MOTION COMPARISON AND TACTIC ANALYSIS IN SPORTS TRAINING

by

Hao Sun
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Computer Science

Committee:

_____________________________ Dr. Jim X. Chen, Dissertation Director

_____________________________ Dr. Yotam Gingold, Committee Member

_____________________________ Dr. Harry Wechsler, Committee Member

_____________________________ Dr. Qi Wei, Committee Member

_____________________________ Dr. Sanjeev Setia, Department Chair

_____________________________ Dr. Kenneth S. Ball, Dean, Volgenau School of Engineering

Date: ________________________ Fall Semester 2014
George Mason University
Fairfax, VA
Motion Comparison and Tactic Analysis in Sports Training

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

by

Hao Sun
Bachelor of Engineering
Beijing University of Technology, China, 2008

Director: Jim X. Chen, Professor
Department of Computer Science

Fall Semester 2014
George Mason University
Fairfax, VA
This work is licensed under a creative commons attribution-noderivs 3.0 unported license.
DEDICATION

This is dedicated to my parents Xiaoyan Sun and Hong Yu, and my girlfriend Yao Meng.
ACKNOWLEDGEMENTS

I would like to thank Dr. Jim X. Chen for his time, guidance and advice. Without his generosity, this thesis would not have been written. I would also like to thank Dr. Yotam Gingold, Dr. Harry Wechsler and Dr. Qi Wei for sharing their knowledge. Finally, I would like to thank Dr. Amihai Motro and Dr. Sanjeev Setia for their guidance in finishing this thesis.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Tables</td>
<td>vii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>viii</td>
</tr>
<tr>
<td>Abstract</td>
<td>ix</td>
</tr>
<tr>
<td>Chapter 1: Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Problems</td>
<td>3</td>
</tr>
<tr>
<td>1.1.1 Individual mechanics training</td>
<td>4</td>
</tr>
<tr>
<td>1.1.2 Team tactics training</td>
<td>4</td>
</tr>
<tr>
<td>1.2 Contributions</td>
<td>5</td>
</tr>
<tr>
<td>1.2.1 A new virtual racket sports training system and SWING distance</td>
<td>5</td>
</tr>
<tr>
<td>1.2.2 A new sports video segmentation method and evaluation metric</td>
<td>6</td>
</tr>
<tr>
<td>1.2.3 A new tactics analysis system</td>
<td>7</td>
</tr>
<tr>
<td>1.3 Organization</td>
<td>8</td>
</tr>
<tr>
<td>Chapter 2: Individual mechanics training</td>
<td>9</td>
</tr>
<tr>
<td>2.1 Related work</td>
<td>9</td>
</tr>
<tr>
<td>2.1.1 Motion capture systems</td>
<td>9</td>
</tr>
<tr>
<td>2.1.2 Virtual motion training system</td>
<td>15</td>
</tr>
<tr>
<td>2.2 The research sports activity</td>
<td>16</td>
</tr>
<tr>
<td>2.3 The approach</td>
<td>17</td>
</tr>
<tr>
<td>2.3.1 Capturing phase</td>
<td>18</td>
</tr>
<tr>
<td>2.3.2 Comparison phase</td>
<td>24</td>
</tr>
<tr>
<td>2.4 Experiment results</td>
<td>31</td>
</tr>
<tr>
<td>2.4.1 Capturing and comparison results</td>
<td>31</td>
</tr>
<tr>
<td>2.4.2 Generate training advices</td>
<td>35</td>
</tr>
<tr>
<td>2.4.3 Training case study</td>
<td>37</td>
</tr>
<tr>
<td>Chapter 3: Broadcast sports video segmentation</td>
<td>40</td>
</tr>
<tr>
<td>3.1 Related work</td>
<td>40</td>
</tr>
</tbody>
</table>
3.2 The approach .............................................................................................................. 46
3.2.1 CRP: the Chinese Restaurant Process ................................................................. 51
3.2.2 S-CRP: a new segmentation method ................................................................. 54
3.2.3 Levenshtein distance Ratio ................................................................................ 58
3.3 Experiment results .................................................................................................. 60

Chapter 4: Team Tactics training .................................................................................. 65
4.1 Related work ............................................................................................................. 65
4.2 Players and ball tracking ......................................................................................... 68
4.3 Camera calibration .................................................................................................... 73
4.4 Optimal defense tactic ............................................................................................. 75
  4.4.1 Player attribute vector ......................................................................................... 75
  4.4.2 Player offense threat ............................................................................................ 77
  4.4.3 Player defense ability .......................................................................................... 78
  4.4.4 Optimal defensive tactic ....................................................................................... 79
4.5 Experiment result ...................................................................................................... 83
4.6 Conclusion ................................................................................................................ 89

