TAMPERE UNIVERSITY OF TECHNOLOGY

MIERAIDIHAIMU MIERAISAN

BASKETBALL GAME ANALYZING BASED ON COMPUTER VISION

Master of Science Thesis

Examiner: Professor IREK DEFÉE
Examiner and topic approved by the Computing and Electrical Engineering Faculty Council meeting on
6th October 2013

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Information Technology
MIERAIDIHAIMU MIERAISAN : BasketBall Game Analyzing Based on Computer Vision
Master of Science Thesis, 50 pages
November 2013
Major: Multimedia
Examiner: Professor IREK DEFÉE
Keywords: Computer vision, sports intelligent, play-field detection, player tracking

As tremendous improvement in computer vision technology, various industries start to apply computer vision to analyze huge multimedia content. Sports as one of the biggest resource invested industries also step up to utilize this technology to enhance their sports intelligent products.

The thesis is following this development to provide prototype implementations of computer vision algorithms in sports industry. Main objective is to develop initial algorithms to solve play-field detection and player tracking in basketball game video. Play-field detection is an important task in sports video content analysis, as it provides the foundation for further operations such as object detection, object tracking or semantic event highlight and summarization. On the other hand, player tracking highlight player movements in critical events in basketball game. It is also a challenging task to develop effective and efficient player tracking in basketball video, due to factors such as pose variation, illumination change, occlusion, and motion blur.

This thesis proposed reliable and efficient prototype algorithms to address play-field detection and single player tracking. SURF algorithm is utilized and modified to offer precise location of play-field and overlay trajectory data to improve viewer's experience on sports product. And compressive tracking algorithm implemented for the aim of capture and track single player in important events to reveal player's secret tactics. Prototype implementation to meet the current needs in basketball video content analyzing field.

PREFACE

The practical part of this work conducted at SstatzZ Oy in Helsinki, Finland. The chance which led me to do this thesis stared in SstatzZ's project. The project aim was to find efficient algorithms fro detecting play-field, tracking the single basketball player in game video. A very special thanks to Harri Hohteri who gave me opportunity to work with fantastic team and guided me to apply my knowledge to the real industry area. Also I wish to thank Jirka Poropudas, Tomas Gatial, Vesa Matti Makinen and Teemu Spiridovitsh who are part of SstatzZ team for helping me and giving me advise during the work.

Thesis has been carried out at Department of Signal Processing, Tampere University of Technology, Finland. Professor Irek Defee providing premises where I could continue working with the topic after I left SstatzZ Oy. I wish to express my deepest gratitude to my thesis supervisor Professor Irek Defee for his invaluable guidance, advise and patience through my thesis work. In addition to that, I highly appreciate Department of Signal Processing in Tampere University of Technology for offering for me their cherish academic knowledge.

I wish to thank my family for their invaluable love. My parents, my brothers and sister's constructive guide me in life. Beside, I am appreciative to my friends Alaadin Lulu, Amir Shokouhi and Armin Iraqi who always support me and provide me with positive energy.

Tampere 11.11.2013
Mieraidihaimu Mieraisan

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Over the past decades, there is an explosive growth in the amount of available multimedia information in our daily lives. This trend necessitates the development of content-based video analysis, indexing and retrieval technology. In recent years, extensive research efforts have been devoted to sports video content analysis and applications due to their wide viewer-ship and high commercial potentials. Technologies and prototypes have been developed for automatic or semiautomatic sports video content analysis [1].

As the field of computer vision become more and more mature in academic research area, the applications based on computer vision are growing rapidly in various industries. Computer vision is a field that includes methods for acquiring, processing, analyzing, and understanding multimedia content, in general, high dimensional data from the real world in order to produce numerical or symbolic information, in the forms of decisions [2]. In other word, computer vision field is developed based on human vision mechanism. Human eyes obtain natural image and brain responsible for analyzing the image content to react based on decision made by human brain. Even though computer vision technology, it still cannot reach the accuracy and efficiency of human eyes and brain analyzing, gives us easy way to handle huge number of problem sets in academic, industry and military area. For instance, the field of intelligent sports can utilize computer vision technology to build rich content multimedia production.

There are increasing demands in the sport community to have sophisticated sports analyzing system which will help coaches and sport scientists to evaluate the performance of the team as well as the individual players during training or official games. Sports analyzing is including sports knowledge, mathematical modeling, data analyzing and vision based technologies to achieve semantic analysis for the game. Basketball game is one of sports field which has strong demand for integrating information technology. Basketball is a team sport where the teams score points by shooting the ball to the basket of the opponent(FIBA 2010).

Since the introduction of film to competitive sports, sports coaches have used video footage of recorded games in order to analyze efficacy of plays run by their team, or to strategist against plays run by opposing teams. To accomplish this, coaching staff often manually track players and annotate player positions throughout

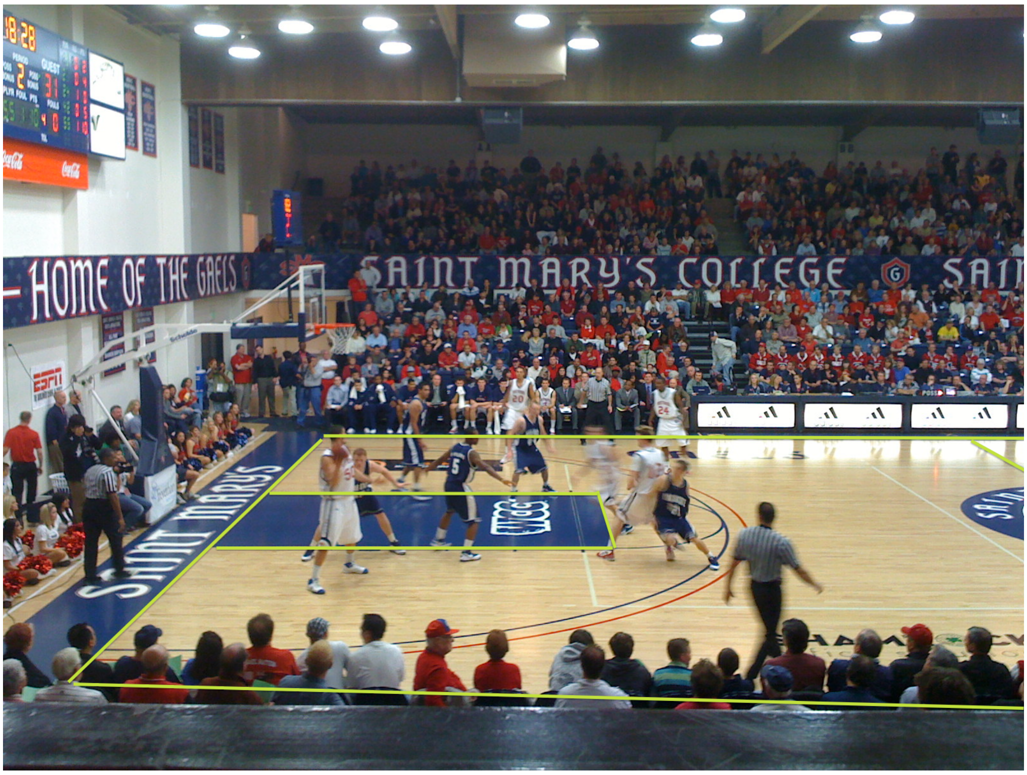


Figure 1.1: Play-field detection for basketball game (source of Saint Mary's College men's basketball team, during game against Utah State at McKeon Pavilion)

videos. With the ever increasing quantity of available recordings, this process can become highly time consuming. In addition, broadcast sports games are often shot at an angle above the court which makes it difficult to determine the true position of a player in any given play. Although there have been a considerable amount of research for play annotation in sports such as tennis and soccer, the additional challenges in basketball are related to the varied patterns on the playing field, movement of the camera during game has motivated us to create a system for analyzing plays in basketball match video [3].

In computer vision based sports analyzing, play-field (court) detection and player tracking are the key problem that need to be solved. Persons, object, noise in the audience may be excluded from further processing based on the fact that they are not on the court. That is why we need to detect court boundaries. In addition to that, we want to overlay exclusive information on play-field and this needs finding play-field location coordinates. Play-field detection to decide court area of team sports in video clips, has sophisticated and essential role for further video analyzing. The basic target and result of the play-field detection revealed at Figure 1.1, is to find the position and essential edges of the court to reconstruct court area. There is plenty of research going on for finding better and efficient solution to this problem.

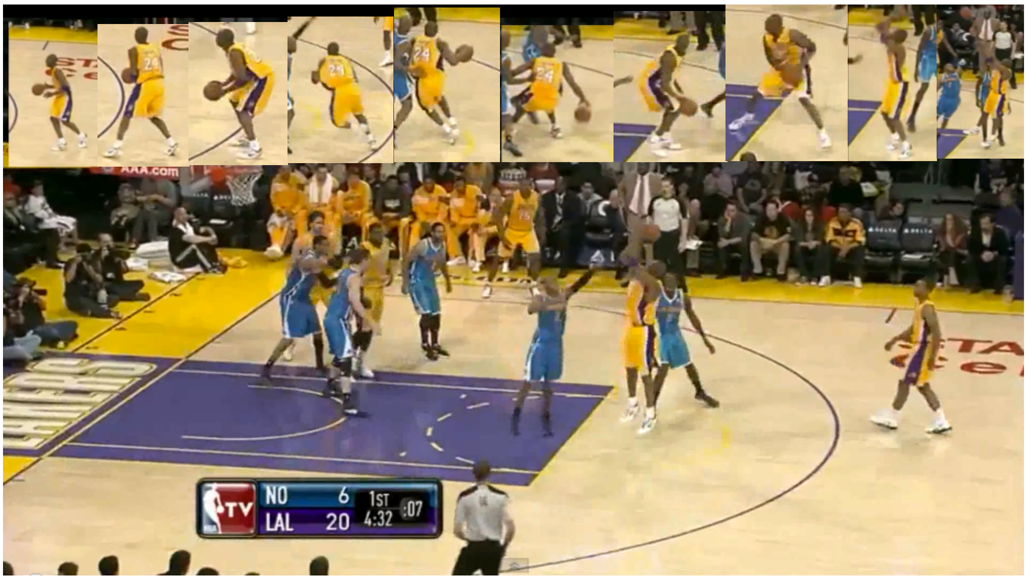


Figure 1.2: Player tracking for basketball game (source of NBA game 31.3.2013 between Lakers versus New Orleans Hornets, Kobe Bryant)

In most of the research literatures, researchers usually use histogram to model the play-field [4], [5], [6]. In A.Ekin et al. [4], the dominant color detection algorithm is first described in detail. They exploit two color spaces, one is used as control space, and the other is primary space. The dominant color is represented by the mean value of the main region in the histogram. The algorithm calculates the cylindrical metric to determine whether a pixel belongs to the play-field. In [5], Xie also uses the dominant color in HSI space to analyze soccer video . Besides, in [1] Gong et al. adopt Laplacian operator to extract edges and find lines in the court by subtracting those edges with non-white pixels. Through analyzing relationship among lines, circle and semicircles with domain knowledge, the zone of play-field is recognized. In addition to those method, in [7] Shuqiang et al. exploits Gaussian mixture model (GMM). In [7], Shuqiang et al. use adopted GMM modeling which is more sufficiently general to model complex distribution density than histogram for accurately modeling the play-field, and then incremental EM (IEM) algorithm [8] is employed to update the GMM for the purpose of adapt the model to play-field evolution, at the end perform one-pass training to determine the court. Although these references are the current state-of-art, there still are not capable of solving some play-field detection problems since the shape, lines, colors and camera shooting angle of the court are varying. For the sake of simplicity and accuracy in our prototype algorithm, we pursue adapted court detection using speed up robust fast detection algorithm assisted with dominant color histogram.

On the other hand, sports player tracking also attracts a lot of scholars and vision developers. During sports games, it is nearly impossible to pay attention to the movement of all the players on the field during any given play. Due to some of critical players movement, match situation will change dramatically, thus tracking players movement during the specific time will reveal the trick how player made that shot in specific basketball match. Figure 1.2 illustrates one of these critical times, at which Kobe Bryant (one of the best superstar in NBA) made the critical shot that helps Lakers starting to lead the match. For developing efficient and robust algorithms toward sports player tracking, more and more scholars, researchers and developers made their best to improve state-of-art tracking algorithms. Our prototype algorithm aims to solve single player tracking in basketball game, focus and zoom out player for visualization player movement. For this purpose we choose and implement efficient and robust tracking method compressive tracking algorithm.

The goal of the thesis is to implement several algorithms by analyzing the game videos for basketball coaching system to improve the viewer's experience in sports intelligent product. Despite the challenges in play-field detection and single player tracking, our methodologies bring a new perspective of computer vision application in sports intelligent product. A first attempt is to demonstrate the ability to precise detect play-field and track the basketball player using Finnish National Korisliiga broadcast videos. The system is a first prototype attempt to extend the intelligent basketball system by appending it with computer vision algorithm.

The thesis is organized as follows. In Chapter 2, we review some related work in intelligent sport video analysis. Chapter 3 overviews the structure of our proposed basketball play-field detection and single player tracking algorithms; Chapter 4 describes the implementation details of our pursued system. In Chapter 5, we report our experiment results and offer discussion toward our results. We conclude this thesis in Chapter 6 and also provide some suggestions for future extensions in sports intelligent based on computer vision.

2. RELATED WORK

This chapter develops the foundations for the key concepts used in this thesis. Sports video analysis, especially team sport as basketball, has received much of the attention due to its popularity as well as potential commercial value. A broad range of applications such as: content retrieval and indexing, semantic event highlight and summarization or objects recognition and tracking make building sports video analysis system a hot research area. In those video analysis systems, play-field (or court) detection and player tracking are usually very key steps to provide the ground for further detection operations. When examining relevant researches, we must consider two types of references. The first one covers play-field detection which is conducting the method of object detection, and then second one describes single player tracking that is using state of art in object tracking algorithms.

2.1 Play-field Detection

Play-field detection is an important task in sports video content analysis, as it provides the ground for further operations such as object detection, object tracking or semantic event highlight and summarization. By identifying the whereabouts of the play-field region, noise from unimportant area could be eliminated and detection job becomes much simpler.

While play-field detection is an essential part of sports video analysis, there are still numerous of obstacles that make a robust and accurate detection system become challenging. The variance of information on basketball play-field detection such as court patterns, which are color, texture, court lines, model or camera illumination, make play-field detection difficult. Beside, various modeling also introduces quite huge complexity to the process. A lot of recent studies are trying to find easy, cost-effective, low complexity and robust algorithm to solve the play-field detection difficulties. One big part of ongoing researches is targeted on single camera captured video clip [4; 9; 10; 12; 13; 14], by conducting edge detection, dominant color or Gaussian mixture modeling individually or in a combined way to address the detection problem. In some references a large number of synchronized cameras were placed on predefined locations [11]. In such cases, the court detection represents a trivial task. Since our landing target is single camera captured basketball game video, we will introduce current state-of-art play-field detection algorithm.

