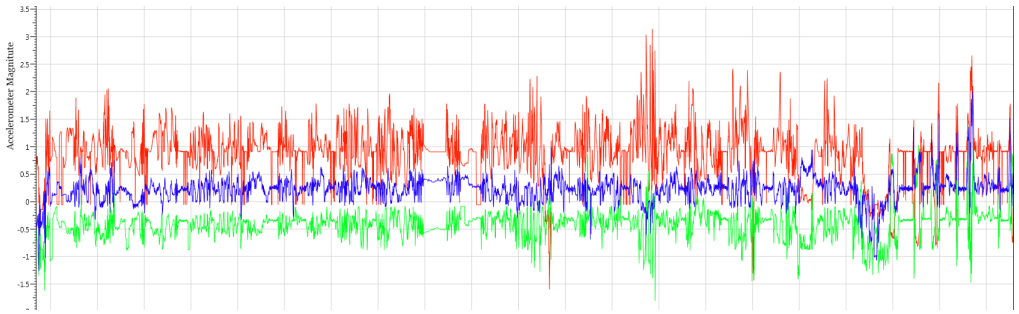
movements [10]. Using inertial sensors for this purpose overcomes many of the problems with visual sensors, such as scene clutter and view invariance.

4.3.1 Approach Overview

Figure 4.1 shows an overview of the process used to recognise human actions using inertial sensors. Data preprocessing is used to filter the data and temporal smoothing provides a constant sample rate. The data is then normalised to compensate for variances in force generated by different human subjects. The training data is generated using a ground truth which was manually annotated. All actions are tagged and a classifier is built which models each action. Test actions are identified by analysing the accelerometer data to identify motion. In this approach, the beginning of a new action is identified by recognising movement in a single axis of the accelerometer. Since the human subjects will be stationary between actions, monitoring accelerometer motion will help differentiate between activity and inactivity. The following sections provide more details of the various stages of this approach.



(a) Buffered Signal is received from wireless sensor



(b) Results of signal smoothing

Figure 4.2: Signal smoothing in (b) shows how buffered data is aligned equally in time.

4.3.2 Signal Pre-processing

Inertial sensors capture an abundance of data, but in order to obtain a clear signal which gives a good representation of the action being performed, several signal pre-processing steps are necessary. In this work, filtering, signal smoothing and normalisation are used to remove noise from the wireless inertial sensors.

Filtering, Smoothing and Resampling

Filtering high frequency noise from the sensor signal is an important step and in this work we employ a generic digital filter[1] to remove high frequency noise from the inertial sensor. The majority of inertial sensors today transmit data wirelessly to facilitate real time analysis. Inertial sensors generally transmit data in real time from the sensor nodes to a nearby base station.

[1] http://www.mathworks.co.uk/help/techdoc/ref/filter2.html

Figure 4.3: Example of inertial sensor attached to human subjects forearm.

However the raw sensor data can be transmitted in bursts to save battery life on the nodes, which is an approach used by the manufacturers of the inertial sensors used in this thesis. In this situation, however it is necessary to temporally smooth the data (after it is received by the base station), so that when buffered data is received, an even temporal distribution from all sensors is achieved. Essentially the data samples are evenly distributed over time. Figure 4.2 illustrates the process of smoothing the raw sensor data after it is received by the base station. Signal smoothing is applied offline to the accelerometer, magnetometer and gyroscope data. In this approach we inspect the delivery timestamp on each received sample and where the difference between two consecutive samples exceeds a threshold of .065, the timestamps of all samples since the previous signal smoothing are adjusted. The time adjustment At value added to each time is calculated as follows:

$$At = (t - pAt)/Ns \tag{4.1}$$

where Ns is equal to the number of samples which have elapsed since the previous time adjustment, t is the current time and pAt is the previous time adjustment.

After a smooth signal is obtained, the data needs to be resampled to a suitable range and this step is again achieved offline. This step is necessary as standard inertial sensors capture at a variable rate. The inertial sensor we use have a variable sampling rate (averaging between 100hz-1300hz). A uniform sampling rate per second will create feature vectors which give better representations of the actions being performed. In this experiment a sampling rate of $120hz$ was used, which provides a generous representation of an action while allowing the classification process to operate without the need to abstract the feature vector. For this resampling function, we simply extract all samples received in a given second and take 120 evenly distributed samples. Where the samples in a given second do not exceed 120, the array is padded with duplicate samples from the given second.

Normalisation

No two human subjects will typically generate the same magnitude and therefore it is necessary to normalise the signals of all three sensors to account for variance in actions performed by different subjects. A simple but powerful normalisation process is used, which normalises the sensor signal so that the mean is equal to 0 and the standard deviation is equal to 1. This approach is applied to accelerometer, gyroscope and magnetometer signals and helps the classification process to find similarities in different human subjects. Each action performed is then manually annotated and these annotations are required for training a classifier.

4.3.3 Action Segmentation

Determining regions of inactivity when analysing human motion with inertial sensors is a necessary pre-processing step. In this work we employ an inactivity recognition step which uses a sliding window to analyse a single

accelerometer axis and identify sudden peaks in activity which correlate to the beginning of a new activity. The first step is to segment all actions performed by a human subject and this is achieved by identifying locations where the subject is stationary. Therefore if a period of inactivity can be detected in the inertial sensor, where movement is detected in the sensors after a period of inactivity, this is sufficient to segment unknown human actions.

Automatic segmentation is achieved by analysing the accelerometer's z axis and using a sliding window of size 120 samples which corresponds to 1 second of data. We analyse the data in the window and measure the difference between the minimum and maximum values within the window. Where the peak difference is above a predefined threshold, we assume the minimum value is the beginning of a new action. Empirical inspection of several human subjects concluded a variable non critical threshold can be used, which is defined by the average magnitude generated by the human subject. After the beginning of a new action has been recognised, the data for the next 3 seconds is extracted and this represents an unknown action. Different sample sizes were tested but the highest classification accuracy was achieved when a sample of 3 seconds was obtained. Figure 4.4 illustrates where a human subject performs a number of sit on chair actions, the green line indicates where the action segmentation algorithm recognised the beginning of a new action.

To assess this action boundary detector, we developed an evaluation experiment which measures the accuracy of this approach. For this experiment we applied our detection algorithm to detect all the actions of five human subjects. We applied our action segmentation algorithm to each subject in turn and where the beginning of a new action was detected to within 1 second of the actual start of an action it was deemed to be the correct. Given

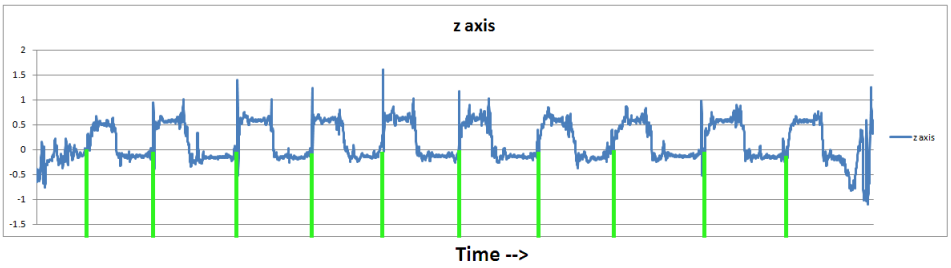


Figure 4.4: Accelerometer z-Axis while person performs a series of sit on chair actions. The green vertical line represents where the action recognition detector detects the beginning of a new action.

that we use 3 seconds of data for classifying actions using inertial sensors this level of accuracy is sufficient for classifying basic human actions. Each subject performed seven different actions and each action was executed 10 times giving a total of 70 actions per human subject and 350 actions overall. The precision and recall scores for detecting the beginning of a new action is .76 and .84 respectively.

4.3.4 Classification

To train the classifiers a supervised learning approach was used as shown in Figure 4.5. Having tested on various classifiers the IB1 classifier was selected. Bayesian Network are commonly used classifiers for training inertial sensors [93] [154] [37], but our research has found IB1 obtain high accuracy results. Given that this was also the case for visual sensors, this approach was adopted for further exploration. The process in Figure 4.5 is as follows. A ground truth was used to manually segment and group all the actions performed by each human subject. A fixed window length was used to segment each clip and this window length is generic across all actions. Each feature vector was tagged as a specific action and added to the training set. The IB1 classifier was then applied to the training set to obtain the Actions Model in Figure 4.5. Once segmentation of the test data was complete and

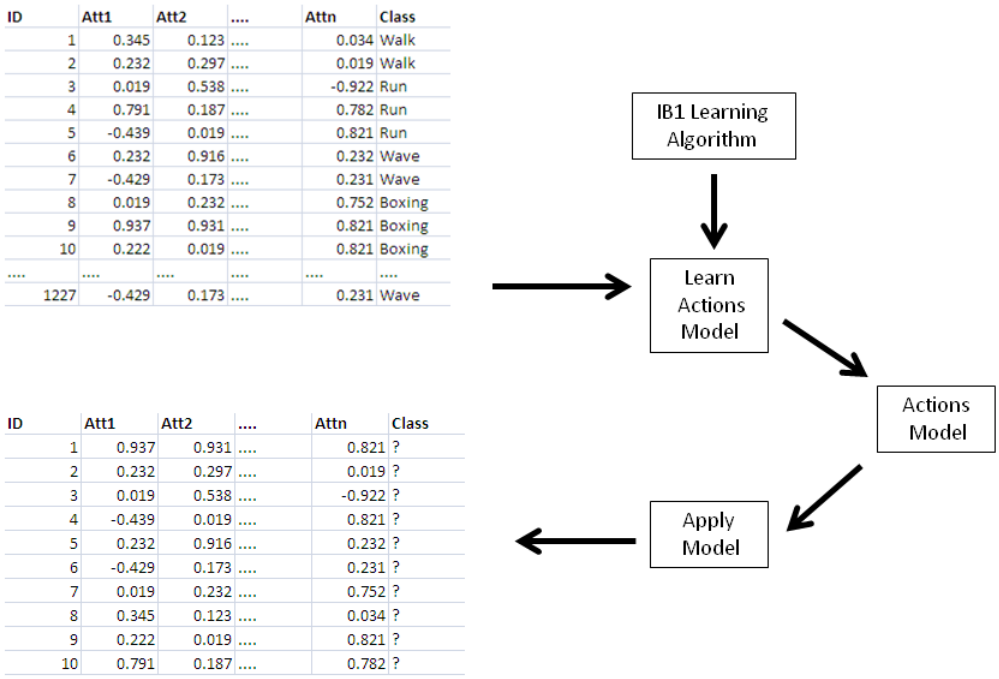


Figure 4.5: Training & testing a classification model for inertial sensing of human actions

the training classifiers for each action learned, the classifier predicts which action in the training model the input action is most similar to.

4.4 Experiments

For this experiment we used the indoor[2] human action recognition dataset which was also used in the previous chapter (Section 3.6.1). The inertial sensor was attached to the person's right forearm and each of the seven actions was performed ten times. Table 4.1 shows the gyroscope classification

[2]Note that we only use the indoor dataset as inertial sensors are unaffected by the indoor/outdoor distinction

scores for the actions trained on a random four people and tested on an unseen person who is not present in the training set. Table 4.3 shows the accelerometer classification scores and Table 4.2 shows the magnetometer scores. In this experiment accelerometers, whose accuracy is 70%, perform the best of the three sensors.

In the next experiment we assess whether increasing the number of human subjects in the training set improves the accuracy. Table 4.4 illustrates the results of this experiment where we train the classifier on ten human subjects then classify on an unseen person. We then train the classifier on a random five people and again test on an unseen person. As the results show there is no significant improvement on classification accuracy as the training set grows.

	Crawl	Walk	Jacks	Jump	Boxing	Wave	SitonFloor
Crawl	.70	.13	.02	.07	.02	.04	.02
Walk		.90	.02			.06	.02
Jack		.02	.73	.10		.16	
Jump		.12	.06	.69		.08	.06
Boxing		.10	.02	.22	.14	.24	.08
Wave	.02	.21		.10		.65	.02
SitonFloor		.06	.10	.08	.04	.16	.56

Table 4.1: Confusion matrix showing the accuracy of recognised actions for **Gyroscopes** features on seven actions trained by 5 subjects using Leave One Subject Out. The overall accuracy is 63%.

	Crawl	Walk	Jacks	Jump	Boxing	Wave	SitonFloor
Crawl	.83			.05			.12
Walk	.06	.86		.08			
Jack		.02	.67	.04	.10	.16	
Jump	.12	.11	.24	.36		.01	.15
Boxing			.18		.30	.28	.04
Wave			.20		.12	.69	
SitonFloor	.17			.25	.06	.02	.52

Table 4.2: Confusion matrix showing the accuracy of recognised actions for **Magnetometers** features for seven actions trained by 5 subjects using Leave One Subject Out. The accuracy is 60.4%

	Crawl	Walk	Jacks	Jump	Boxing	Wave	SitonFloor
Crawl	.48	.25	.06	.02			.19
Walk	.04	.82	.13				.01
Jack		.03	.76	.01	.06	.14	.01
Jump	.01	.10	.03	.82	.01		.03
Boxing			.16	.08	.74	.02	
Wave		.01	.21		.06	.72	
SitonFloor		.05	.22	.01		.03	.69

Table 4.3: Confusion matrix showing the accuracy of recognised actions for **Accelerometer** features on seven actions performed by 10 subjects using leave one subject out and trained on an IB1 classifier. Accelerometers leave one out achieves 70% accuracy.

Sensor	**Training Size 10**	**Training Size 5**
Accelerometer	72	70
Gyroscope	63	63
Magnetometer	65	60

Table 4.4: Human action classification, comparison of different training set sizes. We trained the IB1 classifier on 5 people and also on 10.

4.4.1 Discussion

Accelerometers were found to be the most accurate inertial sensors for detecting human actions, whereas the other two sensors were not accurate at classifying human actions using the "leave one subject out" approach. Table 4.4 shows that the accuracy does not improve significantly as the number of human subjects in the training set increases from five to ten. This is most probably because the classifier only needs training instances from five different human subjects and doubling the number of human subjects is not of benefit to the classification process. Although gyroscopes and magnetometers are not accurate alone, in the following section we explore the potential for fusing these sensors to increase human action recognition accuracy.

However, body worn inertial sensors can potentially overcome the challenges with which visual sensors struggle. The next chapter introduces an approach to action recognition using inertial sensors and uses the same dataset

to detect which sensor is most appropriate for human action recognition.

4.5 Conclusion

In this chapter we presented a novel approach for detecting human actions using a single inertial sensor attached to a human subject's wrist. The algorithm presented was evaluated against a dataset where ten people performed ten different human actions, ten times each. This chapter first evaluated which sensor (accelerometer, magnetometer or gyroscope) is the most accurate at detecting human actions. The outcome of this experiment was that accelerometers are the most accurate when classified using an Instance Based Learner.

Chapter 5

Multimodal Human Action Recognition

5.1 Introduction

This chapter introduces the approach used for sensor fusion of inertial sensors and visual sensors. In the next section the literature concerned with the fusion of inertial and visual sensors to achieve human action recognition is explored. We stress that it is beyond the scope of this thesis to investigate the wider area of sensor fusion and instead we only focus on sensor fusion techniques which are relevant to this thesis. We then provide details of the approach used in this work which uses data from inertial and visual sensors to detect basic human actions in an indoor environment.

5.2 Related Work

In this section we explore the different forms of sensor fusion relevant to this research. The first section looks at related work in the fusion of inertial sensors and then we explore the combination of visual and inertial sensors. Finally we discuss related work in the area of early and late fusion schemes.

5.2.1 Fusion of Inertial Sensors

In [176], Zhu et al. use two inertial sensors, one on a human's waist and the other on the foot. Each inertial measuring unit contains accelerometers and magnetometers. First the data from the two inertial sensors are fused for coarse-grained classification in order to classify the general type of the activity into one of the three following groups, zero displacement activity (standing or sitting), transitional activity (sitting-to-standing, standing-to-sitting), or strong displacement activity (walking upstairs, walking downstairs). A second step performs a fine-grained classification on the data to recognise a finer granularity of the action being performed. e.g. walking upstairs, or walking downstairs. The first classification step uses a neural network and the second step uses a Hidden Markov Model (HMM) to further distinguish the activities with promising results. In another approach to fuse inertial sensors, the authors of [4] placed one inertial sensor on the chest and a second on the rear leg and then fused the acceleration samples of the chest and the thigh. They were able to classify activities such as standing, lying, sitting and dynamic activities such as walking.

5.2.2 Combining Visual and Inertial Sensing

Nowadays since mobile electronic devices are shipped with small inertial sensing devices, research in the area of fusing inertial and visual senors within these devices has grown. This is largely because inertial sensors will more accurately calculate orientation data than vision-based tracking. For example, in [82] the authors use a gyroscope attached to a camera to calculate the current location of feature points that have been used in the previous frame. This enables Kanade Lucas Tomasi (KLT) feature tracking to adjust to bigger optical flows without losing track. Bazin et al. [14] use the knowledge of rotation from a gyroscope to measure the relative change in orientation of

the camera between two images. The second image is then warped so that its aligned with the first image and then a matching algorithm is used to find similar features of the two images.

To help create an augmented reality, the authors in [172] use a framework which combines visual sensors with gyroscopes to obtain six degrees of freedom (6DOF) pose tracking. They use a Kalman Filter and evaluation results prove that the combined method improves tracking stability over a single sensor. In [177], the authors recently introduced a novel approach to recognise indoor human daily activities by combining motion data and localisation information. In this approach, one inertial sensor is attached to the leg of the human subject to capture motion data and a visual motion capture system records localisation information. It is reported that this combination has the advantage of significantly reducing the obtrusiveness to the human subject at a moderate cost of vision processing, while maintaining a high accuracy of recognition. Bayesian Fusion is employed to fuse the motion data with the data from the localisation system.

5.2.3 Early & Late Fusion

Snoek et al. [141] define early fusion as a fusion approach which merges the features of each modality before any machine learning is conducted. In their approach, feature vectors from each sensor are concatenated to obtain a fused multimedia representation of visual, textual and audio sensors. After the multimodal features are concatenated a supervised learning approach is used to classify the semantic concepts. One advantage of early fusion is that the concatenated vector is a true representation of all multimedia streams and also that only one machine learning stage is required to classify the sensors.

Late fusion schemes have also been successfully employed to classify fea-

tures. Westerveld et al. [162] use individual probabilistic models to classify text and video using a late fusion approach. In their approach, the textual model is based on the language modeling approach to text retrieval and the visual information is modeled as a mixture of Gaussian densities. The scores are joined afterwards to give the accuracy score. The authors in [141] define late fusion as an approach which first learns the concept scores from individual unimodal features and afterwards these results are fused to learn concepts. Late fusion will allow the accuracy of each sensor to be recognised since individual sensor data is not touched in any way before the first machine learning stage. However as pointed out in [141], late fusion is expensive with regards to learning effort, as a classification stage is required for each unique modality. In addition to this, after the learned concepts are merged, a further learning step is required to obtain the final fusion prediction.

5.3 Early Fusion

In the early fusion approach used in this work, after the features are extracted from each sensor, the data is concatenated into a linear vector. In our approach, feature vectors from accelerometer, gyroscope, magnetometer, video contours are concatenated to obtain a fused multimedia representation of visual and inertial sensors. After the multimodal features are concatenated an instance based learning approach is used to classify the required actions as shown in Figure 5.1.