Chapter 5: Future Work .................................................................................................. 90
Chapter 6: Conclusion ..................................................................................................... 92
Bibliography .................................................................................................................... 96
LIST OF TABLES

Table                                                                 Page
Table 1: Pseudo-code of the DVT distance ................................................................. 28
Table 2: SWING distance timeline ................................................................................. 38
Table 3: Notations of frames and shots........................................................................ 45
Table 4: L-Ratio pseudo-code ...................................................................................... 60
Table 5: Segmentation result strings ............................................................................ 63
Table 6: The frame classification accuracy and L-ratio ................................................. 63
Table 7: Court positions of the 6 reference points (in meters) ..................................... 75
Table 8: Offensive and defensive attributes correspondences ..................................... 76
Table 9: The effective range (distance to the hoop in meters) ...................................... 77
Table 10: Role and number correspondence .................................................................. 79
Table 11: Players’ attributes ......................................................................................... 85
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1: Checker board pattern</td>
<td>20</td>
</tr>
<tr>
<td>Figure 2: Marker placement</td>
<td>21</td>
</tr>
<tr>
<td>Figure 3: Racket before and after background subtraction</td>
<td>22</td>
</tr>
<tr>
<td>Figure 4: Marker tracking results</td>
<td>24</td>
</tr>
<tr>
<td>Figure 5: An example of Levenshtein distance</td>
<td>25</td>
</tr>
<tr>
<td>Figure 6: The DVT distance</td>
<td>27</td>
</tr>
<tr>
<td>Figure 7: The “match – delete – match” patterns</td>
<td>29</td>
</tr>
<tr>
<td>Figure 8: The “match – insert – match” patterns</td>
<td>29</td>
</tr>
<tr>
<td>Figure 9: Locate the turning points</td>
<td>30</td>
</tr>
<tr>
<td>Figure 10: Forehand drive trajectory and paddle orientation</td>
<td>32</td>
</tr>
<tr>
<td>Figure 11: Two drive motions with different swing velocity and paddle orientation</td>
<td>33</td>
</tr>
<tr>
<td>Figure 12: The two trajectories from a different view point</td>
<td>33</td>
</tr>
<tr>
<td>Figure 13: The visualized DVT distance</td>
<td>34</td>
</tr>
<tr>
<td>Figure 14: Training advices generation</td>
<td>36</td>
</tr>
<tr>
<td>Figure 15: Visualization of 13 weeks training</td>
<td>38</td>
</tr>
<tr>
<td>Figure 16: Temporal models</td>
<td>44</td>
</tr>
<tr>
<td>Figure 17: DCRP table assignment</td>
<td>52</td>
</tr>
<tr>
<td>Figure 18: False detected shots</td>
<td>55</td>
</tr>
<tr>
<td>Figure 19: Miss detected shots</td>
<td>55</td>
</tr>
<tr>
<td>Figure 20: Bridge over the noisy gap</td>
<td>56</td>
</tr>
<tr>
<td>Figure 21: Frames connect to their own class side</td>
<td>56</td>
</tr>
<tr>
<td>Figure 22: Connections cross over another class</td>
<td>56</td>
</tr>
<tr>
<td>Figure 23: Examples of the four semantic classes</td>
<td>62</td>
</tr>
<tr>
<td>Figure 24: Player tracking</td>
<td>72</td>
</tr>
<tr>
<td>Figure 25: Ball tracking</td>
<td>73</td>
</tr>
<tr>
<td>Figure 26: Court coordinate system and 6 reference points (a-f)</td>
<td>74</td>
</tr>
<tr>
<td>Figure 27: Reference point tracking</td>
<td>75</td>
</tr>
<tr>
<td>Figure 28: Examples of defense distance and angle</td>
<td>79</td>
</tr>
<tr>
<td>Figure 29: A defense assignment</td>
<td>80</td>
</tr>
<tr>
<td>Figure 30: The screen action between SAS Duncan and Parker</td>
<td>82</td>
</tr>
<tr>
<td>Figure 31: The defense tactic of MIA (first footage)</td>
<td>82</td>
</tr>
<tr>
<td>Figure 32: Open shot opportunity for SAS Leonard</td>
<td>84</td>
</tr>
<tr>
<td>Figure 33: The optimal defense tactic (first footage)</td>
<td>86</td>
</tr>
<tr>
<td>Figure 34: The optimal defense tactic (second footage)</td>
<td>88</td>
</tr>
<tr>
<td>Figure 35: The defense tactics of MIA (second footage)</td>
<td>88</td>
</tr>
</tbody>
</table>
ABSTRACT

MOTION COMPARISON AND TACTIC ANALYSIS IN SPORTS TRAINING
Hao Sun, Ph.D.
George Mason University, 2014
Dissertation Director: Dr. Jim X. Chen

There are two important subjects in competitive sports training, namely the athlete (student athlete)’s individual mechanics training and the team’s tactics training.