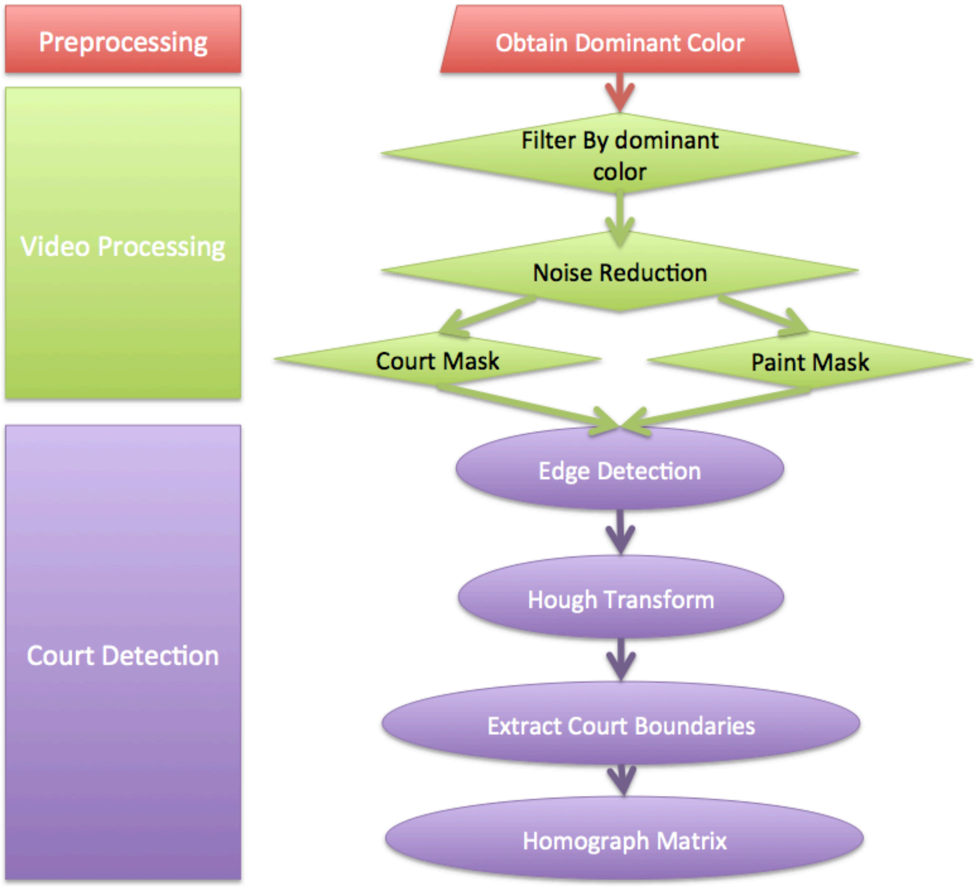


Figure 2.1: The block diagram of court detection, mainly include three parts: preprocessing, video processing, court detection

The straightforward method introduced in Micheal et al. [9] court detection implementation is based on edge detection and playing field mask, Figure 2.1 shows the block digram of court detection algorithm. The shinning point of this algorithm is camera calibration which can map the court in frame to 2D court model. However, this algorithms needs preprocessing stage which needs a lot of manual work to acquire data to help to find and locate court region.

In related research papers, play field detection is mainly based on the dominant colors [13; 4; 14]. Dominant color approach initially took advantage of the court having unique color distribution than other objects, and noise in sports match game. For instance, in football match the dominant color of play-field will be green and easier to distinguish from foreground objects. However, in some cases, the play court does not represent a dominant color. For example, in the 2011/2012 NBS finals, all the Oklahoma fans were wearing blue shirts just like the color of the home team Jerseys together with the area below the basket. In a situation like this, the court does not represent the dominant color and the whole algorithm is not applicable.

Wu et al. [13] tried to overcome this problem by taking into account the fact

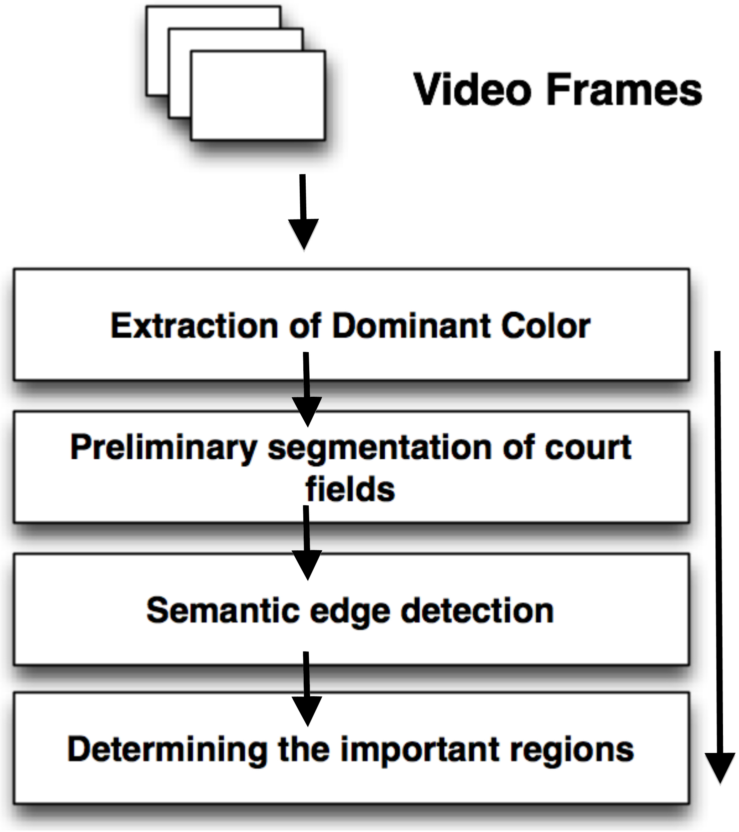

Video Frames
Extraction of Dominant Color
Preliminary segmentation of court fields
Semantic edge detection
Determining the important regions

Figure 2.2: The block diagram of play field detection part in semantic aware seam carving algorithm

that the court is always on the lower half of the image. That is why they considered only the dominant color on the lower half and determined the court by calculating histogram of hue (H) component of the HSV color system. In this reference, they proposed a semantic aware image resizing approach. They segment the play fields as important regions and detect the boundary of court fields as the semantic edges. Figure 2.2 is the block diagram of their play-field detection algorithm part. They aim is to segment the regions of a court field as important regions and extract boundary of the court field as semantic edges. The advantages of this algorithm are automatically recognized court field using dominant colors and semantic edges detected in the candidate court field followed by semantic important regions are segmented accurately in most of the cases. However, the algorithm have two shortcomings complicating to achieve what we want. Firstly, due to relying too much on dominant color it can cause some fail cases in our database. Secondly, the algorithm can not point out and map play field from frame to 2D court model.

Chang et al. [14] also used the principle of the dominant color to determine the court, and create a mask that will later be used in the court detection. They take an assumption that the lines defining the play court are white, which in some halls

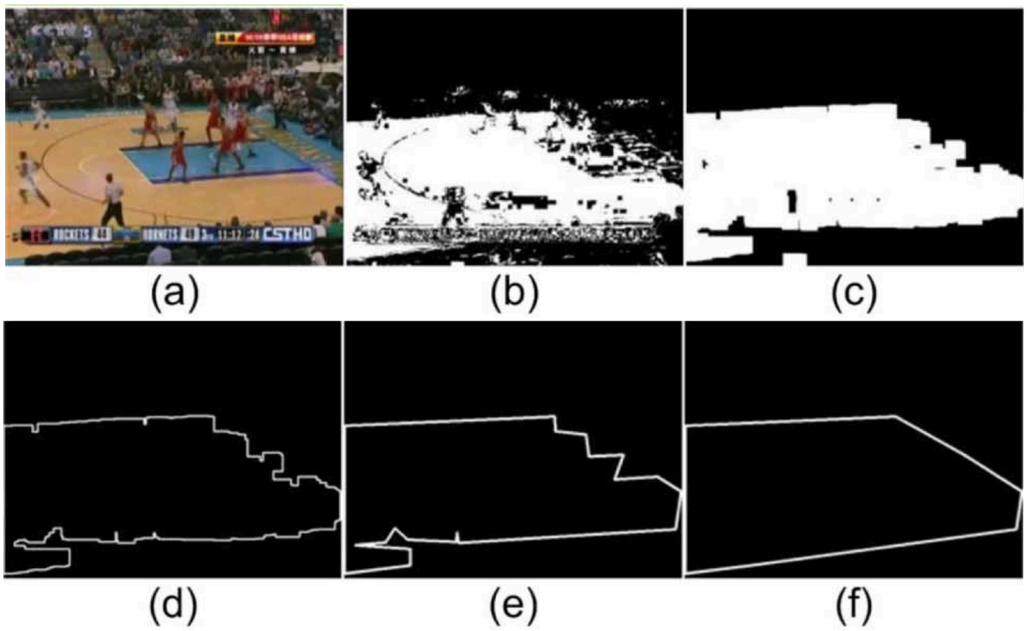


Figure 2.3: Illustration of the play-field contour extraction process (a) Input frame. (b) After ST-GMM segmentor. (c) After temporal propagation. (d) Extracted maximal contour (e) After edge compact. (f) Convex hull as final play-field contour [12]

and arenas is not the case, and so this approach is not robust enough. Ahmet et al. [4] proposed generic algorithm that automatically extracts dominant color statistics and adapts them to the variations in imaging conditions that are typically for sports video.

Junliang et al. [12] proposed multiple object tracking system architecture for sports video, which mainly consists of progressive observation modeling and dual mode two-way Bayesian inference. At classifying play-field part in progressive observation model, they design a multi-cue learning algorithm to obtain a robust play-field segmentor which consists of two stages, the offline stage and the online stage. Within these two stages, multiple useful features, including color, motion and shape are explored and learned to form the final play-field segmentor. In the offline stage, a spatial temporal Gaussian mixture modeling (ST-GMM) is learned to classify the play-field. In the online stage, a motion model is employed to capture the correlation between successive frames, followed by a shape model is adopted to extract the final play-field contour. Figure 2.3 illustrate the process for obtaining play-field using this algorithm.

Viet Anh Ngo et al. [10] proposed novel approach to detect the play-field region using the concept of Area-of-Coverage (AoC). They improved the accuracy of play-field detection by combining color and homograph information. Figure 2.4 state their system overview. First, the system uses Gaussian mixture model (GMM) to obtain rough segmentation of the play-field are. Then, based on this initial segmentation,

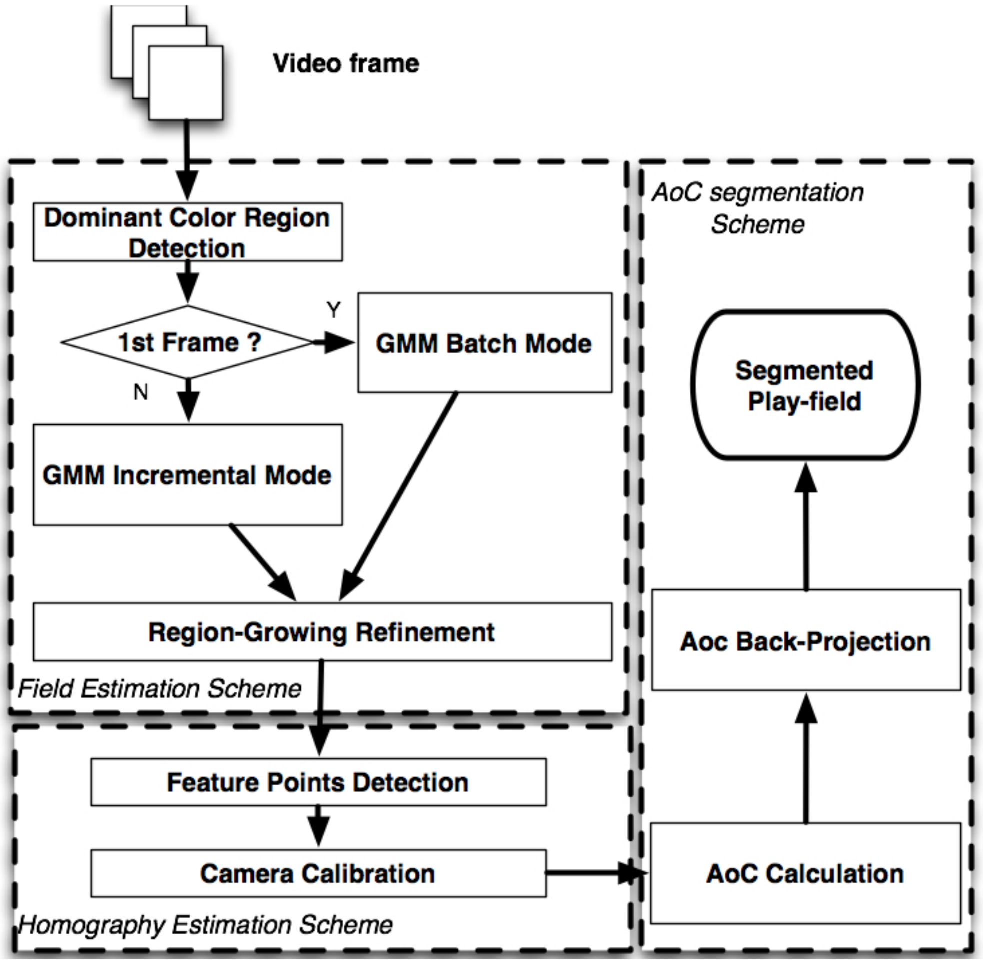


Figure 2.4: System overview of accurate play-field detection using Are-of-Coverage algorithm [10]

feature points of the field are detected for calibration using homograph and then the area of coverage of the camera view is calculated. Finally, this AoC is back-projected to the image frame to provide a refined detection result. This algorithm performed on soccer game did achieve great results both the region of the play-field and court contour. However, when it is dealing with basketball game the result are not really sufficient since basketball video contains huge amount of noise, court line color is not the only one color and so on.

Our goal of play-field detection is to find the court region and homography matrix to precisely locate and overlay exclusive trajectory information content on top of court in video frame. The key point is not only to segment but also obtaining exact position of the court. Thus in next chapter, we will present extended SURF features to detect and locate play-field. on chapter 3 we will cover our proposed method.

2.2 Single Player Tracking

Object tracking has been a hot topic in the area of computer vision. A lot of research has been undergoing ranging from sophisticated algorithms to applications. State-of-the-art object tracking is an important task within the field of computer vision. Object tracking is useful in a wide range of applications: surveillance cameras, vehicle navigation, perceptual user interface, and augmented reality, in addition object tracking will improve the experience of sports analyzing. Proliferation of high speed computers, the availability of high quality and inexpensive video cameras are accelerating the need for automated video analysis, then this opportunity generate the a great deal of interest in object tracking algorithms.

Object tracking in general is a challenging problem. Difficulties in tracking objects can arise due to abrupt object motion, changing appearance patterns of both the object and the scene, nonrigid object structures, object-to-object and object-to-scene occlusions, and camera motion. Tracking is usually performed in the context of higher-level applications that require the location or shape of the object in every frame [15]. Due to the loss of information caused by projection of the 3D context on a 2D image, noise in images frames, complex object object motion, partial or full object occlusions and complex object shapes, the tasks of object tracking are more and more challenging.

The key steps in object tracking are including detection of interesting moving objects, tracking of such objects from frame to frame, and analysis of object tracks to recognize their behavior. However to overcome the difficulties in tracking object, for each and specific application one should find proper algorithm or combination of algorithms to fulfill the demand.