5.4 Late Fusion

In this thesis, we propose two late fusion approaches which are based on the late fusion approach presented in by the authors in [141]. In the first late fusion approach (Late Fusion 2 mode), which is illustrated in Figure 5.2, we

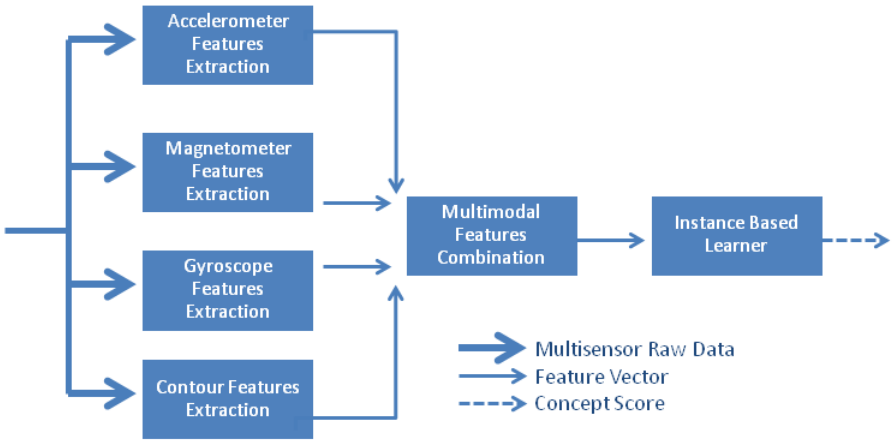


Figure 5.1: Early fusion scheme. Features are fused before a concept is learned

first concatenate the data from the three inertial sensors before performing late fusion. As Figure 5.2 illustrates, concatenation of inertial sensor data means that this approach effectively treats video and inertial sensor as two individual modalities. A supervised learning classifier is then used to obtain confidence scores of how confident we are of a classifier predicting a given class from each individual sensor modality. For each action, we classify the inertial sensor raw data and thereafter classify the visual contour data. The prediction from each modality is then examined and we assume that the sensor which gives the highest confidence is the correct prediction. Figure 5.2 illustrates this process.

The second approach (Late Fusion 4 mode) differs in that each individual inertial sensor (accelerometer, magnetometer and gyroscopes) is treated as a unique modality, therefore concepts are learned from four individual modalities (including visual contour features). Figure 5.3 illustrates this approach.

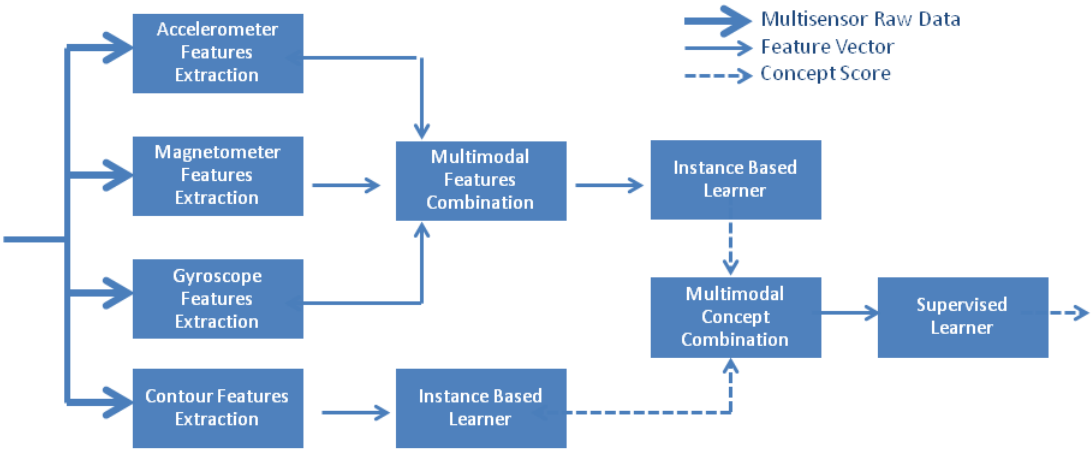


Figure 5.2: Late Fusion 2 mode. Features from two individual sensors are used to learn a concept. Then another classifier will learn from the concepts of individual sensors

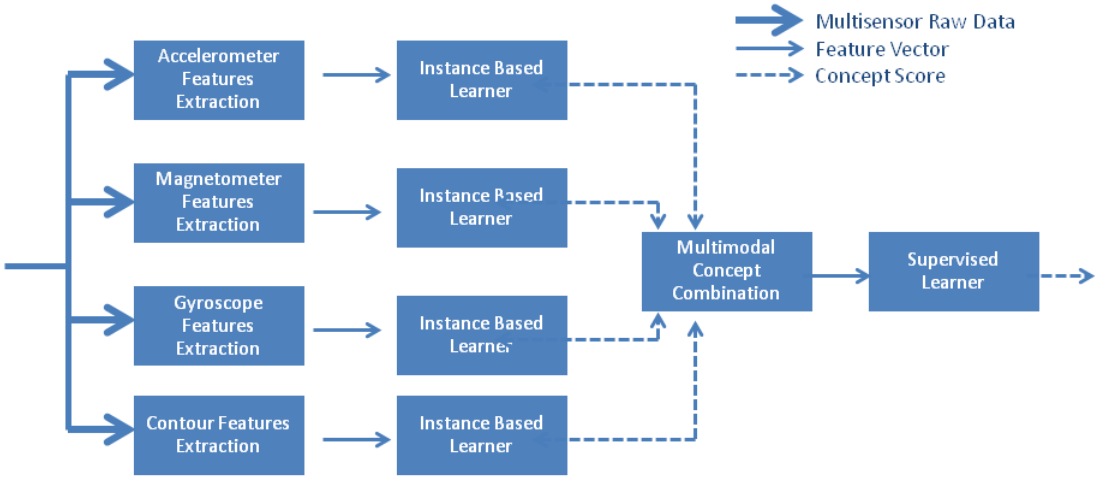


Figure 5.3: Late Fusion 4 mode. Features from four individual sensors are used to learn a concept. Then another classifier will learn from the concepts of individual sensors

5.5 Multimodal Experiments

The same dataset which was described in Section 3.6.1 is used for the following experiments. Action segmentation was achieved using frame differencing of the video and this approach is detailed in Section 3.4. Anywhere an action is detected we extract data from this location to represent each action. The visual features for each action are extracted using contour features only and four seconds of data is extracted for video, where only 3 seconds of data is used for inertial sensors. For the training data, we use the ground truth to manually select the actions used to train each model. The training set contained actions from all humans subjects apart from the person whom the model was tested on (i.e. the leave one out method). The inertial and visual data was manually synchronised offline.

5.5.1 Inertial Experiments

In this experiment we use "leave one subject out", to classify the inertial sensors using early fusion. Table 5.1 illustrates the accuracy scores when different inertial sensors are fused using early fusion. The highest accuracy is 76% which is 4% higher than the results obtained from the best single modality, in terms of highest accuracy obtained. This is obtained when accelerometers, magnetometers and gyroscopes data is fused together using leave one subject out and training the classifier with 10 human subjects.

Accelerometers & Magnetometers & Gyroscopes	76 %
Accelerometers & Magnetometers	72 %
Magnetometers & Gyroscopes	68 %
Accelerometers & Gyroscopes	67 %

Table 5.1: Using the leave one subject out approach, an IB1 classifier is trained on ten human subjects. Early fusion of different sensors and tested on an unseen human subject.

5.5.2 Visual and Inertial Fusion Experiments

In this experiment, we use the data from both the inertial and visual sensors and apply the data to both the early and late fusion approaches which were described in Section 5.3 and Section 5.4 respectively. As Table 5.5 illustrates, the accuracy for the best late fusion approach is 14% lower than early fusion. Table 5.3 illustrates the results using late fusion 4 mode and Table 5.2 shows the confusion matrix for 2 mode late fusion, which is the most accurate late fusion approach we investigated. Table 5.5 illustrates the differences in terms of accuracy between both Late Fusion approaches. Table 5.4 highlights exactly where early fusion obtains its high accuracy. Using early fusion of visual contour and WIMU combined, we obtain an accuracy of 81%, which is 5% higher than early fusion with accelerometers, magnetometers and gyroscopes only. However, visual and inertial fusion uses 1920 visual features and 2340 inertial features per instance, which almost doubles the amount of features in each instance. Therefore visual and inertial early fusion is the most accurate approach, but is also the most computationally expensive.

	Crawl	Walk	Jacks	Jump	Boxing	Wave	SitonFloor
Crawl	.69	.18	.05			.05	.03
Walk		.70	.10			.20	
Jack		.12	.63	.25			
Jump			.05	.66	.14	.05	.10
Boxing			.12		.51	.37	
Wave			.20		.03	.77	
SitonFloor			.20	.04	.01		.75

Table 5.2: Confusion matrix showing the accuracy of **Late Fusion (2-mode)** using **Video Contour and WIMU** features and a Leave One Subject Out approach

	Crawl	Walk	Jacks	Jump	Boxing	Wave	SitonFloor
Crawl	.80	.04		.04			.12
Walk	.06	.72		.22			
Jack	.03	.08	.67		.10	.12	
Jump	.23	.07	.04	.45		.07	.14
Boxing			.18		.68	.14	
Wave			.32	.16		.50	.02
SitonFloor	.04	.08	.34	.02			.52

Table 5.3: Confusion matrix showing the accuracy of **Late Fusion (4-mode)** using **Video Contour and WIMU** features and a Leave One Subject Out approach

	Crawl	Walk	Jacks	Jump	Boxing	Wave	SitonFloor
Crawl	.88	.04	.04	.02			.02
Walk	.05	.94					.01
Jack			.64	.05	.10	.21	
Jump	.03	.05	.05	.76	.01	.01	.09
Boxing			.12	.04	.67	.17	
Wave			.03	.01	.16	.80	
SitonFloor		.01	.04		.04		.91

Table 5.4: Confusion matrix showing the accuracy of **Early Fusion** with **Video Contour and WIMU** fused features, using Leave One Subject Out

Fusion Method	**Accuracy** %
Early Fusion	81 %
Late Fusion (2-mode)	67 %
Late Fusion (4-mode)	64 %

Table 5.5: Early Fusion vs Late Fusion

5.6 Discussion

Several experiments are conducted in this section. Firstly we detected that accelerometers are the most accurate at detecting actions from a single inertial sensor. In the fusion section, results prove that early fusion of all three inertial sensors is more accurate at detecting human actions than using any single inertial sensor. In addition, we conclude that early fusion is more accurate than late fusion by 14%. We also proved that early fusion of both visual and inertial sensors combined obtains slightly higher accuracy than early fusion of inertial sensors. However, combining visual and inertial features doubles the size of each training instance and will be significantly more computationally expensive to classify.

Part III

Towards Next Generation Coaching Tools for Racquet Sports

Chapter 6

Multi-Sensor Event Recognition in Tennis

6.1 Introduction

Tennis is one of the most popular court based racquet sports in the world because of the relative simplicity of the rules and the small amount of equipment needed. Despite this popularity, automated tennis video analysis has not attracted much research. The main technological advancement has been in verification of referee decisions, which has been very popular in professional tournaments. In assessing how best to present information to guide the coaching process in tennis, it is argued in [132] that a combination of both visual and verbal strategies can be effective if used correctly. In fact, empirical evidence has suggested that in tennis, the use of videotaped replay and loop-film technique has merit and can be given consideration for use in instructional settings [114].

This chapter introduces a system which, using the human action recognition tools developed elsewhere in this thesis, can automatically index many of the main tennis events performed by both players in a match. However,

there is one assumption made: that a tennis match must follow the rules of tennis as laid down by the International Tennis Federation[1] . Two cameras are fixed at each end of the tennis court and a third camera is fixed overhead. The overhead camera is used to calculate player and ball locations and the two baseline cameras are used to provide player identification and to determine in which hand the player holds the racquet. All events are stored in a database along with the relevant video resulting in a powerful coaching analysis system where no manual editing of video is necessary. It should be noted that this thesis does not address which feedback should be provided to enhance a player's performance, but instead provides a visual platform where coaches can visualise trends over multiple matches so that the information can then be easily assimilated by athletes.

6.2 Related Work

In this section we explore related work in the area of event recognition in sport. There is not a lot of published research which directly relates to tennis event detection, so we first explore literature of event detection and video summarisation in sport in general. Following this, related work concerning event detection using visual sensors is explored. Finally, this section looks at current uses of inertial sensors in a sporting context.

6.2.1 Event Retrieval in Sports Video

The ubiquitous expansion of multimedia data has driven the need for the development of automatic systems and tools for content-based multimedia analysis [28]. This research field has been very active in recent times due to the commercial interest and entertainment value which can be offered to large audiences. In particular, automatic indexing and video summarisation

[1] http://www.itftennis.com/technical/rules/

of broadcast sports video have been very active fields. In [28], the authors discuss that with an abundance of media available, viewers prefer to retrieve key events in a sporting match, rather than watch a complete sporting event from start to finish. There are numerous approaches for shot classification and highlight extraction for specific sports video, which have been developed based on a combination of extracting low-level visual/auditory features and sports genre-specific rules [29]. Other approaches use ball and/or player tracking techniques for detecting semantic events caused by player-ball interactions.

Much of the published research on event detection in sporting video is general audience oriented, where automatically indexed events are channelled to the audience automatically. In the context of communication, automatic video annotation is of particular interest since video transcoding can overcome communications bottlenecks. Another application for sports video analysis is in home video applications, where video summaries or user annotations are required to provide functionality for searching tools in large personal archives. However, sporting professionals, such as soccer coaches are more interested in the tactical events and useful statistics, which can be inferred from detected events [178]. Zhe et al. [178] explain that manual annotation systems, which provide manual editing tools along with event retrieval interfaces are no longer useful, given the abundance of information which coaches amass over time. Coaches prefer comprehensive statistics, which can be used to infer tactical patterns and help to improve performance during or after a match. To manage the ongoing video annotation operations, coaching teams frequently employ video editors to capture, annotate and organise information, which is then used to build tactical analysis and useful statistics. These video editing duties are extremely time-consuming and the possibility of using automatic multimedia event retrieval technolo-

gies is accelerating research in this area, especially by using visual sensing technology.

Managing visual data is becoming a bigger problem due to the increasing amount of content which is produced. To simplify the problem, identifying semantic indexes which can describe events within the video is very helpful. Manual annotating is simply too tedious and time consuming, making automated video indexing techniques a necessity [11].

High-Level Event Analysis

With the large volumes of sport video being captured by both broadcasters and amateurs, recent years has seen an increase in technologies which can provide high level analysis. In soccer, camera image streams are used to answer high-level analysis questions such as: What attacking plays characterise each team? What are the main characteristics of a specific team player? What team roles do these players have? Are they capable of their assigned team role? Can they accomplish their given duties? How does a particular team attack and create an opportunity to score? What are the main skills of each player? What kind of team formation is being used? In a number of team based field sports, there is an appetite for real time analysis of events from referee associations, the sports press and supporters. Automatic video analysis tools have the potential to detect erroneous refereeing decisions, by monitoring video sequences to prevent misinterpretations due to occlusion or viewpoint error or simply due to an overwhelming number of events taking place concurrently.

To detect and track the ball in soccer videos, Yu et al. [173] use a trajectory based algorithm. Ball candidates are first selected from feature objects (the goalmouth and ellipse). A Kalman filter is then used to generate candidate trajectories from the ball candidates. A confidence measure then

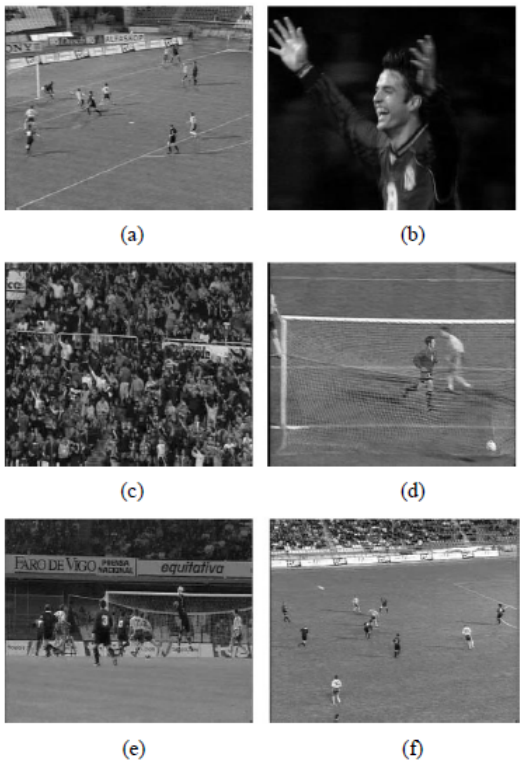


Figure 6.1: The first goal scored (a) a long range of the score, (b) camera zoom to player, (c) crowd response, (d) a replay, (e) another replay, and (f) long range view of the resumption of play. [51]

decides which of the candidate trajectories is the correct ball trajectory. High-level events such as ball touching or ball passing are then inferred, which is then used to detect team ball possession analysis.

In soccer, multi-modal information analysis is commonly used to automatically detect high level events. In [51], the authors use a rule-based and model-based system is used, which exploits heuristic rules and detects the goal event in soccer. In their approach, the authors in [51] capture a close-up of the goal event with an emotional scorer and also a goalie occurring close to shots of the crowd reaction to the goal. This is immediately followed by several slow-motion replays of the goal from different camera angles. Between the long range shot which first shows the goal to the resulting kick

off long shot, the authors in [51] define a cinematic template that needs to satisfy the following sequence of rules:

- Time allocation for a goal will last between 30 and 120 seconds;
- There must be one or more close-up/out of field shots: These may range from a player close up or a view of the crowd;
- There must be a minimum of one slow motion replay, as a reply of the goal is always played several times.

Figure 6.1 provides an illustration of the template for the first goal in the Spain1 sequence of the well known MPEG-7 data set. In this example the break duration is 54 seconds. The method for detecting goals firstly detects slow motion replay shots and thereafter it detects long shot views that mark the beginning and end of the goal sequence. Another commonly sought high level event in soccer is referee detection. The approach presented in [51] exploits the fact that referee's clothing is always distinguishable from the clothing worn by both teams. They use a dominant color region detection algorithm which is applied when there is a medium or out of field/close-up shot.

6.2.2 Event Detection in Tennis

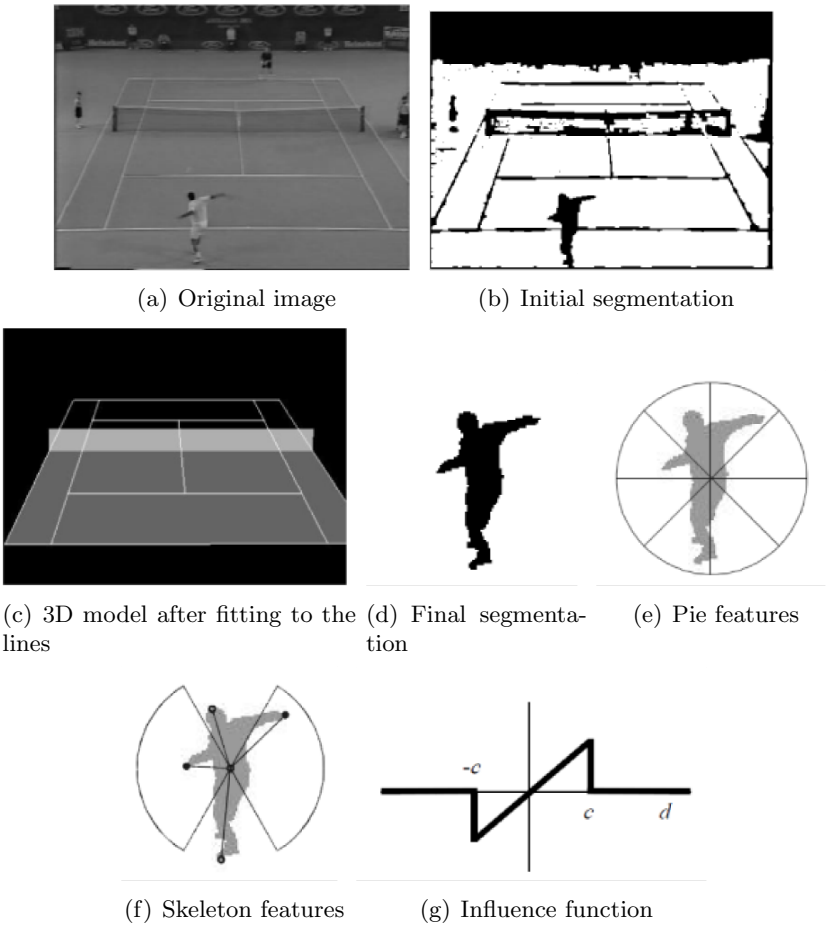
Event detection in tennis can be achieved using either visual [47] or inertial sensors [31] [3]. The techniques used to detect events vary from one sensor to the next, but either sensor can be used individually or data from both can be fused to infer conceptual knowledge of specific events [122]. The next section explores the state of the art for visual event detection in tennis and the following section examines the impact inertial sensors has had in the research field of event detection in sport, since inertial sensors are currently not as widely used in tennis as visual sensors.