For the student athlete’s individual mechanics training, many researchers seek to capture and visualize the student’s motions in 3D virtual environments. Despite most virtual training systems are able to visualize students’ motions from various angles, those systems leave the students themselves to figure out how to revise their motions to improve performance. We present a training system that is able to compare the coach’s and the student’s motions and quantify their distances. In addition, based on the motion comparison, we are able to automatically generate training advice to tell students where and how to improve.

Besides individual mechanics training, tactics training is an important training aspect for team sports. Tactics training studies the current state of the game from existing video footage, to determine a good tactic move such as where to attack or which player to
Among all types of video footage, broadcast matching replays are more widely studied because they are easily accessible and unlikely to have some tactical reservations. Because broadcast sports videos consist of both tactic relevant shots and irrelevant shots, many efforts have been made to automatically segment videos to separate these types of shots. Some methods use domain knowledge of the target sports activity, and are not able to be applied to other sports activities. Other systems use supervised learning approach to improve frame classification and shot boundary detection accuracy, but without focus on maintaining the integrity of the tactic relevant segments and the video structure. We introduce a novel method, named Segmentation based on distance dependent Chinese Restaurant Process (S-CRP), to segment broadcast sports videos into high quality semantic shots without the use of domain knowledge and more sophisticated classifiers. In addition, we also introduced a new performance metric, namely Levenshtein distance Ratio, to provide a more accurate measure of how well the segmentation result maintains the structure of the original video.

After sports videos segmentation, researchers have made progresses in further tactic analysis by automatically detecting and tracking players’ positions or ball movements in order to capture interesting player behaviors or important events in the game. However, current tactic analysis systems provide only low level assistances in tactics training. They are able to capture certain events in the game and help the team with statistical analysis, but they provide little help towards finding a better tactic. We introduce a novel tactics training system that is able to consider players’ attributes together with their positions to accurately estimate each player’s offense threat or defense
ability. Our system formulates an optimization problem to find the optimal defense tactic that minimizes the offense team’s threat.
CHAPTER 1: INTRODUCTION

Competitive sports activities are one of the crucial components of physical sports activities. The goal of competitive sports is to improve human physical ability, skills, and most importantly, to achieve victory. There are a lot of motivations for the development of competitive sports including biological, individual psychology, and social motivations, which originate from the human desire of faster, higher and stronger.

Competitive sports training is the organized activity that aim to improve the athletes’ performance in competitive sports games. It is now become a systematic subject, which includes sports psychology, nutriology, anatomy, physiology, and mechanics training, etc. Among all the different training components, the athlete’s mechanics training is important as it is directly related to the performance in competition. The most common process of mechanics training involves an athlete student (or simply student) and an expert (or simply coach) working together. Of course, a coach can work with a group of students, but we are focusing on one student. The coach often needs to verbally explain the essentials of a certain mechanic (skill), and at the mean time, repeatedly demonstrate the mechanic to the student. In order for the training to be effective, the coach has to do laborious work. Despite the coach’s effort, the student also needs to be able to understand the explanations and demonstrations from the coach. This requires the coach to have good coaching skills. The coaching philosophy needs to be balanced
between the command style and submissive style, and it is difficult to switch between styles depending on the situation and the student involved [1].

Alternatively, a student can train himself or herself by watching existing video footage or tournament replays. By mimicking the techniques or motions of a professional player, the student can revise and improve his/her own skills. This, however, requires the student to be quite familiar with this sports activity and has very sharp observation and mimicry skills in order to understand professional plays and improve personal style. Because of the limitations in a traditional training process, a successful training often takes months or years.

With the development of computer graphics and computer vision, many researchers seek to assist sports mechanics training by capturing students’ motions and visualizing them in 3D space. Such motion capture systems can be divided into magnetic, mechanic, and optical systems, etc. The student is able to visualize his/her motion from different viewpoint in the 3D environment, and researchers hope the motion capture and visualization can help the student to understand and learn the target mechanics better.

Besides individual mechanics training, for competitive team sports activities such as basketball or soccer, tactics training is another important training aspect. Tactics training studies the current state of the game (including players’ positions, ball procession, etc), to determine a good tactic move such as where to attack or which player to defend. Traditionally tactics training requires the team to watch past broadcast matching replays or public training footage. Compare to public training footage,
broadcast matching replays are more widely studied because they are easily accessible and unlikely to have tactical reservations.