The aim of an object tracker is to generate the trajectory of an object over time by locating its position in every frame of the video. Object tracker may also provide the complete region in the image that is occupied by the object at every time instant. The tasks of detecting the object and establishing correspondence between the object instances across frames can either be performed separately or jointly. In the first case, possible object regions in every frame are obtained by means of object detection algorithm, and then the tracker corresponds objects across frames. In the latter cases, the object region and correspondence is jointly estimated by iteratively updating object location and region information obtained from previous frames. In either tracking approach, the objects are represented using the shape or appearance models described in above section object detection. The model selected to represent object shape limits the type of motion or deformation it can undergo. For example, if an object is represented as a point, then only a translational model can be used. In the case where a geometric shape representation like an ellipse is used for the object,

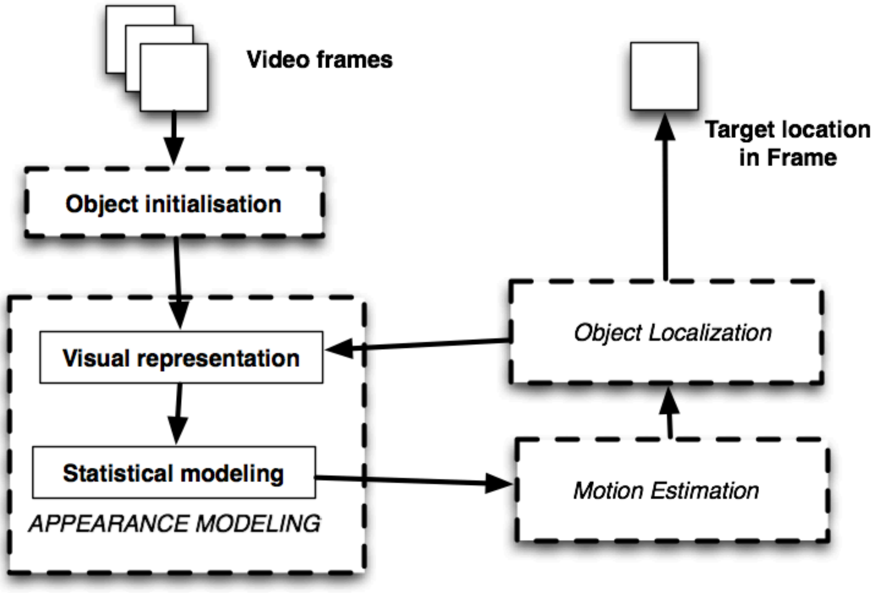


Figure 2.5: Illustration of object tracking components

parametric motion models like affine or projective transformations are appropriate. These representations can approximate the motion of rigid objects in the scene. For a nonrigid object, silhouette or contour is the most descriptive representation and both parametric and nonparametric models can be used to specify their motion.

Every tracking method requires an object detection mechanism either in every frame or when the object first appears in the video. A common approach for object detection is to use information in a single frame. However, some object detection methods make use of the temporal information computed from a sequence of frames to reduce the number of false detections. This temporal information is usually in the form of frame differencing, which highlights changing regions in consecutive frames. Given the object regions in the image, it is then the tracker's task to perform object correspondence from one frame to the next to generate the tracks.

Typically object tracking is composed of four modules: object initialization, appearance modeling, motion estimation and object localization as Figure 2.5 shows.

i. Object Initialization

This process may start manually or automatically. Manual initialization is performed by users to annotate object locations with bounding boxes or ellipses. In contrast, automatic initialization is usually achieved by object detectors such as face detector or human detector. Tracking algorithm based on correlation between succeeding frames is used to achieve its goal.

ii. Appearance Modeling

This process generally consists of two components: visual representation and statistical modeling. Visual representation focuses on how to construct robust object descriptors using different types of visual features (color, shape, texture, model etc.). Visual representation is composed of global visual representation and local based visual representation. Former one which reflects the global statistical characteristics of object appearance, can be investigated in raw pixel, optical flow, histogram, covariance, wavelet filtering and active contour representations. Local feature-based visual representation mainly utilize interest points or saliency detection to encode the object appearance information, and mainly it is conducting using local template-based, segmentation-based, SIFT-based and SURF-based algorithms to extract the visual representation data features. Statistical modeling concentrates on how to build effective mathematical models for object identification using statistical learning techniques. Statistical modeling is dynamically performed to support object detection, and it mainly consists of generative, discriminative and hybrid generative discriminative modeling classes.

It is a challenging task to develop effective and efficient appearance models for robust object tracking due to factors such as pose variation, illumination change, occlusion, and motion blur. Thus proper appearance model is considered as the heart of the tracking system. In addition to that, most of the state-of-art tracking algorithm focus directly to improve appearance model of their system.

iii.Motion Estimation

This is formulated as a dynamic state estimation problem: $x_t = f(x_{t-1}, v(t-1))$ and $z_t = h(x_t, w_t)$, where x_t is the current state, f is the state evolution function, v_{t-1} is the evolution process noise, z_t is the current observation, h denotes the measurement function, and w_t is the measurement noise. The task of motion estimation is usually completed by utilizing predictors such as linear regression techniques [16], Kalman filters [17], or particle filters [18].

Motion estimation take the advantages of correlation between succeeding frames, trying to predict the object motion path according to tracking history. It usually append to tracking system in order to enhance performance and improve the accuracy.

vi.Object Localization

This is performed by a greedy search or maximum a posterior estimation based on motion estimation. The localization procedure starts from the position of the target in the previous frame and searches in the neighborhood. Object localization provide player position so that it can be used for further visualization.

Based on above mentioned fundamental concepts, there are discriminative methods implemented recently to deal with player tracking issues in sports events. A popular method is kernel-based object tracking introduced by Comaniciu et al. in

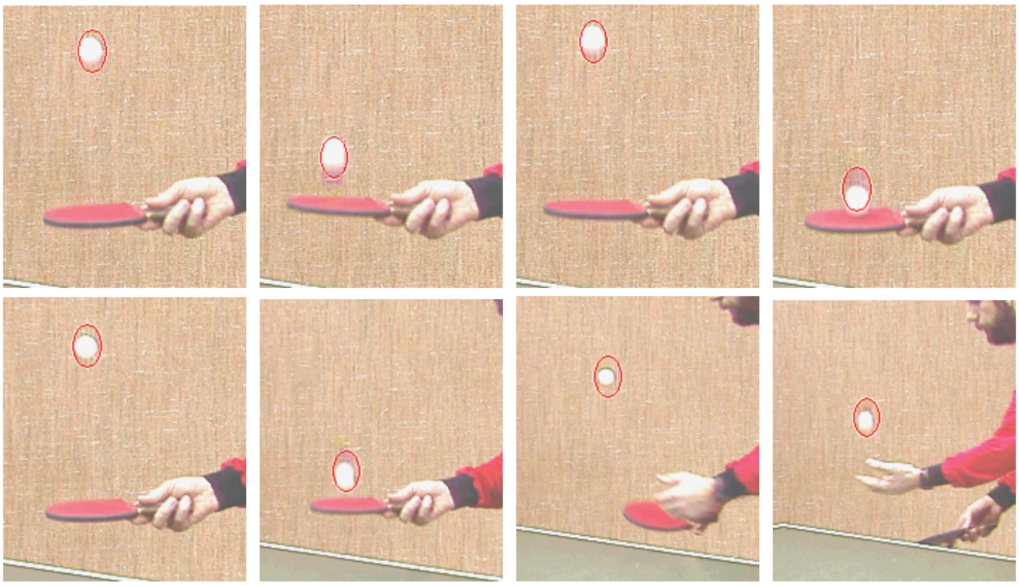


Figure 2.6: Ping-pong ball tracking using kernel based object tracking [4]

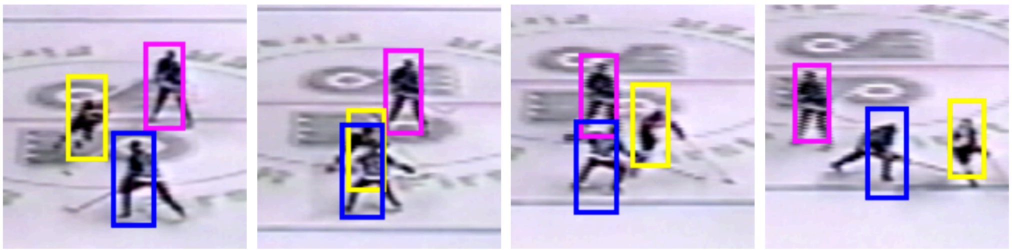


Figure 2.7: Hockey player tracking based on extended BPF tracking algorithm [24]

[20]. Figure 2.6 is the result of utilizing this method to track non-rigid Ping-Pong balls in video clips. The method uses the mean-shift algorithm [33] to maximize kernel estimate of the similarity function and is suitable for problems where the target object is non-rigid, well represented by its color and the fast processing speed is necessary. The kernel-based tracking is commonly used to maximize the Bhattacharyya distance [20; 21], but can be used also to maximize the Kullback-Liebler divergence [34]. Some mean-shift based approaches cope with high values of background colors in the object representation by finding the representation that discriminates the object from the background [35; 22]. The CamShift (Continuously adaptive mean-shift) method [10] uses mean-shift to find the maxima of the background back projection and proposes the object scale adaptation as well.

Recently, particle filters have proved to be a powerful tool for tracking sportsmen [19, 26, 27, 28]. Particle filters tracker is a hypothesis tracker, that approximates the filtered posterior distribution by a set of weighted particles. It weights particles based on a likelihood score and then propagates these particles according to a motion model. The main strength of particle filters is their simplicity and extensibility. On the other hand, the computational complexity limits the method within the tasks of multi-target tracking. In [19] X. Mei et al. propose a tracking method using sparse representation for robust visual tracking. The model tracking as a sparse approximation problem and solve it through an $l_1$regularized least-squares approach. Okuma et al. [25] use the BPF (Boosted Particle Filter) to track players in the hockey games. Cai et al. [24] extended the BPF by introducing bi-partite matching to associate detections with targets. This algorithm is important because it represents one of the first solutions that has been successfully applied in detection and tracking of sport players. Figure 2.7 illustrate the accomplishment for multiple player tracking by utilizing extended BPF algorithm. The particle filter technique is adopted and modified to fit into the multi-target tracking framework. A rectification technique is employed to find the correspondence between the video frame coordinates and the standard hockey field coordinates so that the system can compensate for camera motion and improve the dynamics of the players. A global nearest neighbor data association algorithm is introduced to assign boosting detections to the existing tracks for the proposal distribution in particle filters. The mean-shift algorithm is embedded into the particle filter framework to stabilize the trajectories of the targets for robust tracking during mutual occlusion.

Tracking of multiple players using a graph representation is employed in [30]. When using the graph-based tracking, first the objects are segmented and their properties (e.g. velocity, color) and identified. Second, a graph representing the correlations between objects in the frame is created and finally, the localization of the objects in next frame is performed by finding a mapping between objects using the graph and the object properties. This approach is well suitable for tracking sportsmen, but the limitation of the graph-based tracking is that an efficient segmentation must be employed first. V. Pallavi in [30] combined shot classification, player detection and player tracking to accomplish their goal for tracking soccer player among soccer game video. Figure 2.8 shows the block diagram of V. Pallavi's graph-based multiple player tracking system.

The Lucas-Kanade tracking approach was adopted in [31] and [29]. The Lucas-Kanade tracking is based on matching the templates. In the method is used for feature point tracking, in Flock of Trackers [29] many local trackers are placed on the object and the final localization is determined by the robust combination of their displacements. The tracking algorithm is based on the search of paths in a graph defined by blobs representing segmented players. The different cases of occlusions or contact of these players are treated by splitting the corresponding blobs and taking into account features such as number of components, area of the blobs, players trajectory, and so forth. The advantage of Lucas-Kanade method is its robustness for tracking object with detailed features, whereas a limitation is that it tends to fail when the tracked object is non-rigid or when the sequence is not well detailed.

In [32], the multiple-hypothesis tracking approach for tracking multiple soccer players is used. The players are first segmented as blobs and then localized by color template matching. Then the tracker exhaustively considers possible associations between tracks and the blobs with comparing the hypothesis. Basically, this is probabilistic vision-based tracking system for estimating the position of players in football broadcasting game. The algorithm is fast but the segmentation must be employed first as in the graph-based tracking approach. Since the football game has lower visual sensitivity than basketball game, the method is not suitable for addressing our problem.

In real applications it is enough to just determine movements of single player in particular event time. Thus, the target of this thesis mainly is in particular tracking the selected player's movement in video frames, and the video quality aim is mainly for low resolution basketball game videos. For this problem, we need to find out efficient and low complexity algorithm that can handle single player tracking among their team member that are wearing similar clothes and opponent team members in low resolution qualities. From this reason, we will utilize the algorithm of compressive tracking which will be discussed on next chapter.

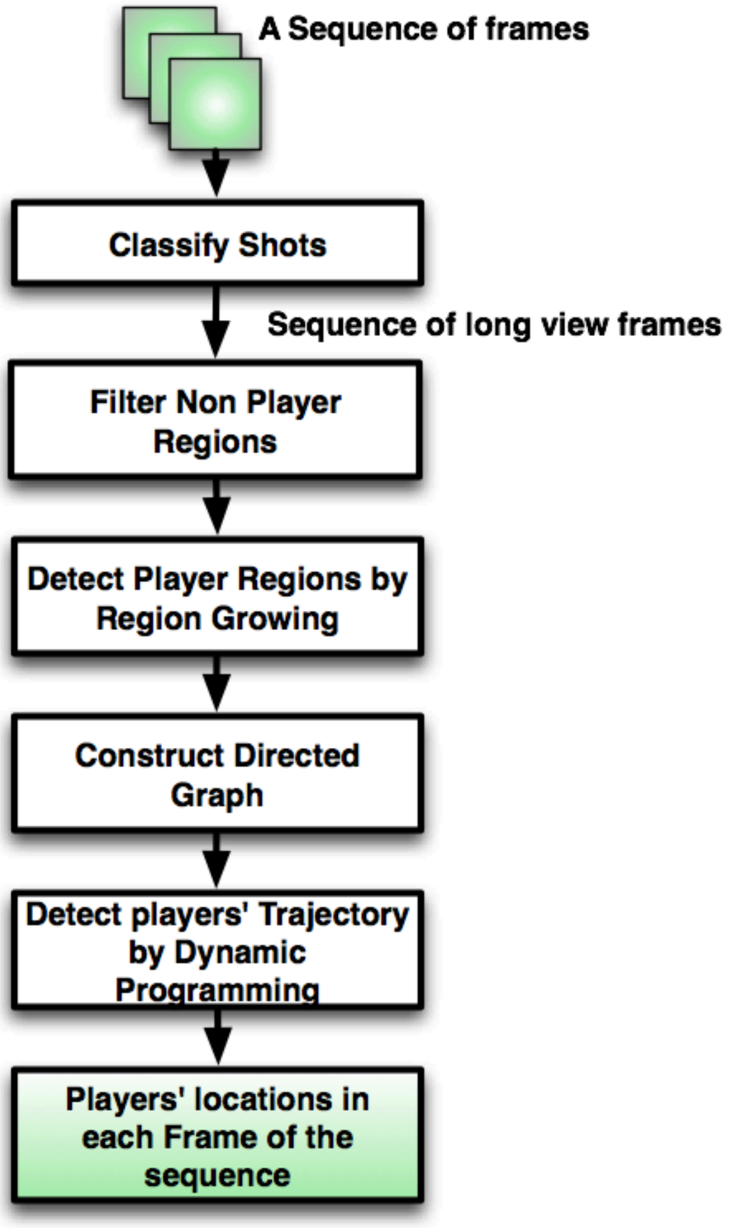

A Sequence of frames
Classify Shots
Sequence of long view frames
Filter Non Player Regions
Detect Player Regions by Region Growing
Construct Directed Graph
Detect players' Trajectory by Dynamic Programming
Players' locations in each Frame of the sequence

Figure 2.8: Block diagram of V. Pallavi's proposed system based on a graph representation

3. PROPOSED METHOD

In this chapter, we propose play-field detection and single player tracking algorithm methodologies. Above mentioned algorithms have their own advantages and disadvantages as we presented in Chapter 2. Due to the requirement that prototype algorithms will be implemented in mobile device, we utilize efficient and robust algorithms to present computer vision application in sports intelligent field. Speed up robust features detection [36] algorithm, which magnificently extract features and detect object according to samples data assists us to achieve play-field precise detection for further overlaying processing, and it is scale, rotation and affine invariance. For solving the challenges in basketball player tracking, we apply highly efficient and low computational complexity compressive tracking algorithm [42] to address single player tracking issues so that algorithm is robust to fast movement, noises and motion blur in low resolution basketball video.

3.1 Speed Up Robust Features

The Speed Up Robust Features algorithm (SURF) [36] is scale and rotation-invariant interest point detector and descriptor which runs at low computational cost. SURF is one of the most used scale and rotation invariant interest point detector and descriptor that helps to find discrete matches between two images for image registration, object recognition and 3D pattern recognition and machine vision tasks. The distinctive features of SURF from other image descriptors are use of hessian matrix for detectors and selection of distribution based descriptors. It uses integral images to improve the speed. The key points are detected by using a Fast-Hessian matrix.