Visual Event Detection in Tennis

There has been much research on tennis stroke recognition using video published over the last decade, where the main focus has been on using broadcast video to detect strokes played. In almost all of the prior works a camera positioned at height behind one baseline was used to classify events. Zhu et al. [179] extract features of tennis motions using optical flow and classify strokes using SVMs. They classify strokes into either a left-swing or right-swing class, which corresponds to a backhand and forehand stroke. Shah et al. [140] extract a skeletonisation of the tennis player's body and feed an orientation histogram of this skeleton into Support Vector Machine classifiers to distinguish forehand, backhand and 'neither'. Bloom and Bradley [20] detect a tennis stroke *keyframe* when the ball and racket collide, they then employ heuristics based on the player and racquet locations to perform stroke classification. Petkovic et al. [127] use *Contour* features and six Hidden Markov Models to classify tennis strokes as forehand, backhand, service, smash, forehand volley and backhand volley.

In our previous work [122] that provided the basis for much of the work in this thesis, we detect tennis strokes in an instrumented environment, where players are positioned in a fixed region on the court. This approach used cameras positioned behind the player whereby for each frame in a tennis shot, the player is segmented into a foreground region and then sliced into pie segments to extract action features (also known as contour features). We used these features to classify strokes into serves, backhands or forehands using either SVM classifiers or K-means nearest neighbor clustering. This approach however had several limitations which are explained in the discussion section at the end of this (related work) section.

An advanced approach which recognises a tennis player's motions is presented in [180], where the authors segment a tennis player from video data



(a) Original image (b) Initial segmentation

(c) 3D model after fitting to the lines (d) Final segmentation (e) Pie features

(f) Skeleton features (g) Influence function

Figure 6.2: Image segmentation and feature extraction. [180]

in a number of steps. Firstly, the approach identifies the tennis grass as a dominant green colour. They conclude that field color distribution can be modeled by a 3-dimensional Gaussian. The results of this can be seen in Figure 6.2(b). Zivkovic et al. then use morphological opening and closing and the player is recognised as the biggest region in the bottom half of the image. The initial segmentation (Figure 6.2(b)) is also used, after thinning, to fit a model of the tennis court lines. The 3D model is constructed having in mind the visibility of the lines. After fitting the 3D model, more knowledge is obtained about the scene at a higher semantic level (Figure 6.2(c)).

Inertial Event Detection in Sport

Only recently with the reduction in size and cost of inertial sensing, has there been a growth of research in the area of inertial sensing for sports analysis. Inertial sensors can capture enormous amounts of data on acceleration, orientation and movement of body limbs for example, which make it a useful tool for in-depth analysis, when modeled correctly. Cheng et al have developed SEnsing for Sports And Managed Exercise (SESAME), which focuses on collecting details of athletes' motion, particularly details of their foot motion which are the most rapidly changing and the most important factor that affects sprinters' performance [31].

In tennis, inertial sensors have also been used for skill analysis, in terms of how to improve a player's swing, for example. One such contribution, investigates the possibility of using wearable gyroscope sensors for skill assessment and skill acquisition in a tennis serve [3]. A marker-based method using the Vicon motion capture system[2] was used to calculate the angular velocity of the upper arm internal rotation, wrist flexion, and shoulder rotation for a range of athletes during the first serve in tennis. Participants from amateur to elite players participated in this study and results showed that the peak values of the upper arm internal rotation, wrist flexion, and shoulder rotation just before impact are indicative in classifying the participants' skill level. The correct positioning of three gyroscope sensors on the player's arm, to detect the same trends as those from the marker-based methods were then determined. As a result it was determined that gyroscope sensors could be used for skill assessment and skill acquisition for a first tennis serve [3].

[2]http://www.vicon.com/

6.2.3 Discussion

To visually detect events in soccer video, the authors in [51] employ a rule based event detection system to detect goal events. Different template based approaches can be constructed for different sports, by understanding the heuristics of the specific sport. For example, in tennis we know that the first stroke in a rally is always a serve event. Therefore, in theory this simple heuristic may help to infer the serve event. motivated by this, an appropriate ontology for tennis is presented in the following section.

There has been a lot of published research which detects strokes in tennis, but the majority of this is from broadcast video. Whilst all these techniques can be used to gain useful analysis for coaches none of these methods use non-broadcast video. The players are always professionals, making the task of stroke detection from video slightly easier than for non-elite athletes because professionals tend to execute similar kinematic movements during a specific stroke. In this thesis, we want to solve the problem of stroke detection for all levels of players from beginners upwards and moreover we want to detect other key tennis events such as rallies, games and player and ball locations.

As was discussed in the related work, we previously published work which detects tennis strokes played from a specific region on the court using either inertial or visual sensors [122]. However this approach did not work well in a real match where the players can be located in any region of the court. The reason there were issues adopting this visual analysis approach to a real tennis match were because pie features were not able to discriminate between different stroke types, if the strokes are not played from a fixed region of the court. This is due to occlusion and scaling factors. In relation to the inertial sensors, the stroke recognition approach in [122] was not accurate in a real tennis match, because there was more noise generated by players during a real match, i.e. from practice swings, running and sudden change

of directions when running. Later in this chapter we introduce a refined approach for stroke detection using internal sensors, which filters these noise events.

Inertial sensors do provide a highly accurate and portable approach for detecting human actions as was found in the last chapter. Placement of a single inertial sensor on a tennis player's wrist should not impede their play. As the related work discusses, inertial sensors have been successfully used for skill assessment and skill acquisition, but they also offer great potential to detect strokes played and later in this chapter we examine how to implement this approach and evaluate the accuracy of detecting events using a single inertial sensor attached to the player's wrist.

6.3 Tennis Event Ontology

In order to construct an event recognition system for tennis, we use the following framework, which explicitly and clearly defines the main events in a tennis match.

- A **stroke** is the act of hitting a ball with a racquet, which at a coarse level is either a forehand, backhand or serve.
- A **valid stroke** is a stroke where the ball is hit from the racquet over the net without bouncing and doesn't land outside the legal boundaries in the opponent's side of the court.
- A **rally** is a collection of valid strokes, beginning with a valid stroke and is terminated when any player fails to complete a valid stroke.
- An **ace** is when a player completes the opening stroke of a rally (serve) and this stroke is not returned by the opponent.

- A **tennis point** is awarded to the player who successfully completes the final valid stroke in a rally.

- The players **change ends** at the end of the first, third and every subsequent odd game of each set. The players also change ends at the end of each set unless the total number of games in that set is even, in which case the players change ends at the end of the first game of the next set. During a tie-break game, players change ends after every six points.

- A **game** contains a count of the total number of points won by each player. When a player's score reaches the fourth point and they are winning by two clear points the game is awarded to that player. If the scores don't have two clear points between them, the game continues until one player takes the lead by two points.

- A **set** is a score count of the games won by each player and a player is deemed to have won the set when the number of games won by that player reaches six and they have won by two clear games. Tie breakers are employed to decide a set where both players reach six games won in a set. Tie breakers are not used in the final set.

- A **match** is a sequence of sets and a player is deemed to have won a match when the player reaches 3 sets.

6.4 Event Selection

To understand which events within a match are of interest to coaches we held several meetings with coaches to extract an achievable set of requirements. We also spent time examining what events coaches currently index manually using existing sports coaching software tools. Our requirements gathering

sessions resulted in an achievable set of events. The following events were decided upon by analysing what events coaches currently index: Serve type (T, Body, Wide), first serves made/missed, return of first serves, return of second serves and strokes which hit the net. Through a series of meetings with tennis coaches (two professional coaches and two club coaches) we learned that player and ball positioning would also be a useful statistic if there was an interface where coaches could run queries based on the locations of players and ball movements. All coaches expressed a desire to be able to run queries to retrieve the number of occurrences where a player plays a specific stroke from inside a user specified area on the court. Coaches also agreed that a tool to find rallies within a match would be a useful component for finding interesting patterns. They remarked that this level of detail is simply too time consuming with manual indexing but can help to determine interesting patterns in a play.

We are not currently able to detect scores or unforced/forced errors, both of which are were sometimes manually indexed by the coaches we interviewed. In relation to forced/unforced errors there is certainly scope for a semi-automatic event indexing approach to accurately detect forced/unforced errors and this is a target for future research. We also found that coaches also record the frequency with which an event occurs and this event frequency counter has been built into our system.

6.5 Tennis Sensing Infrastructure & Dataset

The visual instrumentation includes three low cost cameras, with pan, tilt and zoom (PTZ) capability. One camera provides an overhead view of the court and the other two baseline cameras are positioned at either end of the court, as shown in Figure 6.3. The two cameras at the center of the baseline at either end of the court are AXIS 215 PTZ cameras, which are positioned

(a) (b) (c)

Figure 6.3: Three cameras required to automatically index a match into key events

2.8 meters above the court and have very high zoom functionality, as well as physical pan and tilt. The overhead camera is positioned at 13.8 meters from the ground. This camera is an AXIS 212 PTZ camera, which has a wide angle lens (140°) and includes fast digital PTZ functionality by sub sampling from a high-resolution sensor. The system supports additional cameras for visual feedback, but automatic tennis event detection only requires the three cameras mentioned above.

The specifications of the WIMU used in this experiment can be found in Section 2.4. A single WIMU is attached to the dominant forearm of a tennis player as shown in Figure 6.4. The WIMU receiver was positioned at the side of the court beside the net.

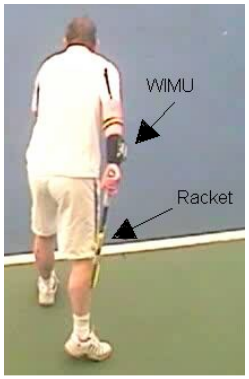


Figure 6.4: WIMU as positioned on a player

Twelve complete matches were recorded from players of various skill lev-

els, corresponding to 825 minutes in total. A ground truth was generated offline by manually annotating each tennis event which is to be automatically indexed. Each match was the best of three sets, played according to the rules of the ITF. Each player wore a single WIMU attached to his/her arm and the video data was recorded from all three cameras. The WIMU and the video data were manually synchronised offline before the event detection algorithms were applied. Six of the matches were played by players who play tennis at least once every two weeks and the other six matches from players who play once every two months at most. The diversity in skill levels is necessary to test whether we can detect events played by players from varying skill levels.

The infrastructure used in this thesis may be assembled indoor or outdoor. In the case of an outdoor setup, two 13 meter poles at either side of the net could be assembled and a cable could run from the top of one pole to the other. A single aerial view camera would then be mounted to the centre of the cable to capture the overhead movements of the ball and players. Aerial view instrumentations are common nowadays to capture overhead shots of American football games in action and also in soccer pitches. In both American football and soccer, the camera is mobilised via a network of overhead cables, to allow it to move across the field, making each instrumentation much more complex than would be required in tennis.

It was not feasible for the research reported here to mount such an outdoor structure. However, our aerial view experiments in Section 3.7.3, have proved that human actions can be inferred from an outdoor aerial view to a high degree of accuracy. Moreover, as long as the court surface is not yellow in colour, which is the colour of a tennis ball, ball tracks can be easily inferred from an overhead camera using the approach used in Section 6.7.2. The inertial sensing approach described here transfers directly to outdoor

scenarios.

6.6 Inertial Event Detection

In this section, we describe our approach to automatically index a tennis match based on strokes played using a single inertial sensor, which is attached to a tennis player's forearm (as shown in Figure 6.4). This section builds on the research conducted in Chapter 4, which detected human actions using a single inertial sensor attached to a human subject's right arm. For tennis stroke classification, we classify the main types of tennis strokes (forehand, backhand and serves) played in a competitive match. This approach delivers a new contribution which can classify tennis strokes performed in a competitive match by players of different levels using either accelerometers, magnetometers or gyroscopes. The two-level classification process used in this approach can filter any player movements where they are not performing a tennis stroke whilst the second step classifies candidate tennis strokes into serves, backhands or forehands. We evaluate the accuracy of using accelerometer, gyroscope and magnetometer sensors to perform tennis stroke classification on previously unseen players.

Automatic detection of tennis events is necessary to reduce the time a coach will spend manually indexing a recorded match. The advantage of using inertial sensing to index tennis strokes is that it does not suffer from the limitations of visual sensing, visual sensing can suffer from self occlusion and player orientation issues, which are inherent in wide area scene analysis.

6.6.1 Tennis Stroke Detection

In this section, we give an overview of our tennis stroke detection system using inertial measuring units only. As each inertial sensor contains accelerometers, gyroscopes and magnetometers, we can detect strokes played

using any combination of these three sensors.

A single inertial sensor placed on a player's dominant arm will register a spike in its accelerometer data due to the impact of the ball on the tennis racket. Detecting such data-spikes provides the temporal location of tennis strokes. To detect ball contact impacts, we first compute the acceleration magnitude for each sensor sample, simply by taking the length of the 3D acceleration vector. We then select all impacts which generate a magnitude which is above an adaptive threshold value. A players average magnitude for all strokes in match will vary from one player to the next and this adaptive threshold is calculated as thus. The top 400 accelerometer magnitudes for each player in a single match are taken and the mean threshold value is obtained from this set. Then by selecting the adaptive threshold value from Table 6.1, we identify all locations where the player's magnitude is above his/her adaptive threshold value. We select the value with the largest absolute magnitude in the data. A 1-second window around this peak is extracted to represent a candidate stroke in progress. In general all tennis strokes will not take any longer than 1 second to complete. Adopting a greedy approach, this window is removed from the data and the procedure is then repeated to find the remaining candidate strokes, until we have extracted all candidate strokes which generate an accelerometer magnitude above a players threshold.

Player's Average Range	**Adaptive Threshold Used**
<3.5	3
3.5-4.5	4
4.5-6	5
>6	6

Table 6.1: Adaptive Threshold table

However, there are other actions which a tennis player will perform during a match that will generate a spike in accelerometer magnitude and there-

fore it is necessary to identify which of these spikes are tennis strokes and which spikes are generated from a player performing a non tennis stroke. A non-stroke action can include using the racket to lift a ball of the court surface or twirling the racket in a players hands whilst waiting on an opponent to serve, which is in fact quite common. Also problematic are activities such as running, practice swings and arm movements performed during rest periods. For this reason, a two level classification system to classify candidate strokes is used.

Stroke Classification

The classification of tennis strokes is accomplished in two steps as shown in Figure 6.5. The top level of this process filters non-stroke events, which generate an accelerometer magnitude from various arm movements during a match. The second stage of the classification uses either accelerometers, magnetometers or gyroscopes to classify all the candidate strokes into either serves, backhands or forehands.

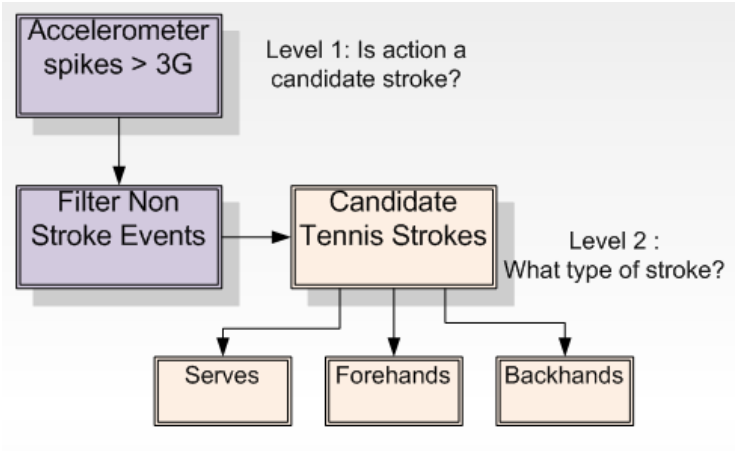


Figure 6.5: Two-level classification: Step one filters noisy data and step two classifies the remaining candidate strokes. Both steps a Instance Based Learning classifier

Filtering Non-Tennis Stroke Events

To filter non-stroke events, we use the accelerometer data from 8 players during a tennis match to create two global feature vectors, one global feature vector contained a mixture of serves, backhands and forehands as played in a competitive match. The second global feature vector contained all non-stroke events. The model was trained using a Instance Based Learning classifier, which are known to be effective at classifying instances with a high attribute list.

Each W-second accelerometer instance with a magnitude of 3g or greater was fed into the binary classifier and any instance which was predicted as a non-stroke event was filtered from the dataset. The remaining candidate strokes were passed to level two of the classification process to predict if the stroke is a backhand, forehand or serve.

Candidate Stroke Classification

To classify candidate strokes into serves, backhands and forehands, we first trained three classifiers for each stroke. For each serve, we trained a classifier with accelerometer data from a subset of serves by various players, then a second serve classifier was trained with gyroscope data of serves and a third serve classifier was trained with the magnetometer data for serves. Similarly, three classifiers were built for forehands and backhands.

Having filtered out noise from the data we have the temporal locations of all candidate strokes that a single player has performed during a match. Using the temporal locations we can select each candidate stroke in turn. For each candidate stroke, we then select the accelerometer data at this time and compare it to the serve classifier, forehand classifier and backhand classifier for the accelerometer data. The model is then able to predict if the candidate stroke belongs to the serve class, forehand class or backhand class.

An identical approach is used for gyroscopes and magnetometer classifiers.

6.6.2 Experiments

In this section we discuss the experiments to assess how accurate tennis stroke recognition is using inertial sensors. The results and findings from these experiments are also discussed in the subsequent sections.

Filtering Non-Stroke Events

Step one in the two stage classification process detects which spikes in the accelerometer data are likely to be tennis strokes and which can be considered non-strokes. To verify the accuracy of this binary classifier, we performed 10 fold cross validation on the entire dataset of accelerometer instances from all players which were above 3g in magnitude.

As the results show, this filtering process has a very high accuracy rate at removing any non-stroke events. This filtering is necessary to create accurate candidate stroke classification in step 2 of this classification process. With respect to Table 6.2, precision is the number of correct results returned, divided by the number of all returned results. Recall is the number of correctly classified strokes divided by the number of results that should have been returned, while Acc. is the percentage measure of the correctly classified instances.

Category	**#**	**Precision**	**Recall**	**% Acc.**
Candidate Strokes	2090	0.911	0.910	91
Non-Candidate Stroke events	5749	0.904	0.909	91

Table 6.2: Detecting all non tennis stroke events which are generating a spike in the magnitude of accelerometer data.