However, broadcast sports videos are designated for the best viewing experience of the TV audiences, and they are commonly synthesized by several camera views which consist of both tactic relevant shots and irrelevant shots. For example, in a basketball game, a tactic relevant shot has to be a long view shot that captures most of the players’ positions and the ball. Medium view shots or player close-up view shots will not contain any tactical information.

For analysis and understanding, many efforts have been made to automatically segment broadcast sports videos into different types of shots. Some of the methods use domain knowledge of the target sports activity while others define some semantic classes and segment based on the supervised machine learning approach.

After sports videos segmentation, tactic analysis can be performed on those tactic relevant shots. Researchers have made progress in tactic analysis by automatically detect and track players’ positions or ball movements in order to capture interesting player behaviors or important events in the game.

1.1 Problems

In this thesis, we focus on the two important tasks of competitive sports training, namely (1) individual mechanics training for an athlete, and (2) tactics training for a team. For the rest of this section, we will explain the problems of current training methods/systems and introduce our approach.
1.1.1 Individual mechanics training

As briefly mentioned above, in mechanics training, traditional training methods (learn from coach or video footage) are highly depending on the coach’s coaching skills and the student’s observation (or mimic) abilities. A successful training often takes months or years. Some systems use the motion capture techniques to capture and visualize student motions in 3D virtual environment. These systems are certainly helpful since they can capture and visualize the student’s motion in different viewpoints. For a virtual motion training system, it is essential to capture the student’s and the coach’s motions to show “what is the student’s current motion” and “what is the coach’s standard motion”. On the other hand, it is very important to have some meaningful comparisons between the two motions and answer the question of “how to revise from the current motion to the standard motion”. The current virtual motion training systems capture subjects’ motions and implement a variety of ways for visualization but leave the “How to revise” question to the student. It is desirable to develop a more effective mechanics training system that is able to compare the coach’s and the student’s motions and quantify their distances. In addition, based on the comparison, the desired training system should be able to generate training advices and tell the student how to revise his/her motions to improve performance.

1.1.2 Team tactics training

Before tactic analysis, broadcast sports videos need to be segmented into semantic shots. Tactic relevant shots will be the input of the tactics training system. Some of the segmentation methods use domain knowledge of the target sports activity, and are not
able to be applied to other sports activities. On the other hand, some other systems use supervised learning approach, and can be used in other sports videos. However, current segmentation methods only focusing on improve frame classification accuracy, shot boundary detection standard recall or precision, and a small number of noisy frames in the video can easily break the original tactic segment without much accuracy loss. If the segmentation result is not able to maintain the integrity of the tactics relevant segments and the video structure, further tactics analysis is meaningless.

After sports video segmentation, the current works in tactic analysis also have their limitations. Current tactic analysis systems focus on automatically detect and track players’ positions or ball movements in order to detect important events in the game. Despite their effectiveness in event detection, the related systems provide only low level assistances in tactics training. They are able to capture certain events in the game and help the team with statistical analysis, but they provide little help towards finding a better tactic.

1.2 Contributions

1.2.1 A new virtual racket sports training system and SWING distance

We developed a virtual racket sports training system for improving a student’s individual mechanics. Unlike other existing virtual training systems that only capture and visualize subjects’ motions in a virtual environment to assist training, our system further quantifies the distance between the student’s and the coach’s motions, and generate training advices.
The training target activity we consider is the forehand drive mechanics training in table tennis. Our system quantitatively compares the student’s and the coach’s forehand drive trajectory and paddle impact orientation. Based on the comparison results, training advices are provided to the student, so s/he can simulate the coach’s standard motion and improve performance. We use optical multi-view marker based approach in the capturing phase with markers only attached to the paddle, allowing motion freedom while preserving accuracy. In the comparison phase, a novel measurement – SWING distance – has been introduced based on Levenshtein distance to quantify the distance between two forehand drive motions (velocity trajectory and paddle orientation).

Based on the experiment from professional athletes and students, a student can perform the forehand drive with a standard velocity trajectory in much shorter time (10-13 weeks) than traditional methods (at least 20 weeks).

1.2.2 A new sports video segmentation method and evaluation metric

We introduce a novel method, named S-CRP (Segmentation based on distance dependent Chinese Restaurant Process), to segment broadcast sports videos into semantic shots and maintain the integrity of the tactics relevant segments.

S-CRP employs distance dependent Chinese Restaurant Process (DCRP) using two segmentation criteria, namely appearance and time distances. It takes advantage of the customer (frame) assignments in DCRP and is able to reduce the negative effect of noisy frames without the use of domain knowledge and more sophisticated classifiers.
In addition, we introduced a new performance metric, namely Levenshtein distance Ratio, which gives a more accurate measure of how well the segmentation result can maintain the integrity of original segments and match the video structure.