The SURF detector was conceived to ensure high speed in three of the feature detection steps: detection, description and matching. The descriptor describes a distribution of Haar-wavelet responses within the interest point neighborhood. The performance of SURF increased by using an intermediated image representation known as the integral image. The integral image is computed rapidly from an input image and is used to speed up the calculation of any upright rectangular area. SURF reveal strong capability to be a successful detection when the image shows the object with different transformations, mainly scale and rotation, or when parts of the object are occluded. The major computational steps of SURF algorithm are presented in subsections below.

3.1.1 Fast interest point detection

The SURF feature detector is based on the Hessian matrix. The determinant of the Hessian matrix is used to determine the location and scale of the descriptor [37]. The Hessian matrix is defined as $H(x, \sigma)$ for a given point $\mathbf{x} = (\mathbf{x}, \mathbf{y})$ in an image by Equation 3.1

$$H(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma) L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) L_{yy}(x,\sigma) \end{bmatrix} \tag{3.1}$$

where $L_{xx}(x, \sigma)$ is the convolution of the Gaussian second order derivative $\frac{\partial^2}{\partial x^2} g(\sigma)$ with the image $\mathbf{I}$ at the point $\mathbf{x}$ and similarly for $L_{xy}(x, \sigma)$ and $L_{yy}(x, \sigma)$. The SURF approximates second order derivatives of the Gaussian with box filters. Image convolutions with these box filters can be computed rapidly by using integral images.

The determinant of the Hessian matrix is written as Equation 3.2

$$Det(H_{approx}) = D_{xx} D_{yy} - (0.9 D_{xy})^2 \tag{3.2}$$

In order to localize interest points [38] in the image and over scales, a non maximum suppression in a $3 \times 3 \times 3$ neighborhood is applied. Finally, the found maxima of the determinant of the Hessian matrix are then interpolated in scale and image space.

3.1.2 Interest point descriptor

For the purpose of obtaining the SURF descriptor, which is extracted from an image, we need to assign an orientation and estimate dominant orientation. To begin with, assigning an orientation is based on the information of a circular region around the detected interest points[36]. The orientation is computed using Haar-wavelet response in both x and y direction. Once the Haar-wavelet responses are computed they are weighted with a Gaussian with $\sigma = 2.5s$ centered at the interest points [39]. Then, the dominant orientation is estimated by summing the horizontal and vertical wavelet responses within a rotating wedge which covering an angle $\frac{\pi}{3}$ in the wavelet response space. The resulting maximum is then chosen to describe the orientation of the interest point descriptor. In this step, the region is split up regularly into smaller square sub regions. Then a few simple features at sample point are computed for each sub-region. The horizontal and vertical wavelet responses are summed up over each sub-region to form a first set of entries to the feature vector.

The responses of the Haar-wavelets are weighted with a Gaussian centered at the interest point in order to increase robustness to geometric deformations and the

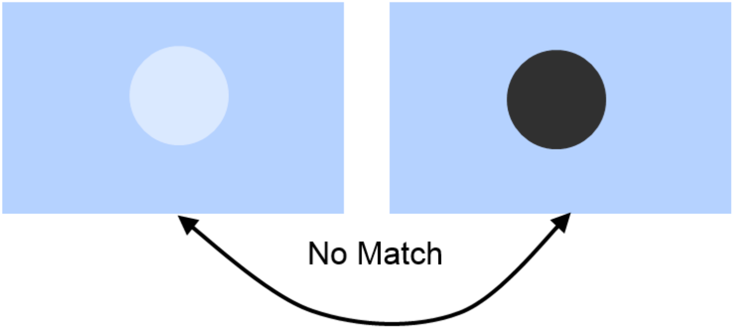


Figure 3.1: The candidate is not considered as a valuable match, if the contrasts between two interest points is different (ie. dark on light background vs light on dark background)

wavelet responses in horizontal d_x and vertical directions d_y are summed up over each sub-region [36]. Furthermore, the absolute values $|d_x|$ and $|d_y|$ are summed in order to obtain information about polarity of the image intensity changes. Therefore each sub-region has a three-dimensional descriptor vector as follow

$$V = (\sum d_x, \sum |d_x|, \sum |d_y|) \tag{3.3}$$

where d_x denotes the horizontal wavelet response and d_y the vertical response. The resulting descriptor vector for all 4×4 subregions is of length 64.

3.1.3 Fast index for matching

The Laplacian for the underlying interest point is included for fast indexing during the matching stage. Typically, the interest points are found at blob-type structures [40]. The sign of the Laplacian distinguishes bright blobs on dark backgrounds from the reverse situation. This feature is available at no extra computational cost as it was already computed during the detection phase. In the matching stage, we only compare features if they have the same type of contrast. For instance, if the contrast between two points is different the candidate is no considered a valuable match, as Figure 3.1 shows.

Hence, this minimal information allows for faster matching, without reducing the descriptor's performance. The KD-tree [41] algorithm is used to match the features of the query image with those of the database images. The KD-tree with BEST bin

First(BBF) search algorithm is used for indexing and matching the SURF features. The KD-tree is a kind of binary tree in which each node chooses a dimension from the space of the features being classified. The BBF algorithm uses a priority search order to traverse the KD-tree so that bins in feature space are searched in the order of their closest distance from the query. The k-approximate and reasonable nearest matches can be returned with low cost by cutting off further search after a specific number of the nearest bins have been explored [40]. The Voting scheme algorithm is used to rank and retrieved the matched images.

Above mentioned SURF algorithm computational steps build foundation of our play-field detection. SURF provide efficient features detection, description and matching to fulfill find out play-field precise location in occluded, scale, rotation and affine changed basketball game video in our database. The detail of play-field detection implementation is carried out in chapter 4.

3.2 Compressive Tracking

Despite that numerous algorithms which are related to tracking have been developed and improved by a lot of scholars and developers, object tracking remains a challenging problem due to appearance change caused by pose, illumination, occlusion, and motion, among others. An effective appearance model is of prime importance for the success of a tracking algorithm that has been attracting much attention in recent years.

As a matter of fact, it is a challenging task to develop effective and efficient appearance models for robust object tracking due to factors such as pose variation, illumination change, occlusion, and motion blur. Existing tracking algorithms have similar common which often update models with samples from observations in recent frames. Even though much success has been demonstrated, numerous issues remain to be addressed. Due to the data-dependent of current tracking models [23, 26, 30] and existence for the sufficient amount of data for algorithms to learn at the outset. Beside, tracking algorithms often encounter the drift problems.

Compressive tracking is a simple yet effective and efficient tracking algorithm with an appearance model [43] based on features extracted from the multi-scale image feature space with data-independent basis. Its appearance model employs nonadaptive random projections that preserve the structure of the image feature space of objects. A very sparse measurement matrix is adopted to efficiently extract the features for the appearance model, and then compress samples of foreground targets and the background using the same sparse measurement matrix. The tracking task is formulated as a binary classification via a naive Bayes classifier with online update in the compressed feature domain. The proposed compressive tracking algorithm runs in real-time and performs favorably against state-of-the-art algorithms

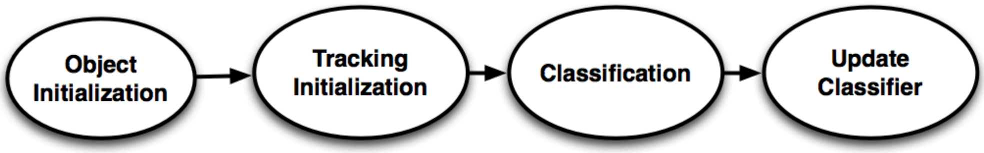


Figure 3.2: Construction of compressive tracking algorithm.

on challenging sequences in terms of efficiency, accuracy and robustness.

Compressive tracking is an efficient tracking algorithm with an appearance model based on features extracted in the compressed domain. The main components of compressive tracking algorithm are initialization of the object tracking classifier with targeted object, updating the classifier and tracking system. The appearance model of compressive tracking is generative as the object can be well represented based on the features extracted in the compressive domain [44], and features due to separate the target from the surrounding background via a naive Bayes classifier the model is also discriminative. In compressive tracking algorithm the features are selected by an information-preserving and nonadaptive dimensionality reduction from the multi-scale image feature space based on compressive sensing theories.

The tracking problem is formulated as a object detection task shown in Figure 3.2.

Below, we begin with presenting some preliminaries about compressive sensing which leads to the compressive tracking algorithm and details of the proposed algorithm.

3.2.1 Preliminary concepts

Random projection

A random matrix $R \in \mathbb{R}^{n \times m}$ whose rows have unit length projects data [45] from the high-dimensional image space $x \in \mathbb{R}^m$ to a lower-dimensional space $v \in \mathbb{R}^n$

$$v = Rx \tag{3.4}$$

where $n \ll m$. Ideally, we expect R provides a stable embedding that approximately preservers the distance between all pairs of original signals. The Johnson-Lindenstrauss lemma [46] states that with high probability the distances between the points in a vector space are preserved if they are projected onto a randomly selected subspace with suitably high dimensions.

Baraniuk et al. [50] also proved that the random matrix staifying the Johson-

Lindenstrauss lemma, and it holds true for the restricted isometric property in compressive sensing. Therefore, if the random matrix R in Equation 3.4 satisfies the Johnson-Lindenstrauss lemma, we can reconstruct x with minimum error from v with high probability, if x is compressive signal such as image frames. In this context, we can ensure that v preserves almost all the information in x. This is very strong theoretical motivate to analyze the high-dimensional signals via low-dimensional random projections in a way of efficiency.

Random Measurement Matrix

As used in numerous works recently [42,49,51], a typical measurement matrix satisfying the restricted isometric property is the random Gaussian matrix $R \in \mathbb{R}^{n \times m}$ where $r_{i,j} \sim N(0,1)$. Since the memory and computational loads are still large when m is large if the matrix is dense, a very sparse random measurement matrix with the following entries is adapted

$$r_{i,j} = \sqrt{s} \times \begin{cases} 1 \;\; with\,probability\; \frac{1}{2s} \\ 0 \;\; with\,probability\; 1 - \frac{1}{s} \\ -1 \;\; with\,probability\; \frac{1}{2s} \end{cases} \tag{3.5}$$

It is proved in the reference [46] that, this type of matrix with $s = 2\; or\; 3$ satisfies the Johnson-Lindenstrauss lemma. Besides, this matrix is very easy to compute which requires only a uniform random generator which helps to avoid unnecessary computations. In addition, when $s = m/\log(m)$, the random projections are almost as accurate as the conventional random projections where $r_{i,j} \sim N(0,1)$.

Dimensionality Reduction and Features Analyzing

To deal with the scale problem, we represent each sample $z \in \mathbb{R}^{w \times h}$ by convolving z with a set of rectangle filters at multiple scales $h_{1,1}, ..., h_{w,h}$ defined as

$$h_{i,j}(x,y) = \begin{cases} 1, \;\; 1 \leqslant x \leqslant i, \; 1 \leqslant y \leqslant j \\ \quad 0, \;\; otherwise \end{cases} \tag{3.6}$$

where i and j are the width and height of a rectangle filter, respectively. Next we represent each filtered image as column vector in $\mathbb{R}^{\underline{wh}}$, then concatenate these vectors as a very high-dimensional multi-scale image feature vector $x = (x_1, ..., x_m)^T \in \mathbb{R}_m$ where $m = (\underline{wh})^2$. Also, the algorithm adopt a sparse random matrix R in Equation 3.5 to project x onto a vector $v \in \mathbb{R}^n$ in a low dimensional space. The random matrix R needs to be computed only once and remains fixed throughout the tracking process. For the sparse matrix R in Equation 3.5, the computational load is very light.

Equation 3.7 below represents compressing a high dimensional vector x to a low-dimensional vector v. The measurement matrix R contains negative, positive and zero entries. According to Equation 3.7 we only need to store the nonzero entries in R and the positions of rectangle filters [45] in an input image corresponding to the nonzero entries in each row of R. At the end, v can be efficiently computed by using R to sparsely measure the rectangular features which can be efficiently computed using the integral image method [52].

$$R^{m\times n}\ as\ \begin{bmatrix} r_{1,1} & r_{1,2} & . & . & . & r_{1,m} \\ r_{1,1} & r_{1,2} & . & . & . & r_{1,m} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ r_{n,1} & r_{n,2} & . & . & . & r_{n,m} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \\ . \\ . \\ . \\ \\ x_m \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ . \\ . \\ . \\ v_n \end{bmatrix} as\ v_i = \sum_j r_{ij} x_j. \qquad (3.7)$$

In Equation 3.7, each element v_i in low-dimensional feature $v \in \mathbb{R}^n$ is a linear combination of spatially distributed rectangle features at different scales. Due to the fact that coefficients in the measurement matrix R can be positive or negative as mentioned in Equation 3.5, the compressive features compute the relative intensity difference in a way similar to the generalized Haar-like features [47].

Due to huge number of Haar-like features, computational complexity of this process becomes heavy. By applying boosting algorithms for selecting important features [53,53], this problem can be alleviated. In compressive tracking algorithm, the large set of Haar-like features are compressively sensed with a vary sparse measurement matrix [43]. The compressive sensing theories ensure that extracted features from object preserve almost all the information of the original object. Therefore, we can classify the projected features in the compressed domain efficiently without the curse of dimensionality.

3.2.2 Classifier construction and update

The low-dimensional features obtained, which is $v = (v_1, ..., v_n)^T \in \mathbb{R}^n$ with $m \ll n$, represent each sample $z \in \mathbb{R}^m$. Due to the assumption that all elements in v are all independently distributed [52], we can model them with a naive Bayes classifier [54] by

$$H(v) = \log(\frac{\prod_{i=1}^{n} p(v_i|y=1)p(y=1)}{\prod_{i=1}^{n} p(v_i|y=0)p(y=0)}) = \sum_{i=1}^{n} \log(\frac{p(v_i|y=1)}{p(v_i|y=0)}) \qquad (3.8)$$

where the algorithm is based on the assumption of uniform prior, which $p(y =$

1) $= p(y = 0)$ and $y \in 0, 1$ is a binary variable which represents the sample label. Diaconis et al. [48] stated that the random projections of high dimensional random vectors are almost always Gaussian. Therefore, the conditional distributions $p(v_i|y = 1)$ and $p(v_i|y = 0)$ in the classifier $H(v)$ are assumed to be Gaussian distributed [44] with four parameters $(\mu_i^1, \sigma_i^1, \mu_i^0, \sigma_i^0)$ where

$$p(v_i|y = 1) \sim N(\mu_i^1, \sigma_i^1), \quad p(v_i|y = 0) \sim N(\mu_i^0, \sigma_i^0). \tag{3.9}$$

The scalar parameters in Equation 3.9 are incrementally updated

$$\begin{aligned} \mu_i^1 &\leftarrow \lambda\mu_i^1 + (1 - \lambda)\mu^1 \\ \sigma_i^1 &\leftarrow \sqrt{\lambda(\sigma_i^1)^2 + (1 - \lambda)(\sigma^1)^2 + \lambda(1 - \lambda)(\mu_i^1 - \mu^1)^2} \end{aligned} \tag{3.10}$$

where $\lambda < 0$ is a learning parameter, $\sigma^1 = \sqrt{\frac{1}{n}\sum_{k=0|y=1}^{n-1}(v_i(k) - \mu^1)^2}$ and $\mu^1 = \frac{1}{n}\sum_{k=0|y=1}^{n-1}(v_i(k)$. By utilizing maximal likelihood estimation [45], the above equations can be easily derived.

We will present our implementation of compressive tracking for highlight single player's critical movements in basketball game in the next chapter. In addition to that, more detailed description about compressive tracking will also be illustrated om chapter 4.