Stroke Recognition

A series of experiments on stroke recognition are reported in Table 6.3, which shows results for advanced (Adv.), intermediate (Inter.), and novice players. The accelerometer stroke classification section in Table 6.3 illustrates how the stroke classifiers performed when trained on 7 players and tested on a unseen player. It also illustrates the results when trained on a random 4 players and then tested on an unseen player. The gyroscope section and magnetometer section in Table 6.3 report results in a similar way.

Testing classifiers on players not in the training set

To evaluate how accurate each sensor is at predicting strokes from a player who is not in the training set, we trained the classifiers with 7 players and tested on the remaining unseen player. For each player the classifiers were retrained using all the other players to find if stroke classification can be achieved by testing on an unseen player. The results are displayed in the 7 player training size section in Table 6.3.

As the results in Table 6.3 show, stroke classification can indeed be accurately achieved without training the classifiers on the player who performed the candidate stroke. It should also be noted that since the classifiers were trained on a variety of players from different skill levels, 79% accuracy is very encouraging. As expected however the gyroscopes perform the worst of the 3 sensors, this is because the measure of temporal orientation during a given stroke will be significantly different between players of different skill levels. This is in contrast to a tri-axis accelerometer, which measures acceleration on three planes, which will be more effective at classifying strokes from different skill levels.

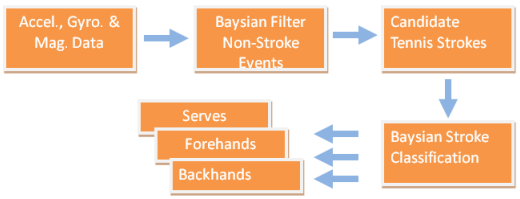


Figure 6.6: Inertial Event Detection Overview

Evaluating classification performance as training size decreases

We trained the stroke classifiers on 4 players and tested on an unseen player. The results from this experiment were compared to the results from training on 7 players. In either of the three sensors, there was no significant decrease between training on 7 players or training on 4 players, as the accuracy results in Table 6.3 illustrates. The T-test difference value is .32 between the results from the two training sizes , which proves that the results do not change as the training set changes the number of players in the set.

Sensor Fusion Comparison

In this experiment we trained the classifiers with a combination of the data from three sensors to identify if stroke recognition performance is improved using early fusion. The classifiers were trained using seven players and again tested on a player not in the training set. Using this leave one player out approach, we retrained the classifiers for each player. The results can be seen in Table 6.6. Interestingly, we discovered that using a combination of accelerometer, gyroscope and magnetometer sensors gives an overall stroke recognition performance of 90%. This accuracy rate is 10% higher than that of accelerometer classification, which gave the highest results in the single sensor classification in Table 6.3.

Training Size:		7 players 2543 strokes	4 Players strokes
Player	**Test Strokes**	**%**	**%**
Accelerometer Stroke Classification			
Adv. Player A	597	79	78
Adv. Player B	197	85	81
Adv. Player C	177	81	82
Inter. player D	220	69	68
Inter. player E	333	85	80
Inter. player F	325	88	85
Novice player G	163	72	85
Novice player H	225	86	83
Overall Accuracy		81%	80%
Gyroscope Stroke Classification)			
Adv. Player A	597	68	68
Adv. Player B	197	82	87
Adv. Player C	177	83	79
Inter. player D	220	63	71
Inter. player E	333	76	55
Inter. player F	325	77	68
Novice player G	163	88	75
Novice player H	225	88	72
Overall Accuracy		77%	72%
Magnetometer Stroke Classification)			
Adv. Player A	597	79	76
Adv. Player B	197	75	81
Adv. Player C	177	65	77
Inter. player D	220	86	54
Inter. player E	333	78	82
Inter. player F	325	80	71
Novice player G	163	79	85
Novice player H	225	82	73
Overall Accuracy		78%	75%

Table 6.3: Tennis Stroke classification classified with an Instance Based Learner, testing on a player not in training set. One classifier is trained on 7 players and the other is trained on a random 4 players to illustrate performance decrease as the training set decreases.

Early Sensor Fusion	Overall Accuracy %
Accelerometer & Gyroscope	83
Accelerometer & Magnetometer	88
Gyroscope & Magnetometer	88
Accelerometer & Gyroscope & Magnetometer	91

Table 6.4: To analyse the benefits of sensor fusion before classification, we performed experiments training from 7 players and testing on an unseen player.

6.6.3 Inferring Rallies, Games & Change of Ends

The first step in this approach is to filter all the feeder strokes, which occur when a player hits balls to a serving opponent before a player serves. Feeder strokes are non competitive and also very common. They can be found by finding backhand or forehand strokes which are not preceded by a serve. To find all rally boundaries, we then group all forehands and backhands which follow a serve until a new serve occurs. To detect game boundaries all rallies are grouped together until the serve changes from a player to the next. Since players change ends at the end of the first and every subsequent two games as defined in the Tennis Ontology (Section 6.3), we can use the game boundaries to infer a new change of end event. We applied our event detectors to the dataset in Section 6.5. Rallies were considered correct if indexed to ± 2 seconds, while a game was considered correct if indexed to ± 10 seconds. Since there is an average break in play of no less than 5 seconds between rallies an accuracy rate of ± 2 seconds is sufficient. There is usually a break in play between rallies of no less than ten seconds, while players will normally collect balls or change ends between games and therefore if we are correct to ten seconds this will be sufficient for indexing a game boundary. The precision and recall for detecting rallies was .86 and .76, while the precision and recall for detecting games was .92 and .92 respectively. Precision is the number of correct results returned, divided by the number of all returned results. Recall is the number of correctly classified results divided by the

number of results that should have been returned.

6.6.4 Discussion

Overall, accelerometers perform the best of the three sensors at stroke recognition when trained on multiple players, but when we fuse the data from the three sensors using early fusion and train the classifiers on a large data set this gives the best performance as illustrated in Table 6.6. By integrating results inferred from inertial sensors with visual sensors, it may be possible to infer new events and sensor integration is detailed later in this chapter.

6.7 Visual Event Detection

This section first outlines the algorithm steps for detecting events using video only and the remainder of this section explains each of these steps in detail. This visual event detection technique combines various computer vision modules with tennis heuristics to detect multiple tennis events in a step by step fashion.

6.7.1 Visual Event Detection Algorithm Overview

1. **Player and Ball Tracking:** The overhead camera is used to detect player movements and ball movements

2. **Serve Detection:** Player localisation from (1) is used to determine when players are in a given serve zone and the opponent is in a respective return zone.

3. **Change of End Detection:** Player identification is based on change of end times detected in step 2., since only two players are on the court, we can assign each player and ball track to each player once we know the side of the court a given player is located within.

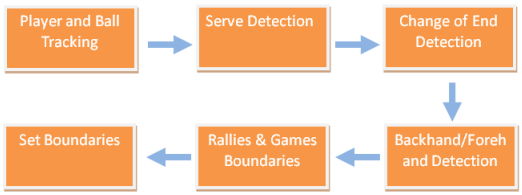


Figure 6.7: Visual Event Detection Overview

4. **Backhand and Forehand detection:** A "Dominant Arm Detector" uses the baseline camera to infer if a player is serving with their left hand or right hand. Once we know which hand the player is holding the racquet with, we use the overhead camera to detect if a player performs a left swing or a right swing when striking a ball during a rally.

5. **Rally Detection:** The occurrence of a rally can be inferred from serve detection and forehand or backhand detection.

6. **Game Detection:** The start of a new game can be inferred from serve detection as we recognise when the serve switches from Player A to Player B and vice versa.

6.7.2 Player and Ball Tracking

Using the overhead camera, we can track the tennis ball using motion images for ball candidate detection followed by linking candidates into locally linear tracks. To detect both players from the aerial view camera, we use background subtraction and hysteresis-type blob tracking to track the tennis players positions. The performance of both modules has been evaluated in a publication [36], which this author was involved in, however it is outside the scope of this thesis to reevaluate the performance of this module as it is not considered a key contribution of this work. Instead we use this module

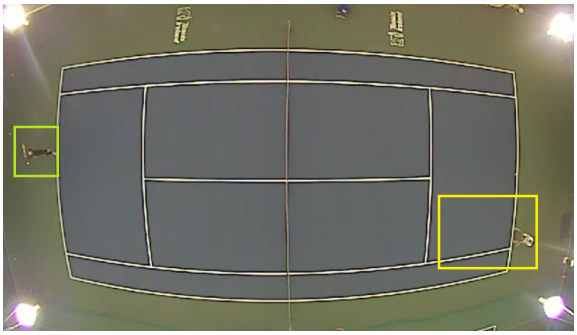


Figure 6.8: Player A is inside serve zone (left side) for two seconds and Player B is inside return zone for four seconds, therefore Player A is serving.

and its functionality to infer new events such as stroke subcategories, for example.

6.7.3 Serve Detection

This section introduces two approaches for detecting a serve event. The first approach uses what we refer to as serve zones and is introduced in the next section. The second approach is based on the output data from the ball tracking algorithm (Section 6.7.2), however for serve detection with ball tracks we enhance the ball tracks by adding a filtering step as outlined below.

Approach 1: Serve Zones

As was discussed in Section 3.7.1, there has been much research conducted which detects human activities from an aerial view camera. With the knowledge learned from previous related work in this area, this thesis introduces a rule based method to detect serves from an overhead camera by using player localisation coordinates along with tennis heuristics. Using the information from the player tracks in Section 6.7.2, we can locate both players positions and map these coordinates to the tennis court to determine each player's location on the court. Then, by determining both players locations on the court at all times during a match, we are able to recognise a serve event.

Conroy et. al. [38] states that a "receiving player is usually beyond the baseline on the opposite side of the court. However in practice, the receiver can be closer to the net, or within the baseline". A server though, is always located behind the center of the baseline and will alternate serves between the left and right of the center mark on the baseline. Taking these starting positions for a server and receiver into account, we created two distinct zones to detect a serve event, one zone to locate a server (serve zone) and the second zone for the receiving player who returns the ball following a serve event (return zone). By observing video footage of where tennis players are located on court whilst a serve event is in progress, we conclude that a serve will originate from four unique serve zones and a serve will be returned by the opposing player from four unique return zones. There are eight zones used to identify serves, four are serve zones (S_1, S_2, S_3, S_4), whilst the remaining four zones are return zones (R_1, R_2, R_3, R_4), as illustrated in Figure 6.9.

The rule for serve detection is as follows. For two players P_1 and P_2, let P_1^t and P_2^t be the positions of P_1 and P_2 at time t respectively. If P_1^t is within the bounds of the serve zone S_x for s seconds and P_2^t is within the bounds of the return zone R_x for q seconds, then P_1 is declared as serving from S_x. A similar rule exists to detect if P_2 is serving. After analysing precision and recall results with different s and q values, s was set to four seconds and q was set to four seconds and these values were used to detect serves in all the matches in the dataset.

Serve zone coordinates were obtained by observing the typical locations of players when serving from each serve zone. A similar approach was used to detect return zone coordinates. A number of evaluations were carried out whereby, we manually tested an initial set of coordinates based on observations and by analysing the precision and recall results, we modified the respective zone coordinates until the error in the precision and recall results

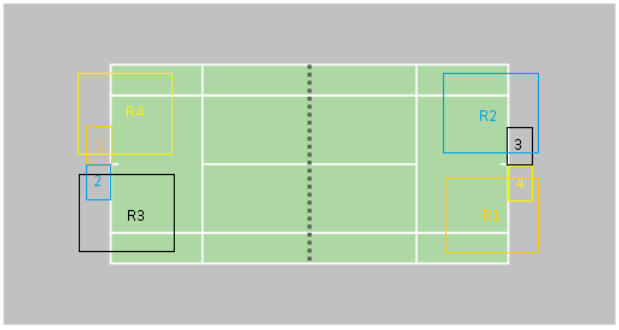


Figure 6.9: Serve zones 1, 2, 3, 4 and the corresponding return zones R_1, R_2, R_3, R_4 are used to detect a serve based on both player's locations.

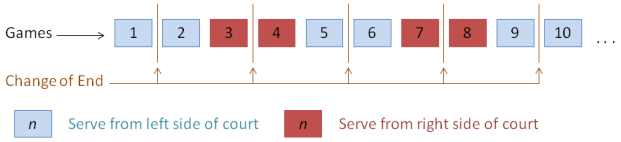


Figure 6.10: Pattern for serve direction and change of end, one change of end occurs for every change of serve direction

were minimised.

Approach 2: Ball Hit Analysis

In this approach we use the ball and player tracks (Section 6.7.2) along with tennis heuristics to infer the temporal locations of all strokes using the overhead camera. We filter out any anomaly strokes which are not preceded or succeeded by a stroke within a given time frame (usually two seconds), these strokes are deemed to be non competitive strokes, where players feed balls to each other between serves. Then we select all the serves, as a serve is always the first stroke in a rally and it should not be preceded by a stroke within a small interval of it occurring.

6.7.4 Change of End Detection

To detect a change of end event using video, we use a three step approach which combines a rule based event detector with tennis heuristics and a person identification method. The steps required for detecting the change of end event are introduced below.

1. Serve Direction Filtering

The first step uses serve direction patterns (see Figure 6.10), which can be inferred from Serve Detection (Section 6.7.3). After careful examination of tennis match patterns, we have concluded that there is only one change of end (COE) event for every change of serve direction event (COSD). By serve direction, we mean the flight of the ball after it is struck by the racquet, during a serve. There are only two serve directions possible in tennis, left court to right court and right court to left court. The serve direction pattern over an entire match will change from left to right and vice versa every two games as indicated in Figure 6.10. This is because Player A will serve the first game and then a change of end event occurs between the end of the first game and the start of the second. However, Player B will then start serving the second game from the same side of the court from which Player A served the first game. Therefore the serve direction is the same for the first two games. Player A will serve for game three and since no change of end event occurs till the end of game three, the serve direction changes at the end of game two. Therefore we can assume that only one change of end event is present between two COSD.

2. Player Tracking - Candidate Change of End

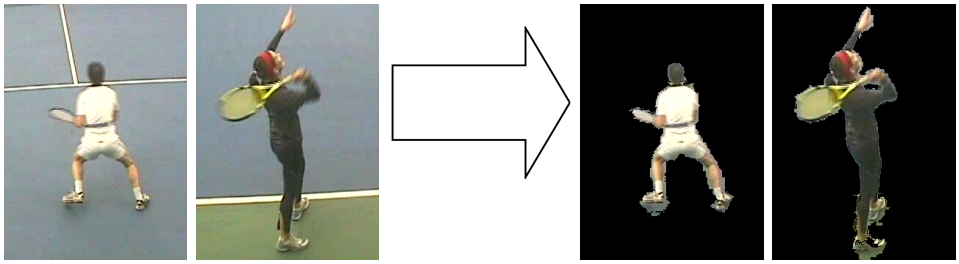
After we have identified all candidate COSD events, we need to find the best choice for a change of end between two COSD events. We use player tracks

(Section 6.7.2) to retrieve a temporal location where both players approach and/or walk from one side of the net to the other, as they will need to walk past the net at the change of ends. The event is automatically tagged if both players are within 2 meters of the net at any time between two consecutive serves. Sometimes the player tracks can get confused when two players walk close to each other, so we cannot rely on player tracks alone to retrieve a change of end event. If the player tracking in this step finds more than one candidate change of end event, between two COSD events, we need to check if there is a new player serving after a candidate change of end event and this is achieved with the following step, Serve Authentication.

3. Serve Authentication

If necessary, a final step can be used, which exploits a recent fashion trend in tennis where players tend to wear colorful clothes. This trend can be credited to the fact that most professionals nowadays are sponsored by clothing companies and as a result amateurs also tend to wear distinguishable clothing. This approach takes all the candidate change of end events between two COSD events and finds the best match. The rear view camera is used to inspect the colour features of a serving player in the previous serve and next serve. Colour features have been used in combination with other visual descriptors for identification of different people in the past [125] and our person identification approach, which also uses colour descriptors is detailed in the following.

To identify if a new player is serving, the previous serve is extracted and 60 frames are extracted from the rear view camera, which gives us two seconds of the serve. The player is extracted as a colour foreground from each image and the resulting HSV image is then split into three channels (Hue, Saturation and Value). We discard the first two channels (Hue and



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

Figure 6.11: Player foreground is extracted and the colour features of the players are compared using the Bhattacharyya coefficient to detect a change of end event.

Saturation), as analysis concluded that the Value channel alone provides sufficient information to accurately detect a change in the player's appearance. We then create an image histogram from the Value channel, which represents the brightness in the player foreground. We then apply an identical image processing technique to the next serve after the candidate change of end and the similarity distance between the histogram for the previous serve and the next serve is calculated using the Bhattacharyya coefficient, which has been widely used to compare image histograms [89]. Calculating the Bhattacharyya coefficient is based on identifying the overlap between two samples as outlined in the following formula,

$$B = \sum_{i=1}^{n} \sqrt{(\Sigma a_i . \Sigma b_i)} \tag{6.1}$$

where the two samples are a and b, n is the number of partitions, and Σa_i, Σb_i are the number of members of samples a and b in the i'th partition. A new player is deemed to be present in the image when the difference between the previous and next serve exceeds a threshold.

6.7.5 Detecting a Player's Dominant Arm

The biomechanical movements of a typical serve are quite similar from one player to another. However, there are clear differences between the serve of a left handed player and right handed player in that a left handed player will throw the ball upward with their right hand and swing the racquet with their left hand. While the reverse is true for right handed players. Using this biomechanical observation, we can automatically infer if a player is left handed or right handed using the camera behind the baseline as illustrated in Figure 6.12. In order to successfully predict if a player is performing a forehand or a backhand using the overhead camera (Section 6.7.6), we need to know if the player is left handed or right handed. The Serve Detector in Section 6.7.3 gives us the temporal and localised positions of serves. With this data we then use the camera behind the baseline of the serving player to extract a rear view of the player's serve. We then use the following approach (which is based on contour features, introduced in Section 3.5.2) to determine whether the player is right or left handed.

Using the camera behind the baseline as illustrated in Figure 6.12, we segment the player as the foreground for each image in a serve, the player is always the largest foreground connected component in the image. In each frame, we use background subtraction to determine pixels belonging to the tennis player, as illustrated in Figure 6.12(b). To extract contour features, we divide the player foreground region into 16 *pie segments*, centered on the player centroid. Over the entire stroke, we extract pie features for each video frame. We then normalise the features in order to make them invariant to the player's distance from the camera, by computing the median of all pie feature values and dividing all features by this value. Figure 6.12 illustrates the pie features we extract from video clips of tennis strokes.

A binary classification system was built to predict if a player is serving

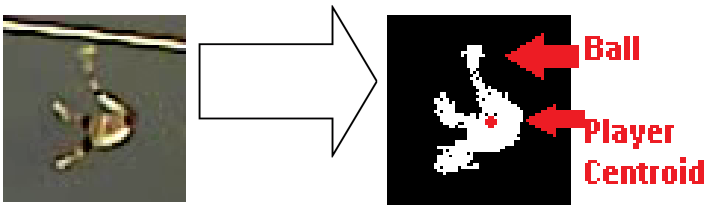
(a) Camera image (b) Foreground (c) Radial map

Figure 6.12: Contour feature extraction.

with their left hand or right hand. Using the image-based pie features of all 60 frames for each serve, an Instance Based Learner was trained with an even distribution of serves from left handed players from various skill levels. Similarly, we trained a second classifier with serves from right handed players of varying skill levels. Then, for each new match, we select the first ten serves detected by the serve detection (Section 6.7.3) from each player, we compare the input serve to both classifiers and aggregate the ten predictions, to ascertain whether the player is left handed or right handed.