1.2.3 A new tactics analysis system

We introduce a novel tactics training system that is able to assess the state of the basketball game in detail, and find better defense tactics.

Besides only analyzing player/ball positions, our system studies players’ attributes together with their positions to accurately estimate each player’s offense threat or defense ability for a certain applied defense tactic. We then formulize an optimization problem to find the defense tactic that minimizes the offense team’s threat. The optimal defense tactic will be suggested to assist tactics training.

In the evaluation of our system, we first focus on situations where the defense team made some bad tactic choices that cause the offense team to score easily. We demonstrate the optimal defense tactic found by our system to help training the defense team to make some better defense decisions and avoid giving such easy opportunities to the enemy team in the future. We also demonstrate the optimal tactic found is indeed helpful by analyzing situations where the defense team’s tactic in the video actually matches the optimal tactic found by our system and results in a successful defense. The optimal tactics generated by our system are commented and evaluated by three basketball coaches from Beijing Sports University.
1.3 Organization

The rest of this dissertation is organized as follows. In Chapter 2, we design and implement a virtual training system for the athlete’s individual mechanics training. The virtual training system is able to capture, visualize and compare motions from the coach and the student athlete. Training advices are generated based on the motion comparison. In Chapter 3, we focus on team sports tactics training using broadcast video replays. We first introduce our method of S-CRP for video segmentation. In addition, we introduce the Levenshtein distance Ratio performance metric, which gives a more accurate measure of the segmentation structural similarity. In Chapter 4, we design a novel tactics training system that is able to find the optimal defense tactics which minimizes the offense team’s threat in a basketball game. We outline possible directions for our future work in Chapter 5 and conclude this dissertation in Chapter 6.
CHAPTER 2: INDIVIDUAL MECHANICS TRAINING

In this chapter, we focus on the first sports training task: individual mechanics training of the student athlete. We first discuss the related work on motion capture technique and virtual training systems. Secondly, we introduce our new virtual racket sports training system and the SWING distance, which is able to quantify the distance between the student’s and the coach’s motions. From training case study, we show our new virtual training system is able to provide training advices that significantly reduce the training time needed.

2.1 Related work

2.1.1 Motion capture systems

Motion capture is the process of recording human movement for visualization and analysis. The idea came from rotoscoping, which is a method to copy human poses for animated characters in movie making. The computer uses these poses as keys for generating a smooth animation. Motion capture has been used as a photogrammetric analysis tool in biomechanics research in the 1970s, and expanded into different areas since 1980s. With the development of motion capture systems, the information captured can be as general as the simple position in space or as complex as the deformations of the target.
Motion capture can be used in entertainment, sports, and medical applications [2,3]. In entertainment activities such as film making and video game development, motion capture can be used to recording actions of human actors, and animate virtual character in 2D or 3D computer animation. When face expressions and finger movements are included, it is often referred to as performance capture. In motion capture sessions, movements of the target are sampled many times per second.

The purpose of motion capture is to record only the movements of the target, not its visual appearance, therefore, researchers have been exploring different directions/methods of building their motion capture systems. Some of the major methods of building motion capture systems include magnetic, mechanic, inertial, and optical methods [4].

Magnetic systems use electromagnetic sensors connected to a computer to produce 3D data in real-time with low processing costs. They calculate position and orientation by the relative magnetic flux of three orthogonal coils on both the transmitter and each receiver. The relative intensity of the voltage or current of the three coils allows these systems to calculate both range and orientation by mapping the tracking volume. The information captured by the sensor is in 6 degree of freedom, and markers are not occluded by nonmetallic objects but are susceptible to magnetic and electrical interference from metal objects in the environment. Interference such as rebar or wiring, which affect the magnetic field, and electrical sources such as monitors, lights, cables and computers can all affect the capture process. The sensor response is nonlinear, especially toward edges of the capture area. The wiring from the sensors tends to preclude extreme
performance movements. The capture volumes for magnetic systems are dramatically smaller than motion capture systems with other methods. In general, a magnetic system restricts movement of the target due to the size of magnetic field and cabling.

Mechanical motion capture systems can directly track target joint angles due to the placement of the mechanical sensors. Target actor attaches the skeletal like structure to their body and as they move together with the articulated mechanical parts, measuring the relative motion. Mechanical motion capture systems can achieve real time capture, and are able to handle occlusions. Compare to magnetic systems, mechanical systems have unlimited capture volume. However, the mechanical sensors have to be attached to a special suit which consists of rigid structures of jointed, straight metal or plastic rods linked together with potentiometers that articulate at the joints of the target. In short, mechanical systems use special suits with integrated mechanical sensors that register the motion of articulation. While mechanical systems are able to capture target motions in real time, the movements from the target are limited due to wearing special mechanical sensors.