4. IMPLEMENTATION

In this chapter, we present our proposed implementation for achieving play-field detection and single player tracking in basketball game video. For efficient and effective of implementation, SURF and compressive tracking algorithm is adopted so that it can meet our goal.

4.1 Play-Field Detection

We choose to use state-of-the-art detection algorithm SURF to detect, locate the court region and overlay the trajectory images to the game video clips. Flow chart in Figure 4.1 illustrates the overview of the play-filed detection steps in our system. We will cover each steps in detail in the following subsections.

4.1.1 Court sample acquiring

Our target is to establish a system for detecting Korisliiga teams' play-field in basketball video. Since each team have various court model (color, texture and court lines etc.), we take sample court images from each team's stadium from the match game video clips. The purpose is to offer samples for locating the play-field and obtaining homograph matrix in an efficient and reliable way. Figure 4.2 shows the court sample images.

At the same time, for the sake of simplicity in locating play-field and obtaining perspective transform, we also tag the four important corner point for establishing the structure of whole court region as Figure 4.3 shows. We will cover handling these sample court images later on.

4.1.2 Trajectory image

Here, we generate trajectory images according to annotation file and location data of targeted match. Trajectory images are the players' trajectory data visualized in 2D court from each team, thus they can illustrate the players movement trajectory from the starting of the basketball event (shot, rebound and turnover .etc.) to completion of the event. More precisely, the most important basketball event is shot event which is our aim to observe players' tactics in the most critical event time. As Figure 4.4 illustrates the trajectory image that we produced is according

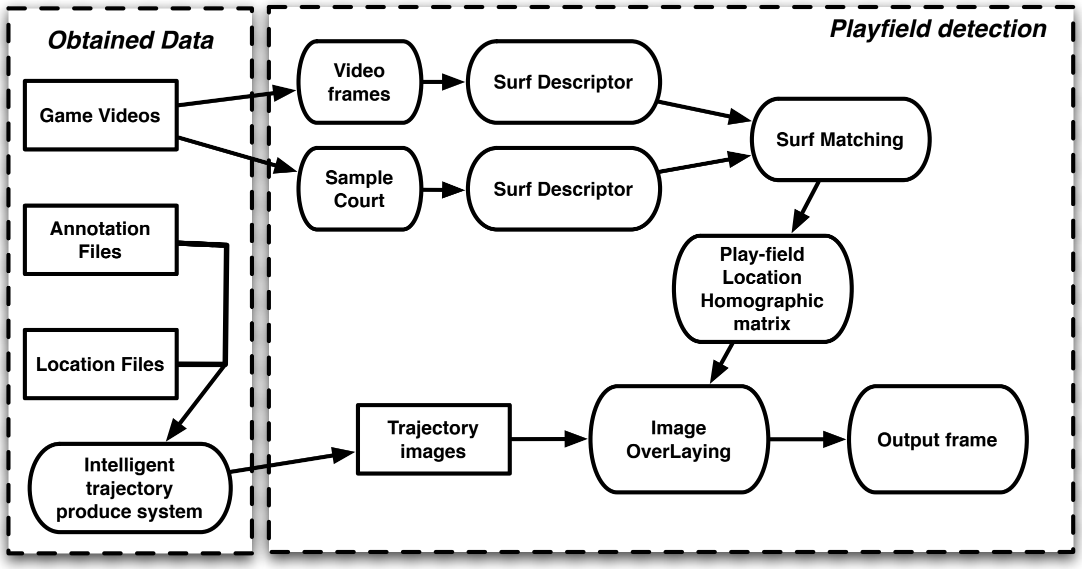


Figure 4.1: System overview of play-field detection

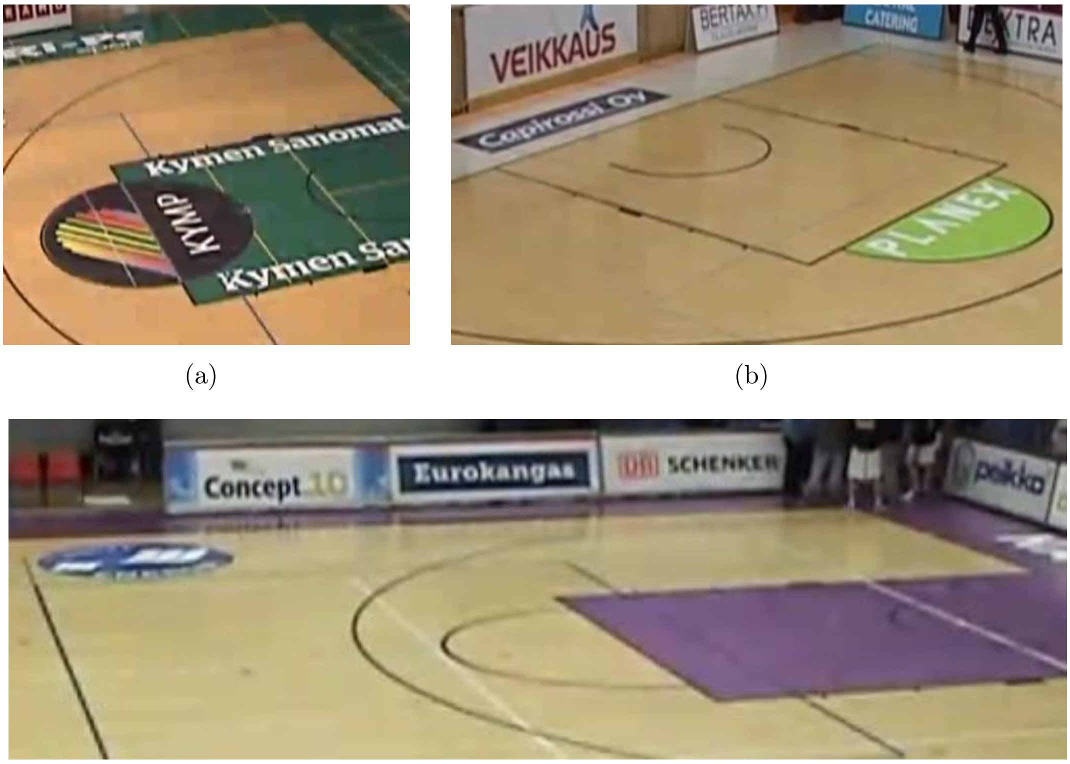
to annotation file and location data. This will be the source of images which will be overlaid onto the original video frame.

As we tagged sample images corner, we also need to record the eight corner points for the purpose of overlaying trajectory image onto a video frame. These eight corner points are corresponding to court sample important corners. Due to the generation of trajectory data in the system, all the trajectory images share the same dimensions and same court positions. Thus we only have to take one trajectory of images eight corner points. Mapping two images using given tagged point coordinates are much cheaper.

4.1.3 SURF algorithm

As we mentioned in chapter 3 section 1 the Speed Up Robust Features, SURF algorithm includes fast interest point detection, interest point descriptor and fast indexing for matching. Using the cross platform OpenCV software library, SURF algorithm is implemented in a smooth way.

In fast interest point detection, the key points are detected by using a fast Hessian matrix. We achieve key points detect by using the SURF detector from the OpenCV library. This lends itself to the use of integral images, which reduces the computation time drastically. SURF detector is based on on the Hessian matrix because of its good performance in accuracy. Precisely, SURF detect blob-like structures at locations where the determinant is maximum. Interest points need to be found at different scales, not least because the search of correspondences often requires their



(a)

(b)

(c)

Figure 4.2: Court sample images for play-field detection. (a)KTP home court, (b)Honka home court, (c)Lahti home court

comparison in images where they are seen at different scales. Scale spaces are usually implemented as an image pyramid. The images are repeatedly smoothed with a Gaussian and then sub-sampled in order to achieve a higher level of the pyramid. Thus scale space representation is demanded for interest point detection step [36]. In order to localize interest points in the image and over scales, a non-maximum suppression in a $3 \times 3 \times 3$ neighborhood is applied.

The interest point descriptor describes a distribution of Haar-wavelet responses within the interest point neighborhood, in OpenCV we used SURF descriptor extraction library. The first step consists of fixing a reproducible orientation based on information from a circular region around the interest point. For this purpose, SURF computes Haar-wavelet responses in horizontal and vertical directions. Then, SURF constructs a square region aligned to the selected orientation and extract the SURF descriptor from it.

At last but not the least, fast indexing for matching the Laplacian which is the trace of the Hessian matrix, for the underlying interest point is included for fast indexing during the matching stage. In this case we matched descriptor vectors by using FLANN matcher, which is Fast Approximate Nearest Neighbor Search

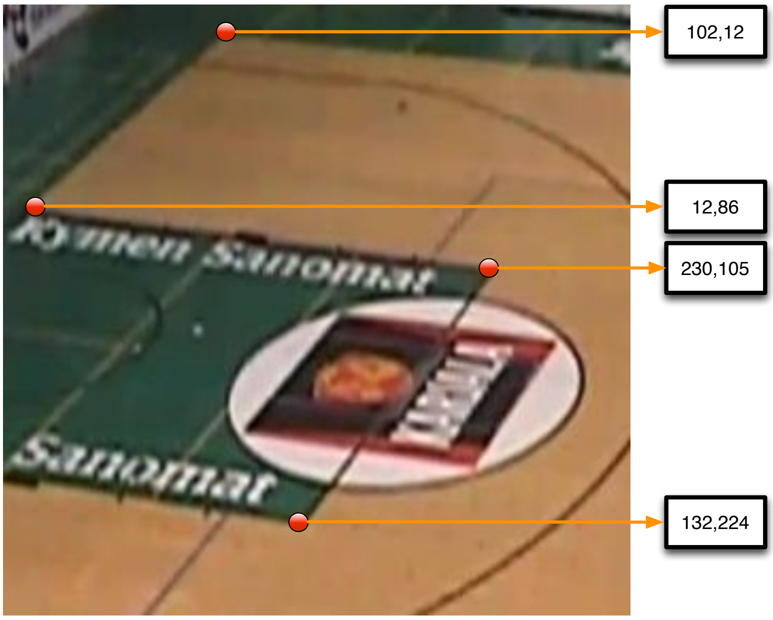


Figure 4.3: Mark the four important corner for perspective transform, for example, the coordinates are pointed out at right side of the figure

OpenCV Library. Basically, it is a library for performing fast approximate nearest neighbor searches in high dimensional spaces. It contains a collection of algorithms to work best for the nearest neighbor search and a system for automatically choosing the best algorithm and optimum parameters depending on the dataset.

In basketball shot event the camera always focus on half of the play-field, in this scenario we will differentiate right half court and left half court according to compare maximum good match key points. We have two court samples from each team home play-field like in Figure 4.2, by comparing the result of FLANN matcher we can initially locate the court as half right or left half court.

4.1.4 Play-field localize

Among all SURF algorithm outputs, there is a good match vector element which matches above distance threshold (we set it as 0.7). Inside good match vector are key points, which we obtain good match points' coordinates, and then we localize the play-field by finding homography matrix and perform perspective transform. Thus we use an OpenCV library called findHomography, which returns us the homography matrix. After that, we will use four critical corner points to perspective transform into video frame to find out the half court four corner points' coordinates.

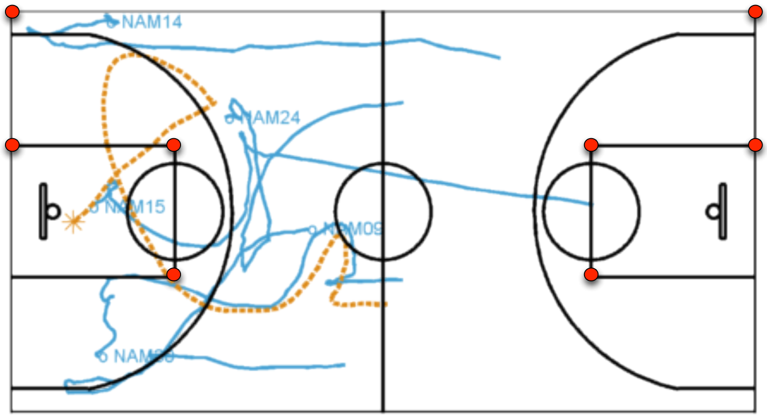


Figure 4.4: Trajectory image that produced from Namika Lahti vs KTP match, blue lines indicate players from Namika Lahti team, orange line indicate basketball. Red points indicate eight important corners.

4.1.5 Image overlaying

In image overlaying part, we will first get the perspective transform matrix, wrap perspective from trajectory image to video frame, obtain player mask and then overlay to video frames. Perspective transform matrix can be calculated from four corner points' coordinates which is the output from play-field localize step and corner points' coordinates from trajectory images. Wrap perspective means apply a perspective transformation to an image, and this first two steps also utilizing OpenCV libraries.

The purpose of obtaining player mask is that we want to overlay trajectory to the game video frame in a condition of not drawing top of the players appearance. This can be done using the dominant color mask algorithm from Chang et al. [14]. Briefly the steps are obtaining dominant color from histogram distribution, applying region growing & shrinking and then using bitwise elimination to obtain non court regions which are mainly players with some noise. After that we again apply region growing & shrinking to obtain overall player mask.

With help of the player mask, we can apply mask to perspective transformed trajectory image and then obtain the image which we overlay to video frame. Beside, we use add weighted function from OpenCV library set of two thresholds which are video source frame and target image weighs. Which lead us to the final output from play-field detection module.

The results of play-field detection will be presented in next chapter followed with assessment and discussion.

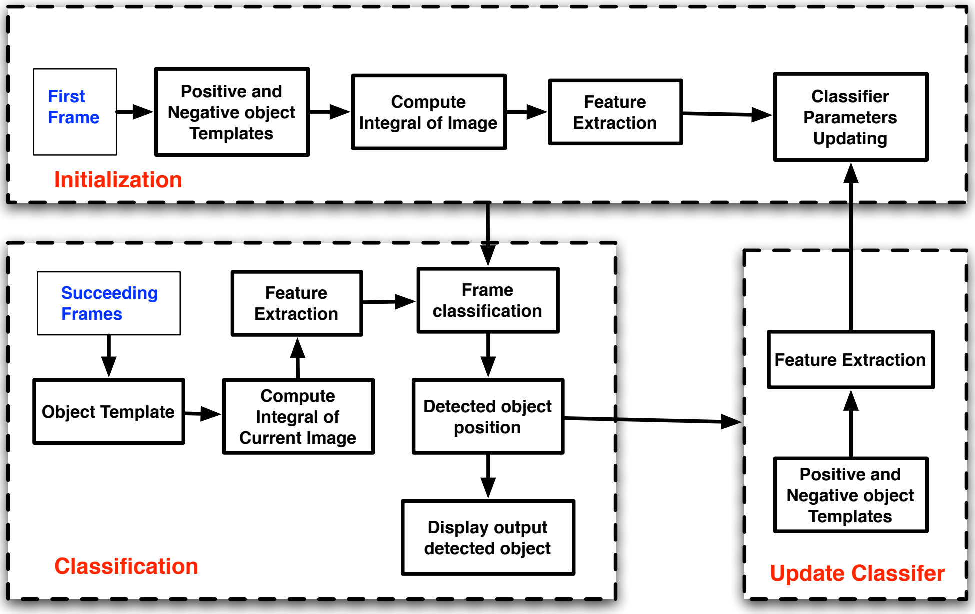

First Frame
Positive and Negative object Templates
Compute Integral of Image
Feature Extraction
Classifier Parameters Updating
Initialization
Succeeding Frames
Feature Extraction
Frame classification
Object Template
Compute Integral of Current Image
Detected object position
Display output detected object
Classification
Feature Extraction
Positive and Negative object Templates
Update Classifer

Figure 4.5: Illustration of single player tracking based on compressive tracking algorithm, 3 main modules: initialization, classification, update classifier

4.2 Single Player Tracking

We have the trajectory image projected to the video frame, we want to check specific player's movement for observing their tactics during the shot event. Thus single player tracking which we utilize compressive tracking algorithm is applied after playfield detection in our system. It is composed of initialization of tracking, processing succeeding frames and delivering the outputs. Outputs consist of zoomed video frame which is focused on player. For implementing above-mentioned Compressive Tracking algorithm, we also used OpenCV functions and libraries to interpret it. Figure 4.5 shows single Player Tracking modules.