6.7.6 Forehand and Backhand Detection

One clear difference between a backhand and a forehand is the positioning of the ball in relation to player (left or right of the player) during the execution of a stroke. In this work, we use this heuristic to detect forehands and backhands from the overhead camera. First, we automatically remove the serves from the collection of ball hits. Using only the overhead camera, we have developed two new approaches to detect if a stroke is a forehand or a backhand. Both approaches use only the overhead camera and the ball tracks. The first approach detects whether the ball is positioned to the left or right of the player when a stroke is executed and the second approach trains a classifier with the pie features of the players actions during a stroke.



(b) (c)

Figure 6.13: Player centroid and ball location are compared on the y-axis to determine if the ball is struck above or below a player

Left swing/ right swing analysis

Given that we know the starting position of each ball track (Section 6.7.2), we can infer which ball hits are serves, since the first ball hit in a rally will always be a serve. We remove the serves from our collection of ball hits. We also filter any non-competitive strokes which commonly occur when one player feeds a ball to another between serves. For each remaining ball hit in turn, we use the ball tracks (Section 6.7.2) for that stroke to find the origin of the stroke and the (x,y) position of the ball at the start of the stroke. We then extract the player as foreground from the video and obtain the centroid of the player as shown in Figure 6.13. This leaves us with the position of the player and the position of the ball at the beginning of a stroke. Next we detect whether the ball is above or below a player at the start of the stroke and given that we know whether a player is right or left handed (Section 6.7.5), we can infer if a stroke is a forehand or backhand. This technique is similar to [20], where their algorithm detects when the ball and racket collide and heuristics are then employed based on the player and racquet locations to perform stroke classification.

Forehand & Backhand Detection with Contour Features

Since the player and racquet movements can be detected during a stroke using the overhead camera, we extract visual features from these movements and then use an Instance Based Learner classifier to learn if the player swung the racquet to the left or right side. This approach also uses ball hit temporal locations to extract 35 frames per stroke. We then extract the player as foreground from the overhead camera and compute the pie features for each frame in the stroke. Noise is removed from each frame, and we concentrate on a specific region of interest, which is calculated from the first ball track.

All candidate forehands and backhands are then sorted into two groups, those that originate from the left side of the court and those that originate from the right side. If the candidate stroke originated from the left side, it is tested against a binary Instance Based Learner classifier which was trained on left side strokes only. The classifier has two classes, one for balls struck above the player and a second for balls struck below a player. The classifier is then able to predict if the stroke which occurred on the left side was struck above or below a player. We merge this knowledge with the dominant arm detector (Section 6.7.5) to infer if the stroke is a forehand or a backhand. A similar supervised classifier approach is used to detect forehands and backhands that originate from the right side of court.

6.7.7 Rallies & Games

Both rallies and games can be inferred by combining the tennis events above with basic tennis heuristics. To detect the rally event, we first tag a serve as the start of the rally. By empirical observation, we conclude that a rally then continues for each sequential non-serve stroke, within two seconds of the previous stroke, until there is at least a five second gap between the next stroke. A new serve will also indicate a new rally has begun. Once we

have identified which player is serving we know that the game boundaries are located wherever the serve event switches from Player A to Player B and vice versa.

6.8 Experiments

This section first details the experiments used for measuring the accuracy of the visual event retrieval system. For the purposes of sensor comparison, the results of how well inertial sensors performed in detecting events from the 12 matches in this dataset are reproduced in Table 6.5. The dataset of tennis matches is detailed in Section 6.5 and the results in the following section are based on experiments conduced on this dataset.

Event Detection Procedure

For each tennis match, we detected the tennis events firstly on the video data and then with the inertial data. A manually annotated ground truth for each match was used to measure event detection accuracy for each event. When all the matches were fully indexed, we took each sensor in turn and calculated the median score of how accurately each tennis event was indexed over all the matches in the dataset.

	Video			**WIMU**		
Event*	**Method**	**P**	**R**	**Method**	**P**	**R**
Serves	Serve Zones Ball Hits	.79 .68	.82 .88	Acc+Gyro+Mag BC	1	.79
FH & BH*	Left/Right Overhead Pie Features	.71 .74	.84 .77	Acc+Gyro+Mag BC	.99	.78
Dominant Arm	Baseline Pie Features	1	1	-	-	-
Change of End	Histogram Differential	1	1	Inferred	1	1
Rallies	Inferred	.78	.71	Inferred	.86	.76
Games	Inferred	.75	.75	Inferred	.92	.92

Table 6.5: Tennis event detection results using video or inertial sensors. *FH & BH = forehand and backhands; P = Precision, R = Recall

A comparison of how well each sensor is able to detect each event is

given in Table 6.5. It can be seen from this table that inertial sensors are significantly more accurate at detecting events than visual sensors. This is due to the high accuracy of stroke recognition obtained from inertial sensors. This table gives the median score of how accurately each sensor can detect a given event when tested on all the matches in the dataset. If a serve, forehand or backhand event was indexed to ± 1 seconds it was considered a correct match and this level of precision is more than sufficient for an event retrieval system. A change of end event needed to be indexed anywhere between the end of one game and the first serve of the following game to be correct. Rallies were considered correct if indexed to ± 2 seconds, while a game was considered correct if indexed to ± 10 seconds. It is clear from the results that inertial sensors are more accurate at detecting all of the events and it is our belief that this score would be much higher if all the players were regular tennis players. This is because novice players do not possess a well defined serve, forehand or backhand stroke.

With respect to the visual event recognition system, the change of end event is generally detected by the first two filtering steps. In fact, player tracking (Step 2) was usually all that was required to detect the change of end event, but sometimes the players would collect balls at the net and therefore Step 3 was invoked to find the best candidate change of end between two change of serve detection events. In the event of no change of end event being detected between two change of serve direction events, the system will flag an anomaly at the next change of serve direction event (see Figure 6.10) and can correct itself, by using the change of end times from the inertial sensors. When all the core tennis events are indexed, it is straightforward to infer other events, such as first serves made/missed, return of first/second serves, and when a player hits the net, for example. We do not give precision and recall results for these further events as they are simply an amalgamation

of the core events in Table 6.5 and the player and ball tracks. However, the following section details how these high level tennis events are detected.

6.9 Sensor Data Integration

Event indexing is first executed from separate sensors and then both sensors are synchronised, before selected events from each sensor are imported into a relational database. This indexing system may work in real time in the future, but for now data integration is performed offline. The visually detected events which are imported are player and ball tracking and change of end events. The inertial sensors provide forehand, backhand, serve events for both players. Structured Query Language (SQL) integration queries then map all player strokes in the database to the relevant player and ball tracks using a rule-based query. Each high-level event in the following section is detected offline, which allows users to retrieve complex high-level queries in a matter of seconds.

6.9.1 High-level Query Generation

Having detected the strokes played (inertial sensors), along with the player and ball tracks (visual sensors), we can detect the movement of the ball after a specific stroke is executed by a given player. An offline rule-based query then detects if the serve is a **T, Body or Wide**. A T serve is when the ball intersects with the middle of the T zone in the opponents service box after it has bounced in the service box. A body serve is when the ball crosses the opponents service box at roughly the middle of the box. A wide serve is when the ball exits the opponents service box at the close to the court boundaries. A similar rule-based query detects if forehand is a **in-to-in, in-to-out, cross, line**. A forehand in-to-out originates from the players left side of court and the ball then travels diagonally across the court into the

opponents side of the court. Forehand line originates from the players right side of court and the ball continues down the right side into the opponents half of court. Forehand cross is when the ball is struck from the players right side of court and the ball then travels diagonally across the court into the opponents side of the court. A similar coordinate system is used to find backhand cross and backhand line strokes.

A rule based query engine is executed offline to identify **first serves made/missed**. This module uses heuristics based on the server's movements immediately after a serve. The rules of tennis state that if the first serve is illegal, the second serve must be taken from the same side of the baseline. Therefore if a serve is executed and the servers location remains on the same side of the baseline for the next serve, then we can infer that the previous serve was missed, otherwise the first serve was made. Following on from the knowledge of first serves made/missed, the strokes played by the returning player are analysed offline to infer the **return of first serves** and **return of second serves**. A rule-based query identifies which player is returning and what type of stroke is being executed.

Ball tracks are used to detect which strokes result in the ball hitting the **net**. This query, which is again executed offline, simply records the origin of a each stroke played. If the ball track terminates in what is defined as the net region, it is assumed that the ball has hit the net.

A **volley** is usually played when a player returns the ball, while positioned close to the net, hence the ball does not have time to hit the ground and is volleyed. To detect forehand and backhand volleys, a query detects forehands and backhands with inertial sensors and then cross examines each detected stroke with the relevant player track to infer if the player is within volley range of the net. A **smash** stroke will have a player executing the same motion as a serve except the smash will be played during a rally, un-

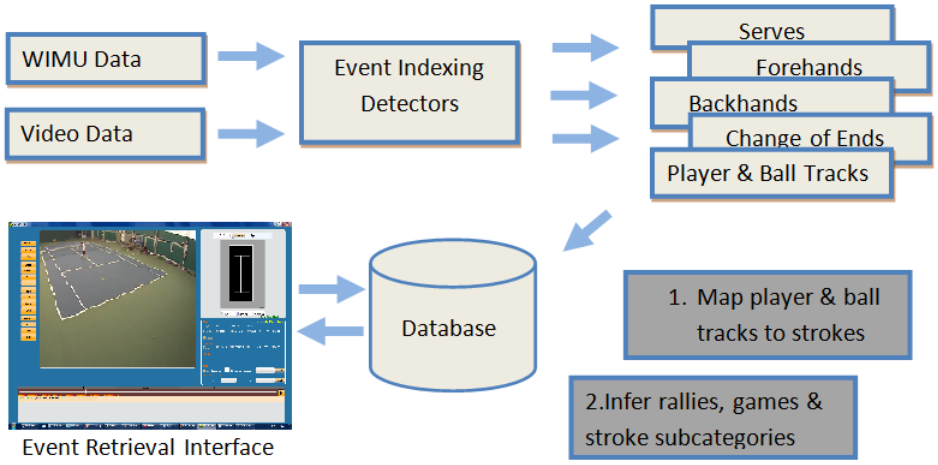


Figure 6.14: Event Retrieval & Detection Overview

like a serve which is always the first stroke in a rally. To detect a smash, a rule-based query analyses the inertial stroke recognition data and labels any strokes which are classified as a serve, but occur while the player is positioned inside the baseline as a smash.

6.9.2 Evaluation

There is no published work which detects strokes using inertial sensors, therefore to compare our approach to [127], we selected the same six events as reported in [127] to be recognised: forehand, backhand, service, smash, forehand volley and backhand volley. 240 sequences were selected, which are performed by 5 players during real matches. Three of the players we used are advanced players and the remaining two were intermediate and beginner level. Inertial sensors are used to detect serves, forehands and backhands. Forehand volley, backhand volley and smashes were detected as described in Section 6.9.1. Stroke recognition classifiers were trained with one group of players and the evaluation set contained strokes from unseen players as described in Section 6.6.2. Although not directly comparable to [127], our recognition results for this experiment are 82% which is in line with the re-

sults found in [127]. Where an event was detected to within 1 second of the correct event time it was deemed correct. Furthermore, our dataset contains novice players, unlike in [127] where only professional players from broadcast video are used. This proves that our system can work on players of all levels. This stroke recognition system was evaluated on an Intel Core 2 Duo Processor and the combined training and recognition time of a single event took on average 7 seconds.

Event	**Accuracy**
Serve	93
Forehand	81
Backhand	78
Forehand Volley	72
Backhand Volley	75
Smash	91

Table 6.6: This table gives the accuracy results for the experiment in Section , where 40 instances of each event from 5 different players are detected

6.10 Conclusion

This chapter presented novel methods to detect key tennis events using either visual or inertial sensors. Related work on automatic event detection in sport, using either visual or inertial sensors was first outlined. After this, a tennis ontology was introduced which defines the key events in tennis. We then introduced the sensor instrumentation and event detection methodology for detecting key events in a court based racquet sport, such as tennis. It is also possible that these event detection algorithms could be used for similar sports, such as badminton or pickleball. The experiments sections evaluated the algorithms ability to detect the key events and we are very encouraged by the accuracy of both the inertial and visual detection algorithms.

One of the aims of this thesis is to assess if automatic event recognition can be beneficial to sports coaching tools. The following chapter introduces

a novel automatic sports coaching solution for the sport of tennis that builds upon the event detection framework described in this chapter.

Chapter 7

Match Point: A Visual Coaching System

7.1 Introduction

In this chapter we present a video analysis tool for tennis coaching which coaches can use to query and retrieve the results of tennis event detection presented in the previous chapter. This chapter also explores if automatic event detection can benefit tennis coaches. Existing tennis coaching software lacks the ability to automatically index a tennis match into key events and therefore a coach who uses existing software is burdened with time consuming manual video editing. The proposed automatic video analysis system can be used to coach from beginners upwards and allows coaches to build advanced queries, which existing video coaching solutions cannot facilitate, without tedious manual indexing. The user interface provides a novel user query panel which coaches can use as a graphical query tool to retrieve and playback video of strokes played by a player from a specific region of interest on the court. Matches are grouped by players so coaches can retrieve videos of particular events from multiple matches for video feedback and tactical

analysis.

7.2 User Interface

In this section we present the coaching system which coaches and players can use to playback tennis events and analyse play statistics. The system is called Match Point and the user interface (Figure 7.1) consists of three main panels. The *Match Timeline* displays all the matches played by a given player. Each time line represents a single match. The *User Query Panel* allows users to draw a rectangle for each player and also a stroke direction. The user can then retrieve all strokes which are played while both players are simultaneously inside their respective rectangles. The *Events Panel* provides an interface for users to build specific queries related to stroke patterns such as "play each video instance in the match where Player A performs a "first serves missed" event", for example. When a user searches from either the user query panel or the events panel all the results are displayed along the match time line and can be played in the video screen by clicking on that event.

7.2.1 Automatic Match Indexing

After the match has finished, the videos are ingested into the system simply by dropping them into the ingest folder. Additional match information is manually entered into the system such as the full names of both players. If either of the two players have been recorded by the system before, the user selects this player from the existing players menu. No further user input is required and the user can activate the automatic event detection algorithms by clicking a button. The algorithms are executed offline and can take several hours to complete depending on the duration of the match.

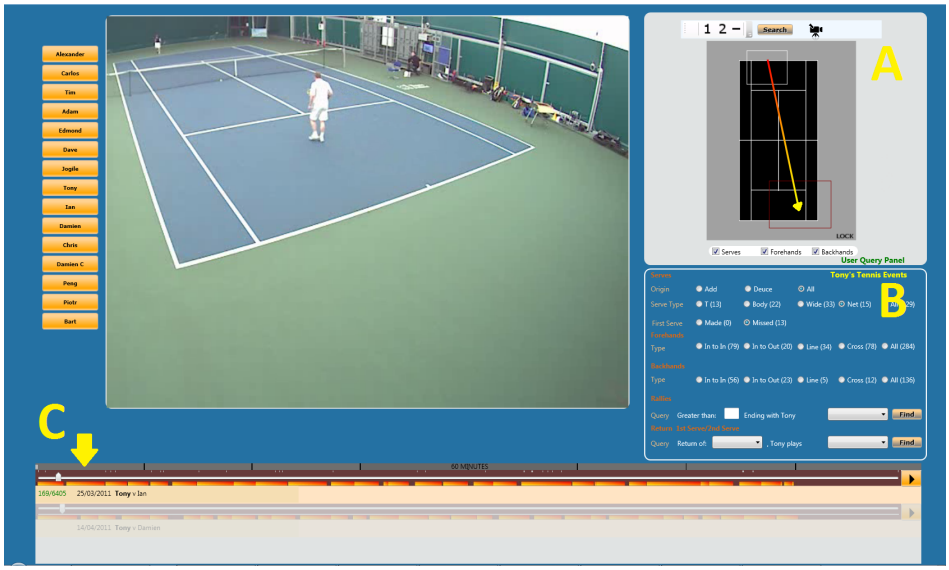


Figure 7.1: Match Point Event Retrieval System, (A) User Query Panel (B) Event panel to retrieve events (C) Match timeline panel is used to display events.

7.2.2 Match Timeline Control

To populate the match timeline with matches from a respective player, the user selects a player from the player panel on the left side of the interface. In the event that a player is captured in multiple matches, multiple match timelines will appear below each other. Each match timeline also visualises all the games in a match. Each game is represented by a brown rectangle along the match timeline. The length of each game is clearly visible and coaches can focus on events which occur on short or long games with this feature. Additional information is also available here such as the player names, match date and match duration. When a user selects specific events from either the User Query Panel (Section 7.2.4) or the Events Panel (Section 7.2.3), all the retrieved events are displayed along the selected match time line control. Video playback is only possible by selecting an event from a match time line panel.

7.2.3 Events Panel

The events panel provides an interface for users to view match statistics and playback video of each event in the panel. For example, the user might want to view the video of all the T-Serves executed by *Player A* from the left side of the baseline. Each event retrieved will then be represented along the match timeline as a vertical tick and the user can click on the relevant event tick along the match timeline to playback the video of the event.

The events panel provides many analysis statistics and video playback of each event in this panel is instantaneous. The frequency of T, body and wide serves from the deuce and add side of the baseline is visible, which can be used to gauge if a player is under/over using a particular serve. This level of pattern finding in a player's play is easily highlighted with the event counter, which counts the frequency of each event as shown in Figure 7.2. In addition to serve subtypes, the event panel displays how many first serve events a player has made and missed and how may of these were T, body or wide serves, or those serves which hit the net.

The number of times a player performs a forehand in to in, in to out, cross, line or a backhand cross or line during a match can be analysed in the events panel. The events panel also contains the number of forehands and backhands which hit the net. Another useful playback feature is to find rallies which contain more than n shots where a given player ends the rally by hitting a particular stroke, where n is defined by the coach before searching. Finally the events panel can quickly playback a player's return of first or second serve during a given match. Video playback of all the aforementioned tennis events is possible by clicking on the relevant button in the events panel. This will populate the match timeline with each event and the coach can playback any event by scrolling through the match timeline.

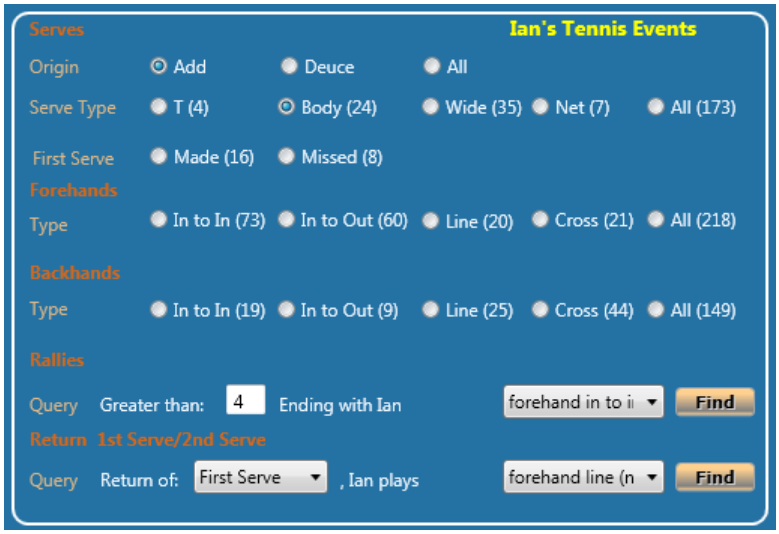


Figure 7.2: Sample Events panel showing individual player statistics over a single match. The indices represent how often in the match the given player performs the specific event.