Inertial motion capture system is based on miniature inertial sensors, biomechanical models and sensor fusion algorithms [5]. The captured motion information of the inertial sensors is often transmitted wirelessly to a computer, where the motion is recorded or viewed. Most inertial systems use gyroscopes to measure rotational rates. These rotations rates are translated to the virtual target in the software. No external cameras, emitters or markers are needed for relative motions, although they are required to give the absolute position of the user if desired. Inertial motion capture systems
capture the 6 degrees of freedom motion data of a target in real-time and can give limited
direction information if they include a magnetic bearing sensor, although these are much
lower resolution and susceptible to electromagnetic noise. Inertial system has great
portability and large capture areas. These systems are similar to the Wii controllers but
are more sensitive and have greater resolution and update rates. They can accurately
measure the direction to the ground to within a degree. However, the captured motions
have lower positional accuracy and positional drift which can compound over time compare to other systems. The popularity of inertial systems is rising amongst
independent game developers, mainly because of the quick and easy set up resulting in a
fast pipeline. In short, although inertial motion capture systems have the advantages of
large capture area and easily set up, they have lower positional accuracy compare to other
methods.

Optical systems are based on photogrammetric methods. Optical systems provide
high accuracy/freedom of movement, and the possibility of interaction between different
actors [6]. Optical motion capture systems can be divided into marker based or marker-
less systems. In marker based motion capture systems, multiple cameras are used to
capture marker positions placed on a performer’s body (e.g. joints). If a marker can be
observed in at least two camera views, the marker positions those views will then be
triangulated to 3D world coordinate system. For example, in the Vicon system, the
performer wears a special suit attached with retro-reflective markers which are captured
by a set of synchronized cameras. Because the retro-reflective nature, markers are easily
detectable in each camera view as they have extremely higher intensity compare to the
performer’s body suit. Later, the detected marker positions in each camera view will be triangulated to 3D coordinate system. Based on the obtained 3D marker positions, rotational data of each joint of the performer is calculated and used for animation rendering.

The downside of such mocap system is that the performer may have some movement restrictions since s/he wears markers near each joint location to help identify the positions or angles of joints.

Besides the widely used retro-reflective markers in marker based systems, acoustic, inertial, LED, magnetic or other type of reflective markers, or combinations of any of these are found in other marker based optical motion capture systems. Some systems use LED markers that can send out signal at different frequencies. Those markers are called “active markers” since their marker identities can be easily distinguished based on their frequencies known in advance. Those types of markers require much less computation in marker tracking, but they are more expensive and sometimes suffer from noisy capture environments.

In real-time optical motion capture systems, markers are tracked optimally at least two times as the frequency rate of the desired motion. Besides the temporal resolution, the spatial resolution of the systems is also important, since low resolution marker positions are inaccurate and causes the same problems as motion blur.

As opposed to marker based mocap systems, markerless systems will not require performers to attach markers on body, which allow more moving freedom. Those systems are based on techniques and research in computer vision, and special vision algorithms
are designed to allow the systems to analyze multiple streams of optical input and identify the performer’s forms, breaking them down into constituent parts for tracking. Some markerless systems use the optical flow of all pixels over all the 2-D planes of the cameras for gesture and facial expression capture.

The Kinect is one of the widely used markerless systems in the gaming or robotics vision research field. Different from marker based systems, the Kinect consists of an infrared laser emitter, an infrared camera and an RGB camera. The laser source emits a single beam which splits into multiple beams by a diffraction grating to create a constant pattern of speckles projected onto the scene. This pattern is captured by the infrared camera and is correlated against a reference pattern. When a speckle is projected on an object whose distance to the sensor is smaller or larger than that of the reference plane, the position of the speckle in the infrared image will be shifted in the direction of the baseline between the laser projector and the perspective center of the infrared camera. These shifts are measured for all speckles by a simple image correlation procedure, which yields a disparity image. For each pixel the distance to the sensor can then be retrieved from the corresponding disparity [7].

However, despite more moving freedom, marker-less systems are more easily affected by noise or shooting distances and will not be as accurate as marker based systems in terms of getting the 3D position of interested points. It has been shown as a comparison in report [7,8], the Vicon system obtains more accurate marker positions than the Kinect system, especially under conditions of lower resolution and longer distance. Some other markerless systems are vision based and use silhouettes or other image
features to locate and extract the performer and compare with stored templates for recognition [9]. They are not focused on accurately extracting points of interest and are mostly used in other purposes such as identity/motion recognition or surveillance.