4.2.1 Initialization

Within initialization step, object location initialization is accomplished by user selection from video frame. User can use finger to select the region of player's stand. Then compressive tracking classifier will be initialized through positive and negative object templates generation, obtaining integral image, feature extraction for positive and negative object templates and classifier parameter initialization.

In the process of positive and negative object template generation, we obtain the coordinate and position of positive and negative samples around object in the image frame. The positive samples near the current location, negative samples far away from the object. The purpose is to use template to improve classification training and testing scenario. This process is followed by computing integral of image frame. Fast computation is an important advantage of integral image and only several accesses to the integral image are required to extract Haar-like features. Beside, Haar-like features are more robust to illumination changes than color histogram.

Feature extraction part is the appearance model of Compressive Tracking. It is generative as the object can be well represented based on the features extracted in the compressive domain, and features separating the target from the surrounding background. Due to the use of integral image the process is accelerated. We will extract Haar-like features from positive and negative object template.

As in Haar-like feature point selection, applying boosting algorithms for selecting important features [20, 21], we obtain feature points with their weights by using object initial position in the frame. The large set of Haar-like features are compressively sensed with a vary sparse measurement matrix. We adopt random matrix to project Haar-like features to low-dimensional compressive features as in the Equation 3.7. The random matrix needs to be computed only once off-line and remains fixed throughout the tracking process. Following the preliminary concepts in previous chapter, we set $s = m/4$ which makes a very sparse random matrix. For each row of $R^{n \times m}$ where m is object size and n equals to 50 a desired feature number. Furthermore, we only need to store the nonzero entries of R which makes the memory requirement also very light. Then, we gain extracted compressive features.

After obtaining extracted compressive feature values, we will initialize tracking classifier by computing the the mean and variance of the Gaussian classifier using Equation 3.10 for each positive and negative templates.

4.2.2 Classification

As tracking algorithm starts, succeeding image frames will be proceeded for locating tracking target object after initialization step. Due to the correlation between succeeding image frames, we will predict object location based on the previous location and classifier.

Firstly, we generate templates by drawing some samples around current target location, then obtain the coordinates of these sample templates. Since object location will change gradually in succeeding frames, these templates will cover most of the prediction position of object in current frame according to previous frame object location. Secondly, we will compute the integral of current image as we did in initialization step for efficiency and effectiveness of feature extraction. Thirdly, after feature extraction we did in the initialization, we will extract feature values among sample templates and obtain compressive features for further classification purposes. Fourthly, we will locate the object location in current image frame by classifying compressive features based on naive Bayes Classifier we trained in initialization step as Equation 3.8. We determine the object current location with the maximal classification score among these samples templates compressive feature classification results. Then, we can store and visualize the location of object in this frame.

4.2.3 Update classifier

The tracking task is formulated as a binary classification via a naive Byes classifier with online update in the compressive domain. Thus classifier update is a good approximation of the features in the projected space and enhance the classifier for tracking followed frames.

After obtaining the position of player in current frame, update classifier will be performed for improving future tracking performance based on online learning update. It repeats the processes after correctly locate object position in current frame. Similar to the initialization step, positive and negative object templates will be selected, and compressive feature extraction will be perform for each template, then classifier parameters are updated according to feature distribution for positive and negative templates.

This section is dealing with implementation of single player tracking which focus on helping user to obtain player's critical tactics in basketball events. Main modular carried out in logical way for easy interpreting in future works. The performance of single player tracking will be introduced and illustrated with proper assessment and discussion in the following chapter.

5. RESULT AND DISCUSSION

In this chapter, we will focus on the result of our methods which are play-field detection and single player tracking. We present the results in a detailed way and then discuss about advantage and disadvantage of our methodologies. We used the open-source cross platform application OpenCV which contains most of computer vision algorithm libraries and Application Programming Interface (API). The goal is to assess play-field detection and single player tracking performance and accuracy.

5.1 OpenCV

Open Source Computer Vision is a real-time computer vision library released under a BSD license. It includes over a hundred of optimized algorithms for machine vision, signal processing and related tasks, with wrappers for C++, C, and Python. In addition to that, OpenCV is also cross platform library which can work in most PC operating systems and mobile operating systems, and that is why we choose this framework for the development.

OpenCV matrices and multidimensional arrays were used for the main data structures of the features and likelihoods inside the framework. These arrays allowed the efficient handling of sub-arrays, slices, rows, and columns of the data as the subparts could be referenced without any copying. There are also plenty of optimized image processing and machine learning functions which allow us to implement state-of-art computer vision algorithms in a smooth way.

5.2 Data Acquiring

All of the data that are needed for the implementation of this system were acquired from SstatzZ Oy with full copyright reserved. SstatzZ Oy is a young innovative company for advanced sports analysis combining video of the events with the statistics collected from the event. This advanced sports events analysis allows interested third parties to find out interesting events from the sports events. Example from such a third parties can include but not limit to coaches, scouts, broadcasting companies and disciplinary units. Customer references of Sstatzz include Finnish Basketball Association, Finnish Floor-ball Association, Finnish Handball Association, Torpan Pjat Juniority ry and Espoon Honka ry. SstatzZ has been founded in 2008.

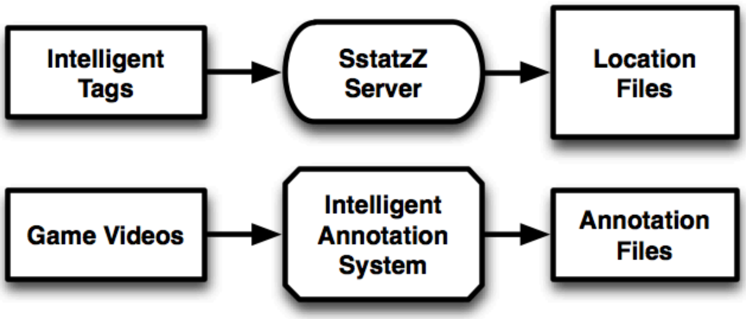


Figure 5.1: This chart indicates location data, game video and annotation files acquiring steps, SstatzZ Oy reserved

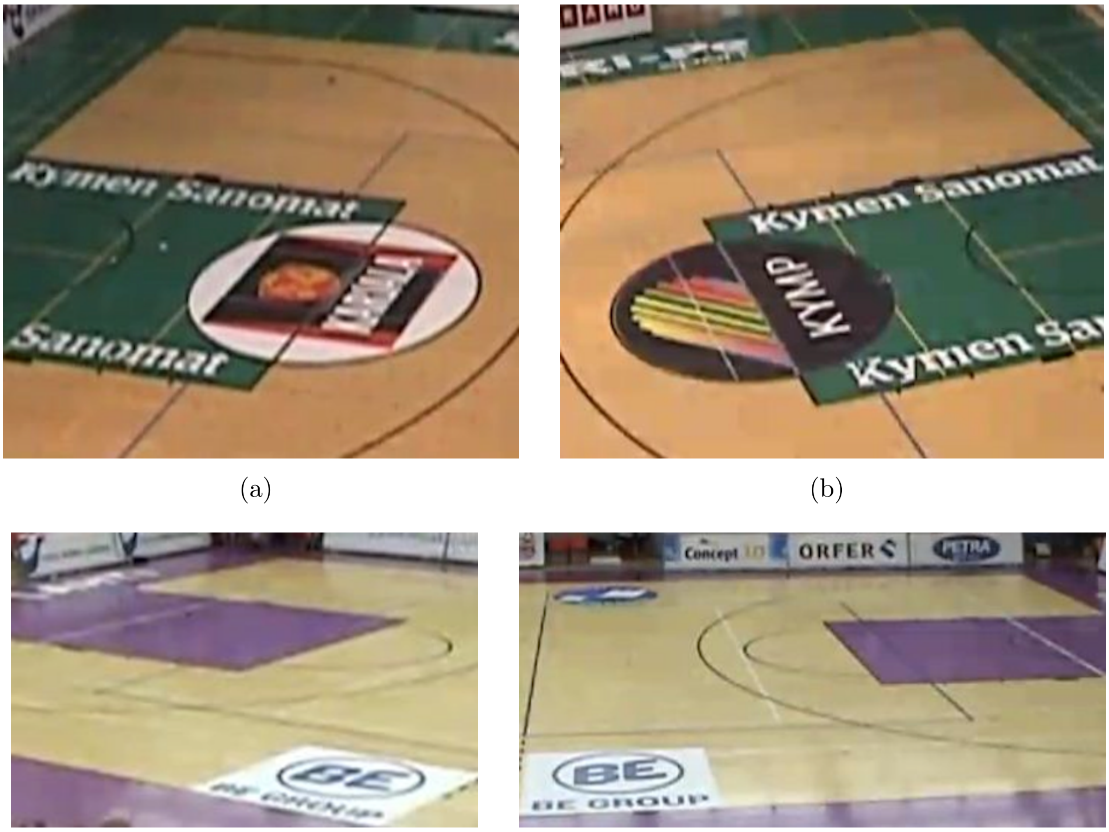
The data are mainly composed of location tag data, basketball videos and annotation files. Location tag data, by which using state-of-the-art technology to obtained each player's movement data in each game, aim is to obtain players' intelligent analyzing report from sports leagues. We will not cover the part of processing tag data and handling tag data. Basketball videos were prepared by SstatzZ Oy from Korisliiga men basketball league matches. Annotation file is basically intend to annotate important events of matches like shot, rebound and fool etc in a time domain. Annotation file is an intelligent system made by SstatzZ Oy team. Chart in Figure 5.1 indicates the data acquiring steps. After obtaining sufficient data, we implement and tested target algorithms: play-field detection and single player tracking.

For experiment purposes, we selected two game from Korisliiga men basketball league matches in 2012-2013 season. There are Namika Lahti versus Korihait on 12.01.2013 in Namika Lahti home court and KTP versus Korihait on 27.03.2013 in KTP home court. For play field detection, we collected right and left court model for each team as Figure 5.2.

As for the game videos, we used low-resolution 720 × 480 pixel frames, frame rate is 40 frames per second. The reason behind of using low-resolution video is that the algorithms will be launched on mobile devices. Since mobile devices have just sufficient resource available, we choose optimal video resolution for smooth experience.

5.3 Result

This section discusses the experimental results for addressing our problem set using our testing data set. For clarification, we will separately describe our results for play-field detection and single player tracking in the following subsections.



(a) (b)

(c) (d)

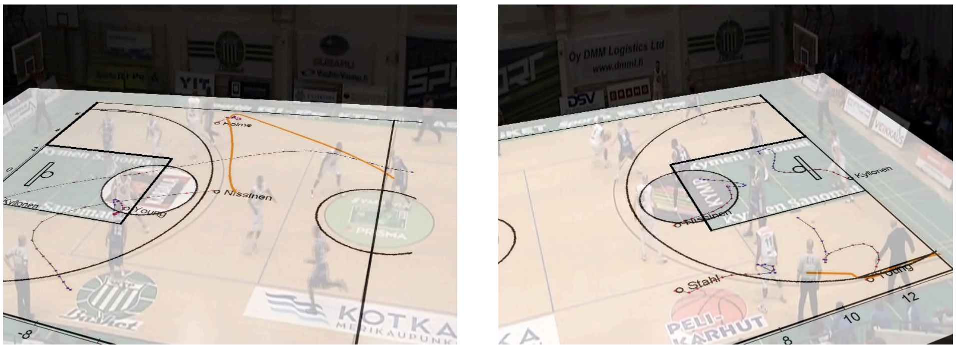
Figure 5.2: Right and left court sample images of each team for play-field detection. (a)KTP left court, (b)KTP right court, (c)Namika Lahti left court, (d)Namika Lahti right court

5.3.1 Play-field detection

By employing the play-field detection algorithm as discussed in section 4.1, we achieved our goal which is play-field region detection, locate the play-field region precisely and graphic overlaying. This method solves the problem of current play-field algorithms, which often leads to uncertain location of play-field because they happen to focus on detecting play-field region.

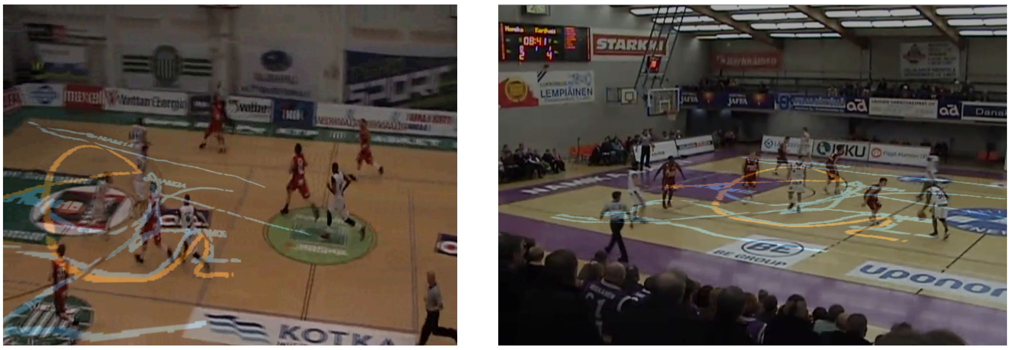
As Figure 5.3 shows, our play-field detection solved our target problem, and correctly overlays the trajectory images. Trajectory images set for testing purposes, did not focus on the visual outcome but for clearly illustrating our achievements. The algorithm first detect play-field court sides which are including right court and left court. Then it precisely locates play-field, using perspective transfer to overlay the trajectory images on video image frame.

For improving visual effect of play-field detection, we transfered our trajectory image to four channel image which contain RGB value with alpha value. This



(a) (b)

Figure 5.3: Right and left play-field detection of KTP home court. (a)KTP left court, (b)KTP right court



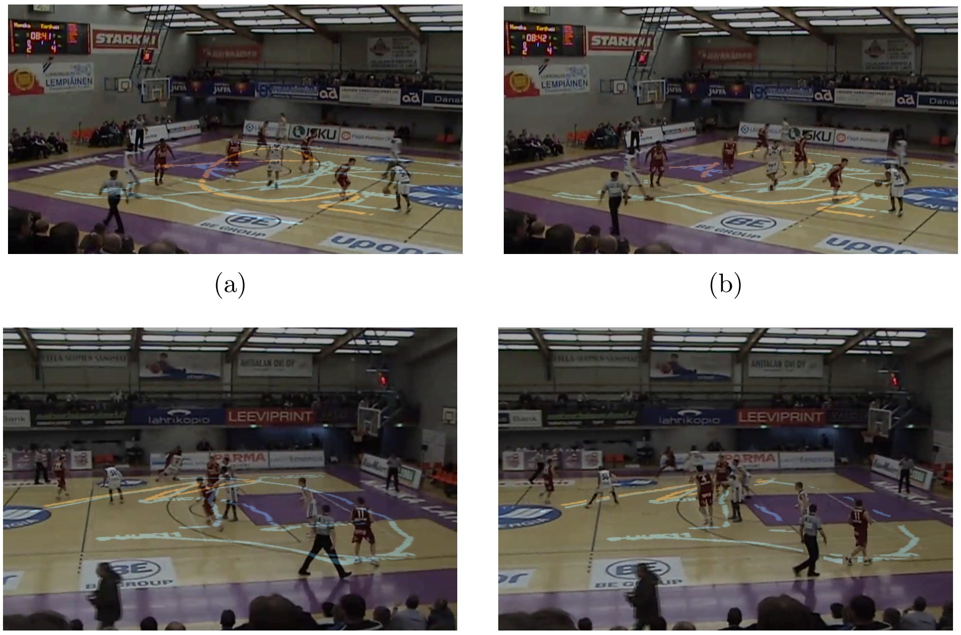
(a) (b)

Figure 5.4: KTP and Namika Lahti home courts play-field detection and overlaying (a)KTP home court, (b)Namika Lahti home court

allow us to overlay trajectory image without disturbance of original video frames. RGB stands for red green blue color model component, and extra alpha channel is normally used as an opacity channel. Also, we modified our trajectory image so that the trajectory of player are solid lines. Figure 5.4 presents KTP and Namika Lahti play-field detection in smooth overlaying methods. In this scenario, we not only improved viewer's experience but also hide minor errors while locating the play-field.