7.2.4 User Query Panel

An important contribution of this system is the User Query Panel, which allows users to visually construct a query by drawing rectangles and line objects which represent both players' locations and the ball in flight. The user can then retrieve all strokes which are played from a given region of interest where the ball travels along a user specified shot line. It is also possible to filter the results returned by any combination of serves, backhands and forehands. Results are displayed along the match timeline for efficient retrieval. Analysing the frequency with which players perform specific strokes from a given region on the court has been highlighted as a potentially useful feature by the tennis coaches we interviewed.

The process to find all strokes played by a given player from a particular region on the court is as follows. A coach draws a rectangle anywhere on the court as shown in Figure 7.3, its possible to draw one rectangle or two to find strokes played while both players are inside their respective rectangles.

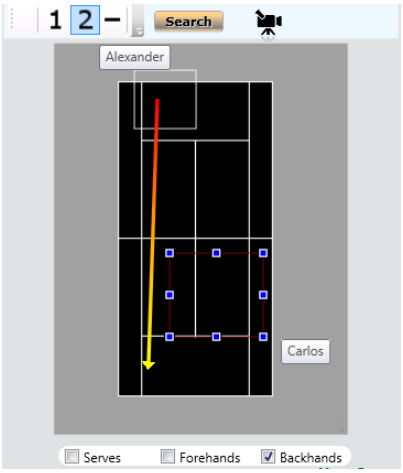


Figure 7.3: User Query panel detects number of times a player performs a stroke from a given region of the court.

Finally the flight of the ball is inferred by drawing a ball line on the panel. Stroke types may be filtered using options as shown in Figure 7.3. The experiments in the next section illustrate how this feature performed in a user study.

7.3 User Study

This section evaluates the practicality of the overall system for tennis coaching. This user study was designed to commit users to (1) understand how the system works, (2) understand what features are available in the system and finally for the users (3) to use all the features of the system in order to give them a subjective view of how practical the system is. All experiments were videotaped to compile accuracy scores of the users ability to complete the test queries. After users had completed all experiments they were asked to anonymously complete a questionnaire. By making the questionnaire anonymous, users are given an unbiased platform to express their opinions on the overall system experience.

To evaluate the system, we performed a number of experiments with

ten users, six of whom were tennis coaches and four were regular tennis players. The experiments evaluated all the components of the interface and a comparison of the user interface was made with Dartfish, which is a state of the art video analysis application for sport. Retrieval experiments were conducted in a room equipped with two desktop PCs, each with a 23 inch monitor, with an instructor present. Each user completed a training session on our system in advance. A similar level of training was then provided for Dartfish, where necessary.

It should be noted that five of the six tennis coaches are Dartfish experts and two of the tennis players regularly use Dartfish. Four of the coaches were professional coaches and two coach in local tennis clubs. The four regular tennis players were recruited from a local tennis club. There are currently no automatic event retrieval systems for sports coaching, so we compare our system to Dartfish, which is the most popular sports coaching analysis tool and has functionality for manual event annotation. Problems or questions on how to use each system were discussed during the training sessions.

7.3.1 Experiment One: Event Panel Evaluation

The aim of the first user experiment was to provide the users with a hands on experience of using the event panel in Figure 7.2 to retrieve various match events, which coaches regularly review for learning purposes. The users were given a task list of five events to retrieve using the events panel in the Match Point system. The events to be retrieved are shown on the next page and were chosen by analysing which tennis events coaches have manually tagged using Dartfish in the past. Therefore each of the events to be detected was previously manually indexed by coaches using Dartfish in the past. A questionnaire was used to assess the user's experience and to assess whether coaches and players found using the event retrieval system beneficial for

learning purposes. The outcome of this experiment is discussed in Section 7.3.5.

1. First Serves Made and First Serves Missed from the left or right side of the baseline. The user must also observe if the serve is a T Serve, Body Serve,Wide Serve or net.

2. Return of first serve and return of second serve. The user must also observe if the return is a backhand cross, backhand line, forehand in to in, forehand in to out, forehand cross or forehand line.

3. Rallies, which are greater than four strokes and end with Player A playing a forehand in to in.

4. Rallies, which are greater than four strokes and end with Player B hitting the net whilst playing a backhand cross.

5. Rallies, which are greater than four strokes and end with Player A playing a forehand in to out.

7.3.2 Experiment Two: User Query Panel

This experiment aimed to investigate whether users can utilise the dynamic drawing features of the User Query Panel to accurately retrieve instances where a player performs a stroke played from a specified region of interest on the court. This experiment was carried out with ten users (six coaches and four players). To correctly retrieve an event, the user was shown an animation of a tennis court with two boxes which represented general locations of each player and a line to represent the ball trajectory. The user then had to construct the query on the User Query Panel. Each user was first trained on how to use the user query panel and then completed two training queries before being asked to retrieve the three queries below. When the results

were displayed on the match timeline panel, each user was then asked to view each of the retrieved results for validation.

1. Find all serves played by Player A whilst Player B is located within a specified region.

2. Find all serve returns by Player B, which are backhands played from a given region of interest on the court.

3. Find all backhands which traveled a particular direction and were played by Player A from a given region of interest.

7.3.3 Experiment Three: Event Retrieval - Match Point vs Dartfish

The aim of the this experiment is to assess, in terms of system usability, which coaching interface is the most efficient user interface for event retrieval in the context of tennis video analysis. For this experiment, we prepared Dartfish by manually tagging two matches fully with the five events from Experiment 1. Each user was then trained how to perform event retrieval with Dartfish and then asked to retrieve a random event from a random match. We stress that in this experiment we are only evaluating the Usability of the user interface between both systems. No existing coaching software automatically annotates video into tennis events so we can only compare system Usability, in terms of how well the interface performs, which is a very important factor in how well an athlete will assimilate information.

7.3.4 Evaluation Questionnaire

When each user completed all the relevant experiments, they were asked to fill in the following user questionnaire. These questions are designed to assess how well Match Point performs for coaches and tennis players alike.

All questions were answered by a score of how true the statement is, where 0= false, 1 = almost false, 2 = weak false, 3 = weak true, 4 = almost true, 5 = true. Questions 8 to 10 are relevant to coaches only and the results of this questionnaire are shown in Figure 7.4. The questions are designed to collectively obtain information on the aspects being evaluated in the user assessments. These questions resulted in a large amount of quantitative data which was obtained from an online survey website. There is no direct relationship between the users and authors and because the questionnaires were answered anonymously there is a good environment for users to give a critical evaluation of the overall system.

1. Learning to operate Match Point is straightforward.
2. Exploring new features by trial and error is straightforward with Match Point.
3. Match Point can reliably detect specific tennis events, which are queried by the user.
4. Match Point's Query Court panel is efficient at finding strokes played from a given region of interest on the court.
5. I found event retrieval easy with match Point.
6. It is easier to visualise tennis events using Match Points timeline control, game bar and event tick, than with the Dartfish application.
7. I prefer to use Match Point instead of Dartfish for tennis video analysis.
8. Using the system would improve my performance.
9. Using the system would enhance my effectiveness on the job.
10. Using the system would make it easier to do my job.

Usability Evaluation

As an additional part of the user questionnaire, we evaluate the user interface using Nielsen's Heuristic Evaluation, which is one of the most common usability heuristics for user interface design [121]. We selected six heuristics for Usability evaluation and these were deemed most relevant to the user interface. The experimental setup for this study was as follows: when each evaluator had concluded all their system training , evaluation exercises and were therefore familiar with the user interface, they were asked the following questions which were to be answered by applying a score of how true the statement is (0 = false, 5 = true)0= false, 1 = almost false, 2 = weak false, 3 = weak true, 4 = almost true, 5 = true, where . All ten test users answered this test and the results are shown in Figure 7.4 and discussed in the next section.

1. Visibility of system status. The system always keeps users informed about what is going on, through appropriate feedback within reasonable time.

2. Match between system and the real world. The system speaks the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. It follows real-world conventions, making information appear in a natural and logical order.

3. Recognition rather than recall. Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

4. Consistency and standards. Users should not have to wonder whether

different words, situations, or actions mean the same thing. Follow platform conventions.

5. User Feedback. The user receives appropriate system notification in response to user actions.

6. Help users recognize, diagnose, and recover from errors. Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

7.3.5 User Study Evaluation

Several experiments were used to assess individual system components. If the experiments were only examined on a small number of people the reliability of individuals could be questioned, so increasing the number of testers to ten reduces the ability of single testers to have a significant impact on the final accuracy score. However, the testers were trained on how to use individual system components directly before the test, so that they were familiar with each component.

In experiment one, the events to be retrieved using our system were selected by observing what events coaches have manually annotated using Dartfish in the past. Our system can automatically annotate all the events used in this experiment. Therefore, the first benefit of using Match Point is that using our system will reduce the time a coach will spend manually annotating videos of matches. The accuracy for Experiment 1 in Table 7.1 is a measure of how often users successfully completed each test within the allocated one minute time frame. When a user was unable to complete the experiment within the allocated time or had to ask for assistance the experiment was marked as a failure.

Coaches unanimously agreed that counting the frequency of stroke subcategories is a significant step forward for video coaching tools as it would

simply take endless hours to manually tag this level of detail and this statistic has not been previously collected, to the best of our knowledge. For example, the frequency of T, body or wide serves performed by a specific player has been identified as an interesting statistic by the system users because it allows coaches to find patterns in how often players are executing each style of serve. We also count the number of times each serve type is executed within the deuce or add side, as baseline positioning whilst serving has been highlighted as an interesting statistic by our coaches during requirements gathering.

Experiment two evaluated the User Query Panel, which allows users to build queries of a particular stroke, executed from a specific region of interest on the court. If the user correctly constructed the query using the graphical tools and therefore successfully retrieved the relevant events after running the query, the result was deemed correct, otherwise the query result was incorrect. It was found that after sufficient training, coaches and players alike were easily able to build queries on the query panel and then playback the video of each returned result on the match timeline with ease. Since each result is represented by a vertical line on the match timeline, the user can gauge how often this stroke was performed from the given region. The retrieval results in Table 7.1 highlight some issues where the users did not retrieve the correct results and this typically occurred when a user did not draw the ball line along the correct plane.

Experiment	**No. Events**	**Mean Accuracy**
1	5	.83
2	3	.77

Table 7.1: Match Point user retrieval Experiments 1 & 2

The event retrieval task in Experiment Three highlighted that our system is designed for inter and intra match analysis, whereas Dartfish is not,

Heuristic	Score
1	80%
2	90%
3	95%
4	80%
5	50%
6	40%

Table 7.2: Nielsen's Usability Evaluation

as new Dartfish users struggled to load different matches with Dartfish. Our system has several features which make it more attractive for video retrieval, such as the easy-to-use time line, the game boundary marker and the vertical line which represents an event along the match timeline and users have reflected these observations with the user questionnaire. To obtain the Match Point accuracy scores for Experiment Three, users were allocated one minute to retrieve each event using Match Point and if they were unable to complete the experiment within the time frame or required assistance, the test was recorded as a failure. The user was then requested to find the same events from particular matches using Dartfish. Users who were familiar with Dartfish passed this experiment with ease, but new Dartfish users, who had received the same amount of training as was granted to Match Point's training exercises, struggled with Usability issues, such as loading the events of match for the first time. The accuracy for this experiment using Dartfish was 71%, whilst Match Point achieved a score of 74%. Table 7.3 illustrates how accurately the users correctly retrieved each event using the different sensors.

The user questionnaire results in Figure 7.4 conclude that coaches, some of whom are Dartfish experts found Match Point significantly better for event retrieval. All users preferred Match Point's user interface over the leading video analysis software for sport. However Match Point's user interface is designed specifically for tennis, while Dartfish is designed for all sports. As

Event	Dartfish Accuracy (%)	Match Point Accuracy (%)
1	75	81
2	67	68
3	71	72
4	63	74
5	77	77

Table 7.3: Event retrieval results - Match point v Dartfish (experiment 3)

the evaluation questionnaire results in Figure 7.4 highlights, users found that using Match Point is more straightforward with the customised layout of events. What was also encouraging was the discovery that all users were able to build informative queries such as those in Experiment One and Two after only a short duration of system training and the accuracy in event retrieval is shown in Table 7.1.

The Usability evaluation in Section 7.3.4 assessed the user interface, in this experiment we selected six relevant heuristics and each user completed the evaluation after using the system. The scores obtained from the first four heuristics are encouraging, however the low scores for heuristics five and six can be attributed to this system being a prototype and therefore it will be a future work to improve user feedback and error messaging. Improvements in these areas would significantly help the system to become more acceptable to the users in a production environment and the findings of this experiment are helpful for future work.

7.3.6 Tactical Analysis

An additional evaluation of the tactical analysis was conducted to highlight what coaches can learn from Match Point that may give them novel insight. Since our system has the ability to record every stroke sub category and also records the location where each stroke was played, along with the ball direction of the stroke, we use this information over multiple matches to gain

insightful statistics on a given players tactical patterns. Example findings which were made by our users were to highlight where a given player has missed a high number of first serves during the first game of a match. Another tactical statistic was that we were able to see that one of our advanced players overuses the Wide serve from the deuce side of the baseline. This pattern in a serve could help an opponent, if they became aware of it. This is because they would have a competitive edge in this scenario, knowing what serve to expect and therefore standing in the optimal position to play the best return stroke.

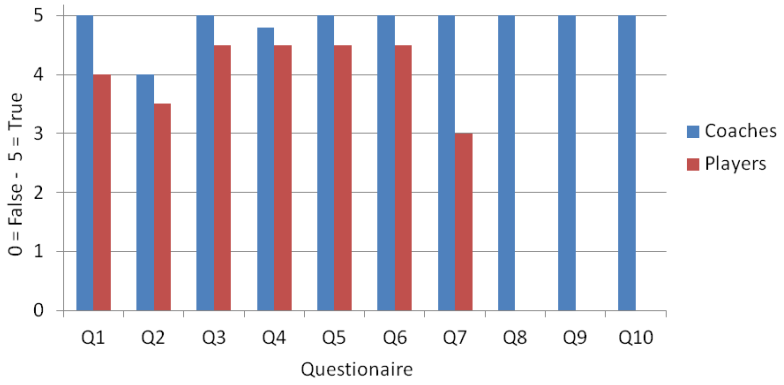


Figure 7.4: User study with six coaches and four players.

7.4 Discussion

A comprehensive event retrieval interface was presented and coaches may retrieve events using multiple interface components. It was agreed by the tennis coaches that they prefer Match Point as an event retrieval interface when compared to a leading state of the art sports coaching software. The

experiments have also proved that Match Point can infer knowledge which is simply not possible with manual annotation without applying significant human effort. For example, for every serve, the system can infer if it is T, Body or Wide. The system can also index every rally in the match, including the rally length. First serves made, first serves missed and return of first serves/return of second serves are also automatically indexed. In conclusion, we found that sports coaching software could be significantly enhanced by utilising automatic event recognition techniques, as proposed in our system.

The event retrieval task in Experiment Three highlighted that our system is designed for inter and intra match analysis, whereas Dartfish is not. Our system has several features which make it more attractive for video retrieval, such as the easy-to-use time line, the game boundary marker and the vertical line which represents an event along the match timeline. The user questionnaire results in Figure 7.4 conclude that coaches, who are Dartfish experts found Match Point significantly more better for event retrieval. All users preferred Match Point's user interface over the leading video analysis software for sport. However Match Point's user interface is designed specifically for tennis, while Dartfish is designed for all sports. As the questionnaire in Table 7.4 highlights, users found that using Match Point is more straightforward with the customised layout of events. What was also encouraging was the discovery that all users were able to build informative queries such as those in Experiment One and Two after only a short duration of system training.

Part IV

Conclusions

Chapter 8

Conclusions

The research reported in this thesis examined the use of visual and inertial sensors in sports coaching tools. We evaluated event detection algorithms for detecting human actions using both visual and inertial sensors. The algorithms were then adopted for use in detecting events in tennis and from these events a fully automatic event retrieval system was designed. The following sections reviews the key aspects of each chapter.

8.1 Chapter Summary

In **Chapter 1** we introduced our thesis, providing a brief overview, hypothesis, motivation and a list of central questions explored throughout the research carried out. We discussed existing research trends in the area of sports coaching software and identified different types of existing coaching tools, in particular analysis tools, video coaching tools and instrumented coaching environment software solutions. The limitations of video coaching tools and instrumented coaching tools are then identified and from these limitations we developed our research objectives. The research objectives were to enhance the coaching experience by automatically indexing key tennis events using visual and inertial sensors and then presenting this information

to coaches through an interface which they can use to efficiently retrieve interesting key points during a match.

Chapter 2 described the technical background on visual and inertial sensing before reviewing literature for machine learning. The final section in this chapter looks at state of the art machine learning techniques, which are used for event classification.

Chapter 3 described existing work used to detect human actions using visual sensors. We introduced two approaches for human action recognition using video (MHIHOGs and contour features) and both were evaluated against an indoor dataset and an aerial view outdoor dataset of basic human actions. We also benchmarked the algorithms against the popular Weizmann dataset. The finding from this chapter was that contour features are highly accurate for event detection of human actions and they are also computationally inexpensive, making them a suitable choice in this work.

In **Chapter 4**, we explored the challenges which arose when using inertial sensors to recognise human actions. This was followed by introducing the technique used in this work to recognise human actions using inertial sensors. Experiments were conducted to evaluate recognition accuracy and then we fused different inertial sensors together and evaluated both early and late fusion techniques. Our investigations concluded that the best accuracy for recognising human actions is to use early fusion of multiple sensors.

Chapter 6 took the visual and inertial sensing algorithms detailed in the earlier chapters and modified these event recognition techniques for detecting events in the sport of tennis. We determined an achievable set of tennis events to be automatically detected by analysing existing coaching practices and interviewing tennis coaches. The chapter explained the process used to detect tennis events from each sensor in turn and experiments were conducted on each modality to determine the accuracy of each sensor.

The findings from this chapter were that early fusion of inertial sensors was more accurate at detecting tennis events than visual sensors. However, visual sensors can detect ball movements which provides a wealth of data for interesting queries based on ball locations during a match.

Chapter 7 described the novel content management and retrieval system. This retrieval system allows coaches to generate advanced queries which would not previously have been possible, without an inordinate undertaking of manual annotation on the part of a tennis coach or tennis expert. An evaluation study evaluated if such a system can significantly improve coaching techniques and also how this next generation coaching software compares to a state of the art commercial coaching tool. The finding from this chapter is that automatic indexing is beneficial to tennis coaching.

8.2 Thesis Contributions

This thesis makes several contributions. The most significant contribution is the introduction of an early fusion technique to automatically detect human actions using a single inertial sensor attached to a human subject's wrist. Using early fusion we fused accelerometer, magnetometer and gyroscope sensor data to detect what actions the human subject is performing. We first applied this technique to recognise human actions in Chapter 4. By adapting the early fusion approach for human actions, this thesis also makes a contribution of event detection in tennis using early fusion of accelerometer, magnetometer and gyroscope sensors. This was achieved using the same fusion technique as that used to infer human actions with only small configuration changes (as outlined in Chapter 6) and in both applications early fusion increases action recognition by almost 10% (compared to single sensor classification) when tested on humans who are not present in the training set.

In Chapter 6, a new approach vis introduced to visually detect tennis events. This algorithm infers tennis events by recording a tennis match using only three cameras. From the videos of two cameras positioned behind the baseline and a third overhead camera, we introduce a novel approach to detect all tennis strokes and recognise what stroke is played. Rallies, games and change of end events are also detected using video. The overhead camera tracks both players and the ball during competitive play and these tracks can be used to infer further tennis events such as stroke subcategories. This level of detail has never previously been automatically detected before.