2.1.2 Virtual motion training system

For a virtual motion training system, it is essential to capture the student’s and the coach’s motions to show “what is the current motion” and “what is the standard motion”. On the other hand, it is very important to have some meaningful comparisons between the two motions and answer the question of “how to revise the current motion to the standard motion”. A lot of virtual motion training systems [10-21] capture subjects’ motions and implement a variety of ways for visualization but leave the “How to revise” question to the student.

For example, P.T. Chua et al. [19] built a wireless virtual Tai Chi training system which is able to capture the student’s and the coach’s motions with a large capture volume but the student has to figure out how to improve his/her motion based on the visualized result. Although several immersive techniques (such as providing multiple copies of a teacher’s body positioned around the student and allowing the student to superimpose his body directly over the virtual teacher) have been tested in their visualization, it is reported in the paper that those techniques cannot provide more effective training than simple visualizations. Similarly, U. Yang et al. [20] built the virtual training environment “Just Follow Me” using an immersive technique called Ghost metaphor. The student observes the coach’s visualized motion and try to “follow” the virtual master as close as (or as fast as) possible in real time.
Follow the coach’s motion will help the training to some extent, but it is likely to cause the student to stubbornly imitate the coach and lose his/her own feeling toward the motion. Moreover, when students reach their plateau, it will be hard for them to find problems themselves, and they need training advices to make further improvement.

Some training systems attempted to compare the standard (desired) motion with the current motion, but they usually lack a thorough comparison of full motion trajectories and provide no training advice on how to improve the current motion. For example, T. Tsuji et al. [22] built a virtual tennis swing training system that compares some hand parameters (hand position, hand velocity etc.) when the racket hits the ball. They separate the swing results into “hit the ball” and “miss the ball” so that the student is able to get an idea of the desired hand position (velocity) when the racket impacts the ball. But because the tennis racket swing motion has a continuous trajectory, without a proper guidance on the full swing motion trajectory, the student will have difficulties to figure out how to revise the motion before or after hitting the ball and the training is likely to result in a stiff and low quality swing motion.

2.2 The research sports activity

The research sports activity we focus on is table tennis. Table tennis is one of the most popular racket sports in the world [1]. The most common way of training such a sports activity involves an athlete student (or simply student) and an expert (or simply coach) working together. Of course, a coach can work with a group of students, but we
are focusing on one student. For example, in teaching table tennis techniques such as driving or slicing, the coach needs to repetitively explain the motion essentials and demonstrate the technique to the student. Alternatively, a student can train himself or herself by watching existing video footage or tournament replays. By mimicking the techniques or motions of a professional player, the student can revise and improve his/her own skills. This, however, requires the student to be quite familiar with this sports activity and has very sharp observation and mimicry skills in order to understand professional plays and improve personal style.

Among all the table tennis techniques, forehand drive is the backbone [23]. It is the strongest shot in the game. If a student can perform a successful forehand drive, her/his confidence and understanding of the sports will be significantly improved, and s/he will be more likely to succeed in a competition. A good forehand drive involves a velocity trajectory and the accurate timing of strength-burst with the perfect impact orientation.

It is not difficult for the student to imitate the driving trajectory, but how to make the drive motion standard is the major concern during the training. This requires the training system to be able to analyze the student’s forehand drive trajectory and paddle orientation and find and quantify the difference between the student’s and the coach’s forehand drive motions.

2.3 The approach
Our training system will focus on the capture, extraction, and comparison of the student’s and the coach’s forehand drives and provide training advices to the student based on the comparison result. We use optical multi-view marker based approach in the capturing phase, with markers only attached to the paddle (rather than the player’s body), allowing complete motion freedom while preserving accuracy. In the comparison phase, a novel measurement – SWING distance – has been introduced based on Levenshtein distance to compare and quantify the difference between the student’s and the coach’s motions.

2.3.1 Capturing phase

In this section, we introduce the design and implementation of our capturing system, including camera calibration, marker placement, marker detection, triangulation and tracking.

2.3.1.1 Calibration

Camera calibration is a necessary step in 3D computer vision in order to extract metric information from 2D images. Camera calibration techniques can be classified roughly into two categories: photogrammetric calibration and self-calibration [24].

Photogrammetric calibration techniques will observe an object whose geometry structure is well known in advance. Those techniques usually observe objects that are consist of planes orthogonal to each other, and the calibration result can be very reliable [25,26]. However, these techniques are expensive in apparatus and setup.