For the purpose of visualizing the game frame for further processing, we also need to eliminate the players from overlaying region by simply adopting dominant color detection. This process helps to recognize player on the overlaid frame. Dominant color detection follows the histogram computation and selects the maximum number of bins. Figure 5.5 presents the results of before applying and after applying player subtraction on overlaid region.

In terms of the timing, the proposed algorithm runs in our final implementation



(a)

(b)

(c)

(d)

Figure 5.5: Namika Lahti home court play-field detection and overlaying (a)Left court overlaying without player mask, (b)Left court overlaying with player mask, (c)Right court overlaying without player mask, (d)Right court overlaying with player mask

Table 5.1: Quantitative measure of play-field detection for our dataset game videos

Home Court	Shot Events Accuracy %	Rebound Events Accuracy %
KTP	82.4	85.7
Namkia Lahti	79.3	81.1

as an Ipad application at about 2.35 seconds per frame. This is nowhere close to what it needs to be to run in real time, but it would be potentially suitable for post processing applications. Furthermore, not a lot of work was done in optimizing the algorithm, and definitely implementing the approach in other languages could potentially yield large timing benefits. For improving performance we could use high level programming like multi-thread programming and resource managing mechanism to optimize this algorithm.

Overall, the SURF detector is by far the most time consuming detector. The largest contributor to timing is the function that iterates over the initial image, SURF descriptor extraction, and SURF matching. This takes about 1.37 seconds per frame. This seems to be more of an issue with implementation within C++ and cross platform in Objective C. Other large contributors to the time in the Player subtracter which is using dominant color detection in OpenCV.

In addition to that, Table 5.1 shows the quantitative measure on the detection accuracy, which is calculated based on the ratio of the total number of correctly detected events minus the number of wrongly detected events over the number of the ground-truth play-field pixels. For this purpose of, we used shot and rebound events which are the most significant moments in basketball for collecting success rate information. There are 94 shot events with 132 rebound events in KTP game, and 103 shot event with 141 rebound events in Namkia Lahti game.

As we can observe, KTP game detection accuracy higher is than Namkia Lahti. The reasons are KTP court is more observable and camera movement is more stable than Namkia Lahti. Also we noticed, that rebound events have more accuracy than shot events. The critical reason is that most of the rebound events take place in clear right side or left side of court which is more clearly to detect, and some shot events might take place in half court region which is our undetected spot of court region.

5.3.2 Single player tracking

Here, we present our result by applying compressive tracking algorithm to track single player in KTP and Namkia Lahti's game. The mechanisms of the algorithm are introduced in chapter 4 section 2 in detail. The processes start with initializing target player by selecting player's location in video frame using finger input from user. Next the classifier is being set up and initialized for tracking target player in succeeding frames as presented Figure 5.6. This implementation addresses the player tracking in high intensive basketball game video even in low resolution video.

For pointing out the target player, we zoomed video frame to be presented for catching player's critical movements as in Figure 5.6 and Figure 5.7. In Figure 5.7, the player is being tracked in low resolution video condition successfully. Video frames usually occur with motion blur and noise which makes it difficult to track player, but compressive tracking algorithm is showing to be robust to motion blur and noise from our observation point of view.

Difficulties in tracking objects also can arise due to abrupt object motion, changing appearance patterns of both the object and the scene. As shown Figure 5.8 and Figure 5.7 reveal, the target player due to fast movement in basketball game, change his pose and appearance in half a second very rapidly. But our proposed tracking system correctly follow target player in succeeding frames.

Beside object-to-object, object-to-scene occlusions, and camera motion there is additional source of breaks in tracking the player's movement. In Figure 5.8, images from (c) to (f) did appear object-to-object occlusions both with team mate and opponent team player. Since the tracking classifier learns and updates from each succeeding frame, this helps to track down the player in noisy frames. In addition

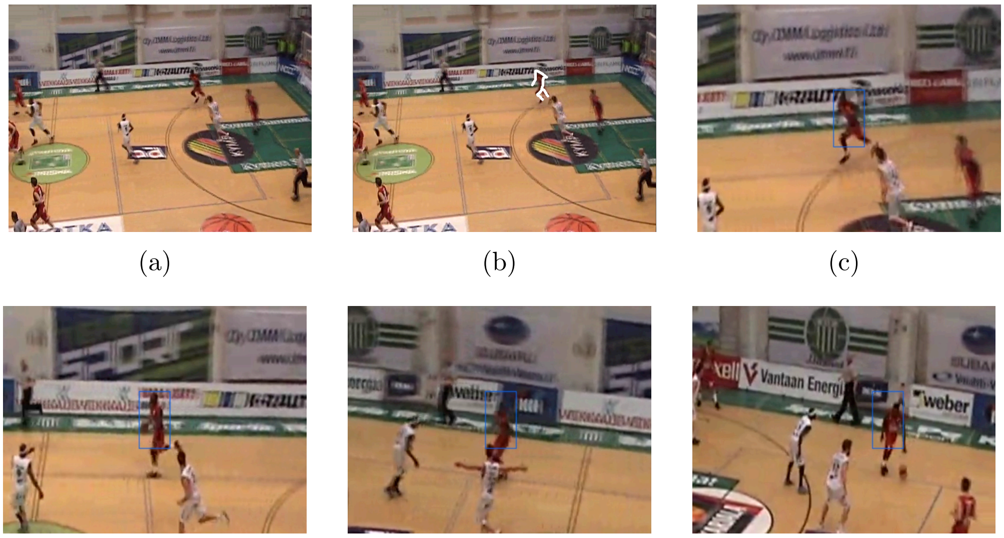

(a) (b) (c) (d) (e) (f)

Figure 5.6: Illustration of single player tracking series in KTP game, (a) original frame, (b) User selection target player, (c) first frame after initial frame, (d) 22th frame, (e) 50th frame, (f) 75th frame

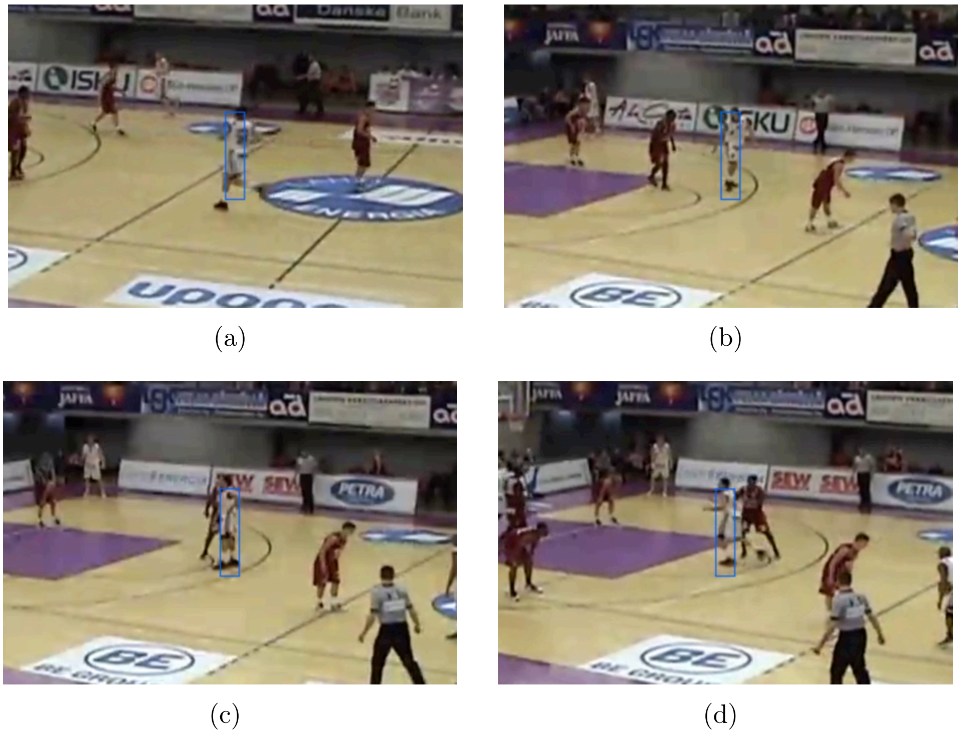
Table 5.2: Performance measurement of single player tracking in our dataset game videos

Home Court	Success rate %	Frames Per Second
KTP	82.3±7.5	0.45
Namkia Lahti	80.7±8.2	0.44

to that, Figure 5.9 also illustrate such difficult tracking moments in basketball game such as teammate occlusions, opponent players occlusion and occlusions with scenes.

Although our method for single player tracking is shown to be robust to usually difficulties in motion blurring, camera motion and occlusions, and it has good success rate for tracking single player during basketball game. For the purpose of obtaining our methods accuracy from our video dataset, we tracked each player during basketball events and recorded success rate information. Every single event are approximately around 5 second, or 200 frames. We selected 20 tracking events form both game videos, they contain 4000 frames from which we manually check tracking results and calculate the performance rate. Table 5.2 shows our single player tracking system's performance which is including success rate and frames per second.

From the Table 5.2, it is seen that single player tracking performance is really good for our player tracking problem in the basketball game video datasets. In the KTP game video, the success rate is 82.3±7.5% at 0.45 frame per second performance.



(a) (b)

(c) (d)

Figure 5.7: Present Namkia Lahti game single player tracking. (a) 1st frame, (b) 80th frame, (c) 174th frame (d) 251th frame

Due to KTP game camera motion is stable and rare noise during the video, the average is success rate launched at 82.3%.

There are some moments the tracking success rate drops to 74.8% happen when continuous occlusion occurs and classifier starts to miss the target player. For the Namkia Lahti game video, the average success rate is more or less same as KTP, the difference occur due to noise and rapid camera motion during basketball game.

We will continuously evaluate and discuss about our methodology in the following Discussion section. In addition to that, we will provide our own assessment of the system as well later.

5.4 Discussion

In the above sections, we presented our experimental results with prototype algorithms implementation. The algorithms successfully addressed play-field detection with trajectory images overlay and single player tracking. For the scientific and practical proof of our methods, we evaluate the results by comparing with previous studies methods which are mentioned in Chapter 2. We also provide assessment whether the results correspond to our objectives and whether we succeeded in answering previously defined research goals.

Our play-field detection algorithm goal is to detect the play-field region, point

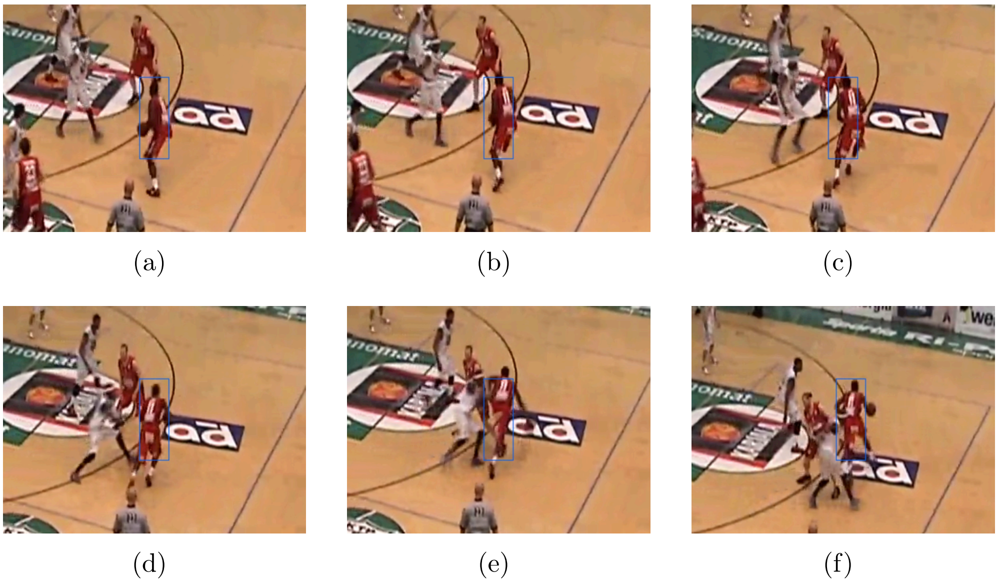
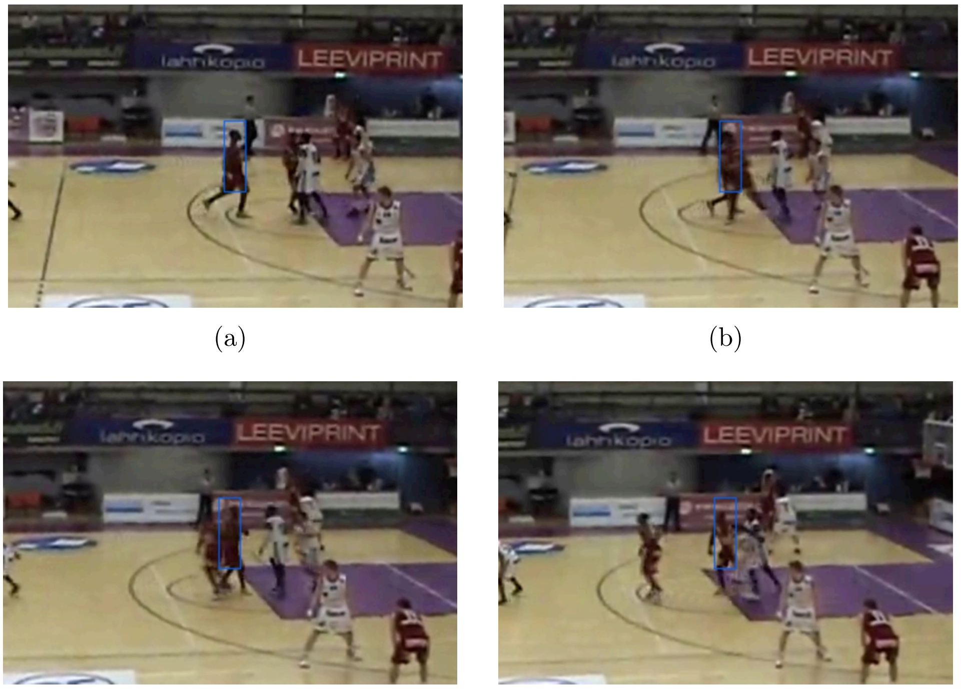


Figure 5.8: KTP game tracking to present robust to the abrupt object motion and changing appearance both in scene and object. (a) 31st frame, (b) 34th frame, (c) 38th frame, (d) 44th frame, (e) 49th frame and (f)58th frame

out right or left court and overlay the trajectory images to improve our product experience by applying magnificent computer vision technology. There are direct court detection [9], dominant color detection [4; 13; 14], Gaussian mixture modeling classifier [12] mentioned which are considered as current state of art addressing the play-field region detection. Although these state-of-the-art methods have good success rate and performance, our method is suitable for implementation in the product which will be integrated in IOS application. Also the other methods are focusing on region detection, and neglect precise play-field coordinates. Our problem is not only to segment out the play-field but also providing overlay. This method is straightforward to help us achieving our target solution. However, our method still needs to be improved both in performance and computational complexity. Since most of important basketball events are carried out in right or left side of play-field, we miss the spot of the middle play-field region at this moment. Also, SURF algorithm will fail when scale and occlusion happens in fast camera moved frame. Due to the system auto detect, finding out play-field region side, and overlaying target trajectory images which is output from the SstatzZ server side, its complexity rises with its operation. The solution will be to optimize algorithm, enhance performance by using multi-threaded computing and resource management at low level computing.