A third contribution is the novel content management and retrieval system (Match Point), which allows users to ingest the sensor information for a particular match and provides a user interface that can query the system to find events. This system can automatically index the key events in tennis using either visual or inertial sensors, or a combination of both. The query engine which this system provides allows users to run queries which are simply not possible without automatic indexing of key tennis events.

8.3 Future Work

The experiments and studies performed in this thesis suggest there are several areas which merit further research. This section explores these areas which may lead to profitable lines of inquiry. It will be difficult for automatic event detection algorithms to detect all the events a coach is interested in indexing, therefore a semi-automatic approach to event indexing may need to be explored. In such an application, automatic event indexing would be performed first and these events would provide a foundation, on top of which manual event indexing would be performed afterwards. This would certainly allow coaches to index forced/unforced errors, which is a statistic that coaches currently index with existing manual coaching tools. Detecting

these events involves a subjective view from an expert and it is difficult to automatically detect these events with computational learning approaches.

Coaches have also recommended that automatic detection of the scores at all times would enable them to infer interesting events during important parts of the matches. For example, a coach is more interested in analysing a serve which hits the net when the game score is even in the last set of the game, rather than at the beginning of a game when the player is not under as much pressure. Automatic score detection may be achieved using a camera at each side of the court to detect if a ball has landed within the legal boundaries. Knowing if a stroke was executed properly and landed within the legal court boundaries is the foundation upon which a successful score detection system would be built. This is achievable although the low cost cameras used in our research are not of sufficient quality to detect ball bounce.

Currently our event detection algorithms take several hours to complete. Computational optimisation of the techniques could significantly reduce the time delay required for automatic event detection and may open up the possibility of realtime event detection. Reducing the duration for event detection to within one hour would be a significant step forward for this system as this would allow coaches to analyse a matches played on the same day.

Differential GPS offers the potential to track players outdoor to within 10cm. There may be scope to use this approach to track outdoor tennis players and integration with our system would allow us to automatically track a player, as well as inferring strokes played, rallies and games using just a WIMU and differential GPS. This would mean that a large amount of tennis events could be automatically detected in an outdoor match at any location without assembling the camera infrastructure, which is time

consuming.

8.4 Final Word

The work in this thesis examined existing sports coaching tools for racquet sports and specifically tennis. We identified limitations with existing solutions and from here identified key tennis events which could be automatically indexed to reduce the time spent manually indexing sports matches. Event indexing algorithms were created for visual and inertial sensors and the different approaches used for detecting events from each sensor were evaluated against multiple datasets. A novel coaching user interface was assessed which allowed coaches to analyse the automatically detected events and a user study concluded that automatic event detection can significantly improve the coaching experience for racquet sports. We believe that automatic indexing is an exciting and challenging research topic and that the work reported in this thesis constitutes a useful contribution to the state of the art.

References

[1] K. T. Abou-Moustafa, M. Cheriet, and C. Y. Suen. Classification of time-series data using a generative/discriminative hybrid. In *Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition*, IWFHR '04, pages 51–56, Washington, DC, USA, 2004. IEEE Computer Society.

[2] David W. Aha, Dennis Kibler, and Marc K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991. 10.1007/BF00153759.

[3] Amin Ahmadi, David Rowlands, and Daniel Arthur James. Towards a wearable device for skill assessment and skill acquisition of a tennis player during the first serve. In *Sports Technology, Volume 2, Issue 3-4*, pages 129–136, 2010.

[4] K Aminian, P Robert, E E Buchser, B Rutschmann, D Hayoz, and M Depairon. Physical activity monitoring based on accelerometry: validation and comparison with video observation. *Medical and Biological Engineering and Computing*, 37(3):304–308, 1999.

[5] Adrienne Andrew, Yaw Anokwa, Karl Koscher, Jonathan Lester, and Gaetano Borriello. Context to make you more aware. *27th International Conference on Distributed Computing Systems Workshops ICDCSW07*, pages 49–49, 2007.

[6] U Anliker, Jamie A Ward, Paul Lukowicz, Gerhard Troster, F Dolveck, M Baer, F Keita, E B Schenker, F Catarsi, L Coluccini, and et al. Amon: a wearable multiparameter medical monitoring and alert system. *IEEE transactions on information technology in biomedicine a publication of the IEEE Engineering in Medicine and Biology Society*, 8(4):415–427, 2004.

[7] M Ardebilian, L Chen, and X Tu. Robust 3d clue-based video segmentation for video indexing. *Journal of Visual Communication and Image Representation*, 11(1):58–79, 2000.

[8] D. K. Arvind and Andrew Bates. The speckled golfer. In *BodyNets '08: Proceedings of the ICST 3rd international conference on Body area networks*, pages 1–7, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[9] R.P. Aylward, Massachusetts Institute of Technology. Dept. of Architecture. Program in Media Arts, and Sciences. *Sensemble: a wireless inertial sensor system for the interactive dance and collective motion analysis*. Massachusetts Institute of Technology, School of Architecture and Planning, Program in Media Arts and Sciences, 2006.

[10] Ling Bao and Stephen S Intille. Activity recognition from user-annotated acceleration data. volume 3001, pages 1–17. Springer, 2004.

[11] Lluis Barceló and Xavier Binefa. Contextual soccer detection using mosaicing techniques. In *Proceedings of the Second Iberian conference on Pattern Recognition and Image Analysis - Volume Part I*, IbPRIA'05, pages 77–84, Berlin, Heidelberg, 2005. Springer-Verlag.

[12] Michael Barry, Juerg Gutknecht, Irena Kulka, Paul Lukowicz, and Thomas Stricker. From motion to emotion: a wearable system for the multimedia enrichment of a butoh dace performance. *J. Mob. Multimed.*, 1(2):112–132, June 2005.

[13] R Bartlett, C Button, M Robins, A Dutt-Mazumder, and G Kennedy. Analysing team coordination patterns from player movement trajectories in soccer: methodological considerations. *International Journal of Performance Analysis in Sport*, 12:398–424, 2012.

[14] Jean-Charles Bazin, Inso Kweon, Cedric Demonceaux, and Pascal Vasseur. Improvement of feature matching in catadioptric images using gyroscope data. *2008 19th International Conference on Pattern Recognition*, pages 1–5, 2008.

[15] Guillaume Becq, Stéphane Bonnet, Lorella Minotti, Michel Antonakios, Régis Guillemaud, and Philippe Kahane. Classification of epileptic motor manifestations using inertial and magnetic sensors. *Comput. Biol. Med.*, 41:46–55, January 2011.

[16] Alain Berthoz. *The Brain's Sense of Movement.* Havard University Press, 2000. ISBN: 0-674-80109-1.

[17] Niranjan Bidargaddi, Antti Sarela, Lasse Klingbeil, and Mohanraj Karunanithi. Detecting walking activity in cardiac rehabilitation by using accelerometer. *Heart*, pages 555–560, 2007.

[18] Michael J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *ICCV*, pages 231–236, 1993.

[19] T. Bloom and A.P. Bradley. Player tracking and stroke recognition in tennis video. In *VSSN*, pages 93 – 94, 2006.

[20] T. Bloom and P. Bradley. Player tracking and stroke recognition in tennis video. In *Proceedings of the WDIC*, pages 93–97, 2003.

[21] G R Bradski and J Davis. Motion segmentation and pose recognition with motion history gradients. In *Proceedings Fifth IEEE Workshop on Applications of Computer Vision*, volume 13, pages 174–184. Springer, 2000.

[22] Jonathan Brookshire. Person following using histograms of oriented gradients. *I. J. Social Robotics*, 2(2):137–146, 2010.

[23] R.G. Brown. *Introduction to random signal analysis and Kalman filtering*. Wiley, 1983.

[24] Muhammad Akmal Butt and Petros Maragos. Optimum design of chamfer distance transforms.

[25] Q Cai and J K Aggarwal. Tracking human motion in structured environments using a distributed-camera system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1241–1247, 1999.

[26] Chia-chih Chen and J K Aggarwal. Recognizing human action from a far field of view. *Soccer*, pages 1–7, 2009.

[27] H S Chen, H T Chen, Y W Chen, and S Y Lee. Human action recognition using star skeleton. *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks VSSN 06*, page 171, 2006.

[28] Hua-Tsung Chen, Wen-Jiin Tsai, and Suh-Yin Lee. Stance-based strike zone shaping and visualization in broadcast baseball video : Providing reference for pitch location positioning department of computer science , national chiao-tung university , hsinchu , taiwan. *Multimedia Tools and Applications*, 2(2):302–305, 2009.

[29] Hua-Tsung Chen, Wen-Jiin Tsai, Suh-Yin Lee, and Jen-Yu Yu. Ball tracking and 3d trajectory approximation with applications to tactics analysis from single-camera volleyball sequences. *Multimedia Tools and Applications*, 4842:387–396, 2007.

[30] Siyue Chen and Henry Leung. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):220, 2004.

[31] Lawrence Cheng and Stephen Hailes. Analysis of wireless inertial sensing for athlete coaching support. In *GLOBECOM*, pages 5269–5273, 2008.

[32] Ming-Ming Cheng, Guo-Xin Zhang, Niloy J Mitra, Xiaolei Huang, and Shi-Min Hu. Global contrast based salient region detection. *CVPR 2011*, pages 409–416, 2011.

[33] Tanzeem Choudhury, Matthai Philipose, Danny Wyatt, and Jonathan Lester. Towards activity databases: Using sensors and statistical models to summarize peoples lives. *IEEE Data Eng Bull*, 29(1):1–8, 2006.

[34] B. Clarkson and A. Pentland. Extracting context from environmental audio. *Wearable Computers, IEEE International Symposium*, 0:154, 1998.

[35] C Ó Conaire, D. Connaghan, P. Kelly, and N. O'Connor. Combining inertial and visual sensing for human action recognition in tennis. In *ACM Multimedia Workshop, Artemis*, 2010.

[36] C Ó Conaire, P. Kelly, D. Connaghan, and N. O'Connor. Tennissense: A platform for extracting semantic information from multi-camera tennis data. In *Digital Signal Processing, DSP 2009*, 2009.

[37] Damien Connaghan, Philip Kelly, Noel E. O'Connor, Mark Gaffney, and S. Cian Mathna. Multi-sensor classification of tennis strokes. In *Proceedings of the IEEE Sensors 2011*, pages 667–670. IEEE, 2011.

[38] Kenneth Conroy and Mark Roantree. Enrichment of raw sensor data to enable high-level queries. In *Proceedings of the 21st international conference on Database and expert systems applications: Part II*, DEXA'10, pages 462–469, Berlin, Heidelberg, 2010. Springer-Verlag.

[39] Sunny Consolvo, David W Mcdonald, Tammy Toscos, Mike Y Chen, Jon Froehlich, Beverly Harrison, Predrag Klasnja, Anthony Lamarca, Louis Legrand, Ryan Libby, and et al. *Activity sensing in the wild: a field trial of ubifit garden*, volume 08, pages 1797–1806. ACM, 2008.

[40] T Cover and P Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[41] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting objects, shadows and ghosts in video streams by exploiting color and motion information. In *11th International Conference on Image Analysis and Processing*, pages 360–365, 2001.

[42] R. Cucchiara, C. Grana, M. Piccardi, A. Pratii, and S. Sirotti. Improving shadow suppression in moving object detection with hsv color information. In *Proceedings of IEEE Intelligent Transportation System Conference*, pages 334–339, 2001.

[43] R. Cucchiara, C. Grana, A. Prati, and M. Piccardi. Detecting objects, shadows and ghosts in video streams by exploiting color and motion information. *Image Analysis and Processing, International Conference on*, 0:360, 2001.

[44] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR '05: Proceedings of the 2005 Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pages 886–893, 2005.

[45] James W. Davis and Aaron F. Bobick. The representation and recognition of human movement using temporal templates. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 928, Washington, DC, USA, 1997. IEEE Computer Society.

[46] G. de Haan, P.W.A.C. Biezen, H. Huijgen, and O.A .Ojo. True-motion estimation with 3-d recursive search block matching. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 3, pages 368–379, October 1993.

[47] Manolis Delakis, Guillaume Gravier, and Patrick Gros. Audiovisual integration with segment models for tennis video parsing. *Comput. Vis. Image Underst.*, 111:142–154, August 2008.

[48] Anind K. Dey. Understanding and using context. In *Personal Ubiquitous Comput.*, volume 5, pages 4–7, London, UK, January 2001. Springer-Verlag.

[49] I M L Donaldson. Movements of the eye, 2nd edition - carpenter. In *Nature*, volume 337, page 517, 1989.

[50] Ling-Yu Duan, Min Xu, Xiao-Dong Yu, and Qi Tian. A unified framework for semantic shot classification in sports videos. In *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, pages 419–420, New York, NY, USA, 2002. ACM.

[51] Ahmet Ekin, A Murat Tekalp, and Rajiv Mehrotra. Automatic soccer video analysis and summarization. *IEEE Transactions on Image Processing*, 12(7):796–807, 2003.

[52] Urs Enke. Dansense: Rhythmic analysis of dance movements using acceleration-onset times. Master's thesis, RWTH Aachen University, Aachen, Germany, September 2006.

[53] M Ermes, J Parkka, J Mantyjarvi, and I Korhonen. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *IEEE transactions on information technology in biomedicine a publication of the IEEE Engineering in Medicine and Biology Society*, 12(1):20–26, 2008.

[54] A Mfit Ferman and A Murat Tekalp. Efficient filtering and clustering methods for temporal video segmentation and visual summarization. *Journal of Visual Communication and Image Representation*, 9(4):336–351, 1998.

[55] M Franks and G Miller. Eyewitness testimony in sport. *Journal of Sport Behaviour*, 9:39–45, 1986.

[56] Hironobu Fujiyoshi and Alan J. Lipton. Real-time human motion analysis by image skeletonization. In *In Proceedings of IEEE WACV98*, pages 15–21, 1998.

[57] Mark Gaffney, Michael Walsh, and Brendan o'Flynn. Tennissense demonstrator rev 1.0 hardware manual, 2009.

[58] D.M. Gavrila. Sensor-based pedestrian protection. *IEEE Intelligent Systems*, 16(6):77–81, 2001.

[59] B Georis, M Maziere, F Bremond, and M Thonnat. A video interpretation platform applied to bank agency monitoring. In *Proceedings*

of IEEE Intelligent Distributed Surveilliance Systems, pages 46–50. IEEE, 2004.

[60] Hassan Ghasemzadeh, Vitali Loseu, and Roozbeh Jafari. Wearable coach for sport training: A quantitative model to evaluate wrist-rotation in golf. *J. Ambient Intell. Smart Environ.*, 1(2):173–184, 2009.

[61] Andrew R. Golding and Neal Lesh. Indoor navigation using a diverse set of cheap, wearable sensors. In *Proceedings of the 3rd IEEE International Symposium on Wearable Computers*, ISWC '99, pages 29–, Washington, DC, USA, 1999. IEEE Computer Society.

[62] Rafael C Gonzalez and Richard E Woods. *Digital Image Processing*, volume 49. Prentice Hall, 2008.

[63] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *Transactions on Pattern Analysis and Machine Intelligence*, volume 29, pages 2247–2253, December 2007.

[64] Marc Gowing, Philip Kell, Noel E. O'Connor, Cyril Concolato, Slim Essid, Jean Lefeuvre, Robin Tournemenne, Ebroul Izquierdo, Vlado Kitanovski, Xinyu Lin, and Qianni Zhang. Enhanced visualisation of dance performance from automatically synchronised multimodal recordings. In *Proceedings of the 19th ACM international conference on Multimedia*, MM '11, pages 667–670, New York, NY, USA, 2011. ACM.

[65] L Gu and T Kanade. A robust shape model for multi-view car alignment. *IEEE Conference on Computer Vision and Pattern Recognition (2009)*, pages 2466–2473, 2009.

[66] Stephen Hailes, Simon Julier, Dipak Kalra, and Tony Austin. Sesame: Sensing for sport and managed exercise. Technical report, 2006.

[67] D.L. Hall, R.J. Linn, and Llinas J. A survey of data fusion systems. In *Proc. SPIE Conf. on Data Structure and Target Classification*, volume 1470, pages 13–36. SPIE, 1991.

[68] D.L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[69] Zhenyu He and Lianwen Jin. Gesture recognition based on 3d accelerometer for cell phones interaction. *Bibliothek*, pages 217–220, 2008.

[70] David Heckerman. A bayesian approach to learning causal networks. In *In Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 285–295. Morgan Kaufmann, 1995.

[71] Ernst A. Heinz, Kai Kunze, Matthias Gruber, David Bannach, and Paul Lukowicz. Using wearable sensors for real-time recognition tasks in games of martial arts an initial experiment. In *in Proceedings of the 2nd IEEE Symposium on Computational Intelligence and Games (CIG*, pages 98–102. IEEE Press, 2006.

[72] Ernst A. Heinz, Kai S. Kunze, Matthias Gruber, David Bannach, and Paul Lukowicz. Using wearable sensors for real-time recognition tasks in games of martial arts - an initial experiment. In Sushil J. Louis and Graham Kendall, editors, *CIG*, pages 98–102. IEEE, 2006.

[73] Michael B. Holte, Cuong Tran, Mohan M. Trivedi, and Thomas B. Moeslund. Human action recognition using multiple views: a comparative perspective on recent developments. In *Proceedings of the 2011*

joint ACM workshop on Human gesture and behavior understanding, J-HGBU '11, pages 47–52, New York, NY, USA, 2011. ACM.

[74] Thanarat Horprasert, David Harwood, and Larry S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *Proceedings of the 7th IEEE International Conference on Computer Vision, Frame Rate Workshop (ICCV '99)*, pages 1–19, 1999.

[75] Ming-Kuei Hu. Visual pattern recognition by moment invariants. In *Information Theory IRE Transactions*, volume 8, pages 179–187. IEEE, 1962.

[76] W Hu, T Tan, L Wang, and S Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews*, 34(3):334–352, 2004.

[77] Chin-Pan Huang, Chaur-Heh Hsieh, Kuan-Ting Lai, and Wei-Yang Huang. Human action recognition using histogram of oriented gradient of motion history image. In *2011 First International Conference on Instrumentation Measurement Computer Communication and Control*, number 2, pages 353–356. IEEE, 2011.

[78] M Hughes and M Franks. Notational analysis of sport. *Oxon: Taylor and Francis*, pages 13–14, 2004.

[79] M Hughes and M Franks. The essentials of performance analysis. *Oxon: Taylor and Francis*, 13:8–13, 2008.

[80] M Hughes and R Meyers. Movement patterns in elite men's singles tennis. *International Journal of Performance Analysis in Sport*, 5:110–134, 2005.

[81] Tâm Huỳnh and Bernt Schiele. Unsupervised discovery of structure in activity data using multiple eigenspaces. In *Proceedings of the Second international conference on Location- and Context-Awareness*, LoCA'06, pages 151–167, Berlin, Heidelberg, 2006. Springer-Verlag.

[82] Myung Hwangbo, Jun-sik Kim, and Takeo Kanade. Inertial-aided klt feature tracking for a moving camera. In *Image Rochester NY*, pages 1909–1916. IEEE, 2009.

[83] D. Harwoodand L.S. Davis I. Haritaoglu. W^4: Who? when? where? what? a real time system for detecting and tracking people. In *International Conference on Face and Gesture Recognition*, April 1998.

[84] Hassan Ghasemzadehand Roozbeh Jafari. Multi-sensor activity context detection for wearable computing. In *Proc. EUSAI, LNCS*, 2010.

[85] Anil K. Jain. *Fundamentals of digital image processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[86] B H Juang and L R Rabiner. Hidden markov models for speech recognition. *Technometrics*, 33(3):251, 1991.

[87] M. Kam, Xiaoxun Zhu, and P. Kalata. Sensor fusion for mobile robot navigation. *Proceedings of the IEEE*, 85(1):108–119, 1997.

[88] Dean M Karantonis, Michael R Narayanan, Merryn Mathie, Nigel H Lovell, and Branko G Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE transactions on information technology in biomedicine a publication of the IEEE Engineering in Medicine and Biology Society*, 10(1):156–167, 2006.

[89] Ke Ke, Tao Zhao, and Ou Li. Bhattacharyya distance for blind image steganalysis. *Multimedia Information Networking and Security, International Conference on*, 0:658–661, 2010.

[90] P. Kelly, N. E. O'Connor, and A.F. Smeaton. Robust pedestrian detection and tracking in crowded scenes. *Image and Vision Computing Journal*, 27(10):1445 – 1458, 2009.

[91] Nicky Kern, Bernt Schiele, and Albrecht Schmidt. Multi-sensor activity context detection for wearable computing. In *IEEE Sensors Journal Special Issue on Cognitive Sensor Networks*, pages 220–232, 2003.

[92] J. Bouguet K.M. Cheung, T. Kanade and M. Holler. A real time system for robust 3d voxel reconstruction of human motions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 714–720, June 2000.

[93] Bernhard Krach and Patrick Robertson. Integration of foot-mounted inertial sensors into a bayesian location estimation framework. *2008 5th Workshop on Positioning Navigation and Communication*, 2008(2):55–61.

[94] Narayanan C. Krishnan, Dirk Colbry, Colin Juillard, and Sethuraman Panchanathan. Real Time Human Activity Recognition Using Tri-Axial Accelerometers. In *Sensors Signals and Information Processing Workshop (SENSIP)*, 2008.

[95] Narayanan C Krishnan, Colin Juillard, Dirk Colbry, and Sethuraman Panchanathan. Recognition of hand movements using wearable accelerometers. *Journal of Ambient Intelligence and Smart Environments*, 1(2):143155, 2009.

[96] John Krumm, Steve Harris, Brian Meyers, Barry Brumitt, Michael Hale, and Steve Shafer. Multi-camera multi-person tracking for easyliving. In *Third IEEE International Workshop on Visual Surveillance*, pages 3–10, 2000.

[97] Jogile Kuklyte, Philip Kelly, Ciaran Ó Conaire, Noel E. O'connor, and Li-qun Xu. Anti-social behavior detection in audio-visual surveillance systems. In *PRAI-HBA - Workshop on Pattern Recognition and Artificial Intelligence for Human Behaviour Analysis*, 2009.

[98] Rakesh Kumar, Naveen K. Chilamkurti, and Ben Soh. A comparative study of different sensors for smart car park management. In *IPC '07: Proceedings of the The 2007 International Conference on Intelligent Pervasive Computing*, pages 499–502, Washington, DC, USA, 2007. IEEE Computer Society.

[99] Kristof Van Laerhoven and Ozan Cakmakci. What shall we teach our pants? In *Proceedings of the 4th IEEE International Symposium on Wearable Computers*, ISWC '00, pages 77–, Washington, DC, USA, 2000. IEEE Computer Society.

[100] C. Laugier, Th. Fraichard, Ph. Garnier, I. E. Paromtchik, and A. Scheuer. Sensor-based control architecture for a car-like vehicle. *Auton. Robots*, 6(2):165–185, 1999.

[101] Insup Lee, George J. Pappas, Rance Cleaveland, John Hatcliff, Bruce H. Krogh, Peter Lee, Harvey Rubin, and Lui Sha. High-confidence medical device software and systems. *IEE Computers*, 39:33–38, 2006.

[102] Jonathan Lester, Tanzeem Choudhury, and Gaetano Borriello. A practical approach to recognizing physical activities. In *Proceedings*

of the 4th international conference on Pervasive Computing, PERVASIVE'06, pages 1–16, Berlin, Heidelberg, 2006. Springer-Verlag.

[103] Zhenjiang Li, Kunfeng Wang, Li Li, and Fei-Yue Wang. A review on vision-based pedestrian detection for intelligent vehicles. In *Vehicular Electronics and Safety 2006 ICVES 2006 IEEE International Conference*, pages 57–62. IEEE, 2006.

[104] K J Liszka, M A Mackin, M J Lichter, D W York, D Pillai, and D S Rosenbaum. Keeping a beat on the heart. In *IEEE Pervasive Computing*, volume 3, pages 42–49, 2004.

[105] Jingen Liu, Yang Yang, Imran Saleemi, and Mubarak Shah. Learning semantic features for action recognition via diffusion maps. *Computer Vision and Image Understanding*, 116(3):361–377, 2012.

[106] Clemens Lombriser, N B Bharatula, Daniel Roggen, and Gerhard Troster. On-body activity recognition in a dynamic sensor network. In *Proceedings of the ICST 2nd international conference on Body area networks*, pages 1–6. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.

[107] Xi Long, Bin Yin, and Ronald M Aarts. Single-accelerometer-based daily physical activity classification. *Conference Proceedings of the International Conference of IEEE Engineering in Medicine and Biology Society*, 2009:6107–6110.

[108] David G Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 60, pages 91–110. Springer, 2004.

[109] Hendrik Johannes Luinge. *Inertial sensing of human movement*. PhD thesis, Enschede, October 2002.

[110] Paul Lukowicz, Andreas Timm-Giel, Michael Lawo, and Otthein Herzog. Wearit@work: Toward real-world industrial wearable computing. *IEEE Pervasive Computing*, 6(4):8–13, 2007.

[111] R C Luo and M G Kay. Multisensor integration and fusion in intelligent systems. *Ieee Transactions On Systems Man And Cybernetics*, 19(5):901–931, 1989.

[112] J Maitland, S Sherwood, L Barkhuus, I Anderson, M Hall, B Brown, M Chalmers, and H Muller. Increasing the awareness of daily activity levels with pervasive computing. *2006 Pervasive Health Conference and Workshops*, 36(4):1–9, 2006.

[113] R Manduchi. Bayesian fusion of color and texture segmentations. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2(c):956–962, 1999.

[114] G Miller and C Gabbard. Effects of visual aids on acquisition of selected tennis skills. *Percept Mot Skills*, 67(2):603–6, 1988.

[115] David Minnen, Thad Starner, Irfan Essa, and Charles Isbell. Discovering characteristic actions from on-body sensor data. In *IN PROC. OF IEEE INTERNATIONAL SYMPOSIUM ON WEARABLE COMPUTING*, pages 11–18, 2006.

[116] David Minnen, Tracy Westeyn, Daniel Ashbrook, Peter Presti, and Thad Starner. *Recognizing soldier activities in the field*, page 236241. Springer, 2007.

[117] Thomas B. Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Comput. Vis. Image Underst.*, 81(3):231–268, 2001.

[118] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst.*, 104(2):90–126, 2006.

[119] H.J. Montoye, R. Washburn, S. Servais, A. Ertl, J.G. Webster, and F.J. Nagle. Estimation of energy expenditure by a portable accelerometer. *Med Sci Sports Exerc*, 15(5):403–7, 1983.

[120] H. Naphide and T. Huang. A probabilistic framework for semantic video indexing, filtering, and retrieval. *Multimedia, IEEE Transactions on*, 3(1):141–151, 2001.

[121] Jakob Nielsen. How to conduct a heuristic evaluation. *useitcom*, pages 1–11, 2002.

[122] Ciarán Ó Conaire, Damien Connaghan, Philip Kelly, Noel E. O'Connor, Mark Gaffney, and John Buckley. Combining inertial and visual sensing for human action recognition in tennis. In *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams*, pages 51–56, 2010.

[123] P O'Donoghue and D Liddle. A notational analysis of time factors of elite mens and ladies' singles tennis on clay and grass surfaces. *In Science and Racket Sports II (ed. A. Lees, I. Maynard, M. Hughes and T. Reilly)*, pages 241–246, 1998.

[124] Antonios Oikonomopoulos, Ioannis Patras, and Maja Pantic. Spatiotemporal salient points for visual recognition of human actions. *IEEE transactions on systems man and cybernetics Part B Cybernetics a publication of the IEEE Systems Man and Cybernetics Society*, 36(3):710–719, 2006.

[125] B B Orten, Medeni Soysal, and A Aydin Alatan. Person identification in surveillance video by combining mpeg-7 experts. *Proceedings of the IEEE 13th Signal Processing and Communications Applications Conference 2005*, (1):352–355, 2005.

[126] I. Pappas, T. Keller, S. Mangold, M.R. Popovic, V. Dietz, and M. Morari. A reliable gyroscope-based gait-phase detection sensor embedded in a shoe insole. *IEEE Sensors*, 4(2):268–274, April 2004.

[127] Milan Petkovic, Willem Jonker, and Z. Zivkovic. Recognizing strokes in tennis videos using hidden markov models. In *VIIP*, pages 512–516, 2001.

[128] S Pirttikangas, K Fujinami, and T Nakajima. Feature selection and activity recognition from wearable sensors. *Ubiquitous Computing Systems*, 4239:516–527, 2006.

[129] S V Porter, M Mirmehdi, and B T Thomas. Video cut detection using frequency domain correlation. *Proceedings 15th International Conference on Pattern Recognition ICPR2000*, 3, 2000.

[130] A. Mitiche Q. Cai and J. K. Aggarwal. Tracking human motion in an indoor environment. In *International Conference on Image Processing*, volume 1, page 215, 1995.

[131] Cliff Randell and Henk Muller. Context awareness by analyzing accelerometer data. In *Proceedings of the 4th IEEE International Symposium on Wearable Computers*, ISWC '00, pages 175–, Washington, DC, USA, 2000. IEEE Computer Society.

[132] M Reid, M Crespo, B Lay, and J Berry. Skill acquisition in tennis: Research and current practice. *Journal of Science and Medicine in Sport*, 10(1):1–10, 2007.

[133] Bradley J. Rhodes. The wearable remembrance agent: A system for augmented memory. In *Personal Technologies*, pages 123–128, 1997.

[134] Mikel Rodriguez, Ivan Laptev, Josef Sivic, and Jean-Yves Audibert. Density-aware person detection and tracking in crowds. *2011 International Conference on Computer Vision*, pages 2423–2430, 2011.

[135] Daniel Roetenberg, Henk Luinge, and Peter Veltink. Inertial and magnetic sensing of human movement near ferromagnetic materials. In *Mixed and Augmented Reality, IEEE / ACM International Symposium on*, volume 0, page 268, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

[136] Bodo Rosenhahn and Dimitris N. Metaxas Reinhard Klette. *Human motion: understanding, modeling, capture and animation.* Computational Imaging. Springer, 2008.

[137] Matthias Sala, Kurt Partridge, Linda Jacobson, and James Begole. An exploration into activity-informed physical advertising using pest. *Pervasive Computing*, 4480:73–90, 2007.

[138] Bernt Schiele, Nuria Oliver, Tony Jebara, and Alex Pentland. An interactive computer vision system dypers: Dynamic personal enhanced reality system. In *International Conference on Computer Vision Systems*, pages 51–65. Springer, 1999.

[139] Andrew Senior. An introduction to automatic video surveillance. *Health Policy*, 92(2-3):1–9, 2009.

[140] Hitesh Shah, Prakash Chokalingam, Balamanohar Paluri, S. Nalin Pradeep, and Raman Balasubramanian. Automated stroke classification in tennis. In *ICIAR*, pages 1128–1137, 2007.

[141] Cees G. M. Snoek. Early versus late fusion in semantic video analysis. In *In ACM Multimedia*, pages 399–402, 2005.

[142] J Soderkvist. Micromachined gyroscopes. In *Proceedings of Sensors and Actuators*, pages 65–71, 1994.

[143] Daniel Spelmezan and Jan Borchers. Real-time snowboard training system. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 3327–3332, New York, NY, USA, 2008. ACM.

[144] S.N. Srihari. Lecture slides for machine learning and probabilistic graphical models. *Department of Computer Science and Engineering, University at Buffalo - http://www.cedar.buffalo.edu/ srihari/CSE574/ (accessed 10/10/12)*, 2009.

[145] Vince Stanford. Wearable computing goes live in industry. *IEEE Pervasive Computing*, 1:14–19, 2002.

[146] Thad Starner, Bradley Rhodes, Joshua Weaver, and Alex Pentland. Everyday-use wearable computers. *IBM Systems Journal*, 35:618–629, 1996.

[147] Thad Starner, Joshua Weaver, and Alex Pentland. A wearable computer based american sign language recognizer. In *Wearable Computers, 1997. Digest of Papers., First International Symposium on*, pages 130–137, 1997.

[148] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 246–252, 1999.

[149] Thomas Stiefmeier, Georg Ogris, Holger Junker, Paul Lukowicz, and Gerhard Tröster. Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario. In *ISWC*, pages 97–104, 2006.

[150] Peter A. Stubberud and Allen R. Stubberud. A signal processing technique for improving the accuracy of mems inertial sensors. In *ICSENG '08: Proceedings of the 2008 19th International Conference on Systems Engineering*, pages 13–18, Washington, DC, USA, 2008. IEEE Computer Society.

[151] Michael E. Tipping and Cambridge Cb Nh. Sparse kernel principal component analysis. In *Proceedings of Advances in Neural Information Processing Systems*, 2001.

[152] Picture Thresholding Using and Iterative Selection Method. Picture thresholding using an iterative selection method. *Ieee Transactions On Systems Man And Cybernetics*, 8(8):630–632, 1978.

[153] Kristof Van Laerhoven and Ozan Cakmakci. What shall we teach our pants? *Digest of Papers Fourth International Symposium on Wearable Computers*, pages:77–83, 2000.

[154] Kristof Van Laerhoven, Nicky Kern, Hans-Werner Gellersen, and Bernt Schiele. Towards a wearable inertial sensor network. *Eurowearable 2003 IEE*, page 125130, 2003.

[155] Ashok Veeraraghavan, Amit K Roy-Chowdhury, and Rama Chellappa. Matching shape sequences in video with applications in human movement analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1896–1909, 2005.

[156] Roberto Vezzani, Davide Baltieri, and Rita Cucchiara. Hmm based action recognition with projection histogram features. *ICPR contest on Semantic Description of Human Activities SDHA in Proceedings of the ICPR contests*, pages 286–293, 2010.

[157] Hongzhi Wang and Ying Dong. An improved image segmentation algorithm based on otsu method. *Proceedings of SPIE*, 6625(2008):66250I–66250I–8, 2007.

[158] Shiaokai Wang, William Pentney, Ana-Maria Popescu, Tanzeem Choudhury, and Matthai Philipose. Common sense based joint training of human activity recognizers. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 2237–2242, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[159] J A Ward, P Lukowicz, and G Troster. Gesture spotting using wrist worn microphone and 3-axis accelerometer. *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence innovative contextaware services usages and technologies sOcEUSAI 05*, 121(october):99, 2005.

[160] Jamie A Ward, Paul Lukowicz, Gerhard Troster, and Thad E Starner. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1553–1567, 2006.

[161] Mark Weiser. Human-computer interaction. chapter The computer for the 21st century, pages 933–940. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995.

[162] Thijs Westerveld, Arjen P. De Vries, Alex van Ballegooij, Franciska de Jong, and Djoerd Hiemstra. A probabilistic multimedia retrieval

model and its evaluation. *EURASIP Journal on Applied Signal Processing*, 2003:186–198, 2003.

[163] F. E. White. Data Fusion Lexicon, Joint Directors of Laboratories. Technical report, Naval Ocean Systems Center, 1987.

[164] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[165] T.C. Wong, J.G. Webster, H.J. Montoye, and R. Washburn. *Portable accelerometer device for measuring human energy expenditure*. Wisconsin. University - Madison. Department of Electrical and Computer Engineering. [Papers]. University of Wisconsin, Engineering Experiment Station, 1980.

[166] Jiahui Wu, Gang Pan, Daqing Zhang, Guande Qi, and Shijian Li. Gesture recognition with a 3-d accelerometer. *Ubiquitous Intelligence and Computing*, 5585:25–38, 2009.

[167] Danny Wyatt, Matthai Philipose, and Tanzeem Choudhury. Unsupervised activity recognition using automatically mined common sense. In *Proceedings of the 20th national conference on Artificial intelligence - Volume 1*, pages 21–27. AAAI Press, 2005.

[168] J Yang, J Wang, and Y Chen. Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. *Pattern Recognition Letters*, 29(16):2213–2220, 2008.

[169] Jhun-Ying Yang, Jeen-Shing Wang, and Yen-Ping Chen. Using acceleration measurements for activity recognition: An effective learning

algorithm for constructing neural classifiers. *Pattern Recogn. Lett.*, 29(16):2213–2220, December 2008.

[170] Ki-Won Yeom and Ji-Hyung Park. *An Approach of Information Extraction from Web Documents for Automatic Ontology Generation.* Springer, 2005.

[171] M. M. Yeung and Bede Liu. Efficient matching and clustering of video shots. In *Proceedings of the 1995 International Conference on Image Processing (Vol. 1)-Volume 1 - Volume 1*, ICIP '95, pages 338–, Washington, DC, USA, 1995. IEEE Computer Society.

[172] Suya You and Ulrich Neumann. Fusion of vision and gyro tracking for robust augmented reality registration. *Image Rochester NY*, pages 71–78, 2001.

[173] Xinguo Yu, Changsheng Xu, Hon Wai Leong, Qi Tian, Qing Tang, and Kong Wah Wan. Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video. In *Proceedings of the eleventh ACM international conference on Multimedia*, MULTIMEDIA '03, pages 11–20, New York, NY, USA, 2003. ACM.

[174] Ramin Zabih, Justin Miller, and Kevin Mai. A feature-based algorithm for detecting and classifying production effects. *Multimedia Systems*, 7(2):119–128, 1999.

[175] Haiyan Zhang and Björn Hartmann. Building upon everyday play. In *CHI '07 extended abstracts on Human factors in computing systems*, CHI EA '07, pages 2019–2024, New York, NY, USA, 2007. ACM.

[176] Chun Zhu and Weihua Sheng. Human daily activity recognition in robot-assisted living using multi-sensor fusion. *Proceedings of the IEEE*

International Conference on Robotics and Automation (2009), pages 2154–2159, 2009.

[177] Chun Zhu and Weihua Sheng. Realtime human daily activity recognition through fusion of motion and location data. In *The 2010 IEEE International Conference on Information and Automation*, pages 846–851. IEEE, 2010.

[178] Guangyu Zhu, Qingming Huang, Changsheng Xu, Yong Rui, Shuqiang Jiang, Wen Gao, and Hongxun Yao. Trajectory based event tactics analysis in broadcast sports video. *Proceedings of the 15th international conference on Multimedia MULTIMEDIA 07*, page 58, 2007.

[179] Guangyu Zhu, Changsheng Xu, Qingming Huang, Wen Gao, and Liyuan Xing. Player action recognition in broadcast tennis video with applications to semantic analysis of sports game. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 431–440, New York, NY, USA, 2006. ACM.

[180] Z. Zivkovic, F. Heijden van der, M. Petkovic, and W. Jonker. Image segmentation and feature extraction for recognizing strokes in tennis game videos. In R.L. Langendijk, J.W.J. Heijnsdijk, A.D. Pimentel, and M.H.F. Wilkinson, editors, *Seventh annual conference of the Advanced School for Computing and Imaging, ASCI 2001*, pages 262–266, Delft, the Netherlands, 2001. Advanced School for Computing and Imaging (ASCI).