On the other hand, self-calibration techniques do not use any calibration object. A captured scene (or image) will generally provide two constraints on the cameras’ internal
parameters [27,28]. If images are taken by the camera with fixed internal parameters, three images are sufficient to recover both the camera’s internal and external parameters [29,30]. Although self-calibration is very flexible, it does not always obtain reliable results [31].

The camera calibration technique used by our systems is from the work of Zhang et al. [24], where only requires the camera to observe a planar pattern shown at least two different orientations. The pattern can be printed on a laser printer and attached to a planar surface. Either the camera or the planar pattern can be moved by hand and the motion need not be known. The technique lies between the photogrammetric calibration and self-calibration, because it uses 2D metric information rather than 3D or purely implicit one. It gains more flexibility when compared with classical techniques and considerable degree of robustness when compared with self-calibration techniques. It has been reported to have very good performance on both computer simulation and real data.

We will briefly describe the process camera calibration below.

In order to accurately detect and recover marker coordinates from multiple views, camera calibration is required before capture. The projection matrix $P_i$ for each camera $i$ needs to be calculated based on its intrinsic and extrinsic parameters.
Figure 1: Checker board pattern

\[ P_{i3x4} = \begin{pmatrix} K_i & 0 \\ R_i & T_i \end{pmatrix} \] (2.1)

\( K_{i3x3} \) is the camera intrinsic parameters, and \( R_{i3x3} \) and \( T_{i3x1} \) are the camera rotation and translation extrinsic parameters estimated using a checker board (Figure 1).

2.3.1.2 Marker placement and detection

The markers used in our system are retro-reflective markers. A retro-reflective marker is a retroreflector (sometimes called a cataphote) that reflects light back to its source with a minimum of scattering. An electromagnetic wave front is reflected back along a vector that is parallel to but opposite in direction from the wave's source. The
angle of incidence at which the device or surface reflects light in this way is greater than zero, unlike a planar mirror, which does this only if the mirror is exactly perpendicular to the wave front, having a zero angle of incidence.

By using retro-reflective markers, each camera view is able to capture the marker with high intensity with a minimum of light attenuation. Markers are attached to the table tennis paddle, where the paddle orientation can be uniquely defined by placing three markers as shown in Figure 2.

For each camera view, markers are detected by performing background subtraction and image binarization with an intensity threshold (Figure 3). Contour of each marker is then extracted and represented by a set of vertices and the 2D marker coordinates in each camera view can be estimated as the geometric center of contour vertices due to its convexity [32].

![Figure 2: Marker placement](image-url)
2.3.1.3 Marker triangulation

After marker detections in 2D images, triangulation can be performed to recover markers in 3D space. Let $U_i$ and $U_j$ be the detections of marker $X$ from camera $i$ and $j$. The relation of $X$ and its 2D detections $U_i$ and $U_j$ can be described by the following equation set:

\[
\begin{align*}
U_i &= P_i X \\
U_j &= P_j X
\end{align*}
\]  

(2.2)

where $P_i$ and $P_j$ are the projection matrices for camera $i$ and $j$. We expand 2D detection $U$ to homogeneous coordinates as $U = (u_1, u_2, 1)^\top$, and represent projection matrix $P$ by row vectors as $P = (p_1, p_2, p_3)^\top$. The equation set (2.2) can be rewritten in the form of $AX = 0$ as shown in (2.3), where $X$ can be resolved using SVD decomposition.
2.3.1.4 Marker tracking

Given marker positions in some initial frames, marker tracking is to automatically estimate its positions in subsequent frames. Our system uses a 3D tracking approach where we estimate the next marker position in 3D space based on a piecewise cubic curve fitted by previous marker positions. The estimated 3D position is then projected back to each 2D camera view, where the corresponding 2D marker detections are used to correct the estimation.

More specifically, in order to determine the 3D position of marker \( X \) in frame \( k \) (denoted by \( X_k \)), an estimation \( X'_k \) is first calculated based on \( X_{k-1}, X_{k-2} \ldots X_{k-m} \) of previous frames. Because of the smoothness nature of table tennis forehand drive trajectory, we choose \( m = 4 \) and fit \( X_{k-1}, X_{k-2}, X_{k-3}, X_{k-4} \) with a piecewise cubic curve to estimate \( X'_k \). \( X'_k \) is then projected back to each camera view and corrected by the marker detections that are closest to the projections. Our system will remain automatic if \( X \) can be observed in at least two camera views. The marker tracking result is presented in Figure 4.
Once all marker positions have been confirmed, the paddle orientation will be uniquely defined. The rotational quaternion is calculated from the paddle coordinates system to the standard world coordinates system and converted to Euler