On the other hand, single player tracking which is using compressive tracking algorithm produce good results as we mentioned in section 5.3. We proposed a simple yet robust tracking algorithm with an appearance model based on non-adaptive random projections that preserve the structure of original image space. A very



(a) (b)

(c) (d)

Figure 5.9: Namkia Lahti Player tracking present method robust to common occlusions. (a) 27th frame, (b) 38th frame, (c) 44th frame, (d) 58th frame

sparse measurement matrix was adopted to efficiently compress features from the foreground targets and background ones. The tracking task was formulated as a binary classification problem with online update in the compressed domain. The aim is to visualize important player's critical movements in basketball events like shot, rebound or turnover, because it is hard to pay attention to single player in fast going basketball game video. This solution brings help to the viewer in focusing target player's movement. Kernel based object tracking [4], extended BPF tracking [24] and Flock of tracker [25] are current popular methodology for pursuing single player tracking in sports game videos. Due to our low-resolution and fast going basketball matches, they are not suitable for addressing our problem set in player tracking. Kernel based object tracking [4] is based on non-rigid object tracking, extended BPF tracking [24] high complexity is not matching with cross platform integration with IOS, also Flock of tracker [25] are not suitable for fast going sports like basketball or ice-hockey game player tracking. For optimal solution with proper complexity for implementing our prototype product, we selected compressive tracking. Although our single player tracking achieved our primary goal, we will still try to improve and optimize this method in the near future. Beside, tracking classifier sometimes loose target in special abrupt motion, heavy occlusions and out of plane situation. This algorithm needs to be more efficient and optimized for our target product mobile devices. Enhancing performance will also bring rich experience for the target viewer.

Sports as a one of the most attracted entertainment industries is highly integrated with game, broadcasting, advertising and other industries. Therefore sports field start to utilize the state-of-art computer vision to improve their productivity. Computer vision applications in sports field are integrated in sports commercial tools, broadcasting industries tools, assisting coach tools or even for individual self-improving.

6. CONCLUSION AND FUTURE WORK

This chapter delivers the conclusion of the thesis. First, we will briefly state over problem and context, and present our solutions. Then we mainly focus on our contribution for computer vision application on sports intelligent system, and our own assessment of the results. In addition to that, we will also point out the possible future work and recommendations.

6.1 Conclusion

Our proposed basketball analyzing system based on computer vision demonstrated the ability to extend the current intelligent sports video analysis system. Sports industry, which involves a huge money and resource investment, needs new technology support to improve their productivity and entertainment. Computer vision is growingly applied in broadcast, training, automatic analysis, decision-making and commerce usage for sports industry. Here we presented prototype algorithms which aim to improve sports intelligent product experience.

Play-field detection is primary stage of video analyzing basketball game, and provide extensive foundation for further computer vision process. Play-field detection needs to locate court region, and find its precise coordinates. Our play-field detection algorithm illustrates the whole process that necessary for solving this problems. It contains automatic region detection, and provide precise court coordinates for further steps. We use SURF algorithm which is scale, illumination and affine invariant robust algorithm that help us reach our goal. AThis algorithm still needs to improve and update on the programming side and vision research side to fulfill its mission.

Single player tracking is also considered as a basic stage of computer vision application for sports industries. Tracking are taken in low resolution video context to provide detail movements of player during important events. Our methodology of single player tracking overcomes some of the challenges in sports player tracking. Compressive tracking offers efficient and robust tracking results, due to appearance model and classifier of tracking which is sufficient enough to solve tracking player movement in intensive sports game. The prototype tracking algorithm have shown its good success rate in our data set.

Most of computer vision technology in sports are implemented in industrial products, and are not properly documented due to patents and copyright. Academic works are not easy to come by or participate in the process of researching, developing and implementing scenario. Challenges in sports computer vision application developing and implementing can be reduced by combining experts from academic and industry fields, integrating with new technology like cloud computing and big data solutions.

6.2 Future Work

Future is so magnificent that each of our individual dream start to become true. Thus, we need to contribute our expertise by applying it to real world, by sharing it with others to build more bright future. The following are some possible directions to improve the computer vision analyzing system in sports industries.

Refinement of play-field detection and tracking methodology

Proposed computer vision prototype algorithm for sports industries have their own limitations. Due to the rapid technology development, the quality of product becomes critical. For improving utilization of vision technology, we need to provide solid solution that guarantee quality of performance.

Play-field detection may take place at the initial-process stage, provide court region information as in meta data format. This will help to easily check quality of the algorithms and improve the performance of product. As for tracking, single player tracking also need to improve so that product performance and experience bring customer amazing results. Multiple player tracking is an alternative which can extract all necessary players information and visualize them.

Integration of theoretical and practical knowledge

Theoretical knowledge builds foundation for what can we do, and practical experience enrich how we will do. Thus, we will increase knowledge by doing more research and applying the computer vision field to practical problems. Proposed algorithm can also improve their performance by utilizing solid programming skills. We can introduce multi-thread programming since most of computer or mobile device CPU are becoming multi-core and powerful. Multi-thread computing and resource managing in programming side shall also be concerned.

Observation Technologies

There are more and more new technologies being introduced in nowadays. Cloud computing is one of them, and provide infrastructure, software and platform for individual or business organization. By using cloud computing, we can enhance our computer vision application performance. For instance, heavy complexity computer vision algorithms can be carried out in a cloud which can use cloud server on Google, and can achieve instant results for computer vision application in sports industries.

REFERENCE

[1] Y. Gong, T.S. Lim, H.C. Chua, H.J. Zhang, and M. Sakauchi: Automatic parsing of TV soccer programs. IEEE International Conference on Multimedia Computing and System, Washington, DC, US, pp. 167-174, May 1995.

[2] L. G. Shapiro and G. Stockman: Computer Vision 1st edition. Prentice Hall PTR Upper Saddle River, NJ, USA, pp. 608-609, 2001.

[3] A. Tomita, T. Echigo, M. Knrokawa, H. Miyamori, and S. Iisaku: A visual tracking system for sports video annotation in unconstrained environments. IEEE International Conference on Image Processing, vol.3, pp.242-245, 2000.

[4] A. Ekin and A. M. Tekalp: Robust dominant color detection and color-based application for sports video. IEEE International Conference on Image Processing, vol. 1, pp 21-24, 2003.

[5] L. Xie, P. Xu, S.-F. Chang, A. Divakaran, and H. Sun: Structure Analysis of Soccer Video with Domain Knowledge and Hidden Markov Models. Pattern Recognition Letters, vol. 25, Issue 7, pp. 767-775, May 2004.

[6] M. Luo, Y. F. Ma, and H. J. Zhang: Pyramid wise structuring for soccer highlight extraction. Information, Communications and Signal Processing, and Fourth Pacific Rim Conference on Multimedia. vol. 2, pp. 945 - 949, Dec. 2003.

[7] S. Jiang, Q. Ye, and W. Gao: A new method to segment play-field and its applications in match analysis in sports video. ACM Multimedia, pp.292-295, October 2004.

[8] R. M. Neal, and G. E. Hinton: A new view of the EM algorithm that justifies incremental and other variants. Learning in Graphical, Kluwer Academic Publishers, pp. 355-368, February 1993.

[9] M. Duarte, J. Inacay, and J. Zhou: Multiple Player Tracking for Sports Application Standford. IEEE Transactions on Image Processing, vol. 20, Issue 6, pp. 1652-1667, June 2011.

[10] V. A. Ngo, W. Yang, and J. Cai: Accurate Playfield Detection Using Areof-Coverage. Circuits and Systems, IEEE International Symposium, pp. 3441-3444, June 2010.

[11] F. Daniyal, M. Taj, and A. Cavallaro: Content and task-based view selection from multiple video streams. Data Semantics from Multi-media Systems, Special Issue: Semantic Multimedia, vol. 46, pp. 235-258, January 2010

[12] J. Xing, H. Ai, L. Liu, and S. Lao: Multiple player tracking in sports video: A dual-mode two-way Bayesian inference approach with progressive observation. IEEE Transactions on image Processing, Vol. 20, pp. 1652-1667, June 2011

[13] L. Wu, Y. Gong, X. Yuan, X. Zhang, and L. Cao: Semantic aware sport image resizing jointly using seam carving and warping. Multimedia Tools Application tool, pp. 1-19, April 2012.

[14] M.H. Chang, M.C. Tien, and J.L. Wu: Wild open warning for broadcast basketball video based on player trajectory. ACM International conference on multimedia, pp. 821-824, October 2009.

[15] A. Yilmaz, O. Javed, and M. Shah: Object tracking: A survey. ACM Computing Surveys, vol. 4, Issue 13, pp. 353-355, December 2006.

[16] L. Ellis, N. Dowson, J. Matas, and R. Bowden: Linear Regression and Adaptive Appearance Models for Fast Simultaneous Modeling and Tracking. International Journal of Computer Vision, vol. 95, Issue 2 , pp 154-179. November 2011.

[17] R. Kalman: A new approach to linear filtering and prediction problems. Journal of Basic Engineering, pp. 35-45. May 1960.

[18] M. Israd, and A. Blake: Condensation-conditional density propagation for tracking. International Journal of Computer Vision, vol. 1, pp. 2-28, September 1998.

[19] X. Mei, and H. Ling: Robust visual tracking and vehicle classification via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, pp. 2259-2272, June 2011.

[20] D. Comaniciu, V. Ramesh, and P. Meer: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, No. 5, pp. 564-575. May 2003.

[21] J. Ning, L. Zhang, D. Zhang, and C. Wu: Robust mean-shift tracking with corrected background-weighted histogram. IEEE Transactions on Computer Vision, vol. 1, pp. 62-69, January 2012.

[22] T. Chiang, J. Leou, and C. Lin: Mean shift algorithm based tracking system for soccer game analysis. Asia-Pacific Signal and Information Processing Association, pp 93-98, May 2009.

[23] T. Vojir and J. Matas: Robustifying the flock of trackers. 16th Computer Vision Winter Workshop, Graz, Austria, pp. 91-97, February 2011.

[24] Y. Cai, N. Freitas , and J.J. Little: Robust visual tracking for multiple targets. 9th European Conference on Computer Vision, Graz, Austria, pp. 107-118, May. 2006,

[25] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. 8th European Conference on Computer Vision, Prague, Czech Republic, vol. 2, pp. 28-49, May 2004.

[26] A. Dearden, Y. Demiris, and O. Grau: Tracking football player movement from a single moving camera using particle filters. 3rd European Conference on Visual Media Production. Part of the 2nd Multimedia Conference, pp. 29-37, January 2006.

[27] A. Lehuger, P. Lechat, and P. Perez: An adaptive mixture color model for robust visual tracking. IEEE International Conference on Image Processing, vol. 2, pp. 573 576, October 2006.

[28] G. Zhu, C. Xu, Q. Huang, W. Gao, and L. Xing: Player action recognition in broadcast tennis video with applications to semantic analysis of sports game. ACM international conference on Multimedia, New York, USA, pp. 431-440, December 2006.

[29] P. J. Figueroa, N. J. Leite, R. M.L. Barros: Tracking soccer players aiming their kinematical motion analysis. Computer Vision and Image Understanding, vol. 2, pp. 122-135, May 2006.

[30] V. Pallavi, J. Mukherjee, A.K. Majumdar, and S. Sural: Graph-based multiplayer detection and tracking in broadcast soccer videos. IEEE Transactions on Multimedia, vol. 5, pp. 794-805, August 2008.

[31] D. kaiser, M. Thaler, A. Kriechbaum, H. Fassold, W. Bailer, and J. Rosner: Real-time person tracking in high-resolution panoramic video for automated broadcast production. 8th European Conference on Visual Media Production, London, UK, pp. 21-29, September 2011.

[32] M. Beetz, S. Gedikli, J. Bandouch, B. Kirchlechner, N. V. Hoyningen-Huene, and A. Perzylo: Visually tracking football games based on TV broadcasts. International Joint Conferences on Artificial Intelligence, pp. 2066-2071. June 2007

[33] K. Fukunaga and L. Hostetler: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Transactions on Information Theory, vol. 1, pp. 32-40, January 1975.

[34] R.T. Collins: Mean-shift blob tracking through scale space. IEEE Computer Society Conference, vol. 2, pp. 234-240, June 2003.

[35] R. T. Collins and Y. Liu: On-line selection of discriminative tracking features. IEEE Computer Society In Computer Vision, pp. 346-352, May 2003.

[36] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool: SURF: Speeded-Up Robust Features. Computer Vision and Image Understanding, vol. 3, pp. 346-359, June 2008.

[37] K. Mikolajczyk and C. Schmid: Indexing based on scale invariant interest points. IEEE Computer Society in Computer Vision, vol. 1, pp. 525-531, June 2001.

[38] T. Lindeberg: Feature detection with automatic scale selection. International journal of Computer vision, vol. 2, pp. 79-116, November 1998.

[39] D. Lowe: Distinctive image features from scale-invariant key-points, cascade filtering approach. International journal of Computer vision, vol. 2, pp. 91-110, January 2004.

[40] H. Bay, B. Fasel, and L. van Gool: Interactive museum guide: Fast and robust recognition of museum objects. International Workshop on Mobile Vision, pp. 86-91, May 2006.

[41] A. Adams, N. Gelfand, J. Dolson, and M. Levoy: Gaussian KD-Trees for Fast High-Dimensional Filtering. Special Interest Group on GRAPHics and Interactive Techniques, pp. 231-238, January 2009.

[42] H. Li, C. Shen, and Q. Shi: Real-time visual tracking using compressive sensing. IEEE Computer vision in pattern recognition, pp. 1305-1312, November 2011.

[43] J. Jing, G. Xu, B. Lei, Y. He, and F. Dong: Using appearance re-matching to improve real-time compressive tracking. ACM International Conference on Internet Multimedia Computing and Service. pp. 28-33. January 2013.

[44] J. Jing, G. Xu, B. Lei, Y. He, and F. Dong: An improved real-time compressive tracking method. ACM International Conference on Internet Multimedia Computing and Service. pp. 380-385, January 2013.

[45] K. Zhang, L. Zhang, and M. Yang. Real-Time Compressive Tracking. 3th European Conference on Computer Vision. vol. 7574, pp. 864-877, May 2012.

[46] D. chlioptas: Database friendly random projections:Johnson Lindenstrauss with binary coins. JOURNAL OF COMPUTER AND SYSTEM SCIENCES. vol. 66, pp. 671-687, November 2003.

[47] B. Babenko, M.H. Yang, S. Belongie: Robust object tracking with online multiple instance learning. 33th Pattern Analysis and Machine Intelligence, pp. 1619-1632, May 2011.

[48] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma: Robust face recognition via sparse representation. 31th Pattern Analysis and Machine Intelligence, pp. 210-227, May 2009.

[49] L. Liu, P. Fieguth: Texture classification from random features. 34th Pattern Analysis and Machine Intelligence, pp. 574-586, May 2012.

[50] R. Baraniuk, M. Davenport, R. DeVore, M. Wakin: A simple proof of the restricted isometric property for random matrices. Constructive Approximation, vol. 28, Issue 3, pp. 253-263, December 2008.

[51] P. Viola, M. Jones: Rapid object detection using a boosted cascade of simple features. Computer Vision and Pattern Recognition, pp. 511-518, May 2001.

[52] S. Li, Z. Zhang: Floatboost learning and statistical face detection. 26th Pattern Analysis and Machine Intelligence, pp. 1-12, January 2004.

[53] Ng, A., Jordan, M.: On discriminative vs. generative classifier: a comparison of logistic regression and naive bayes. Neural Information Processing System Foundation, pp. 841-848, December 2002.

[54] P. Diaconis, D. Freedman: Asymptotics of graphical projection pursuit. The Annals of Statistics. vol. 12, pp. 228-235, September 1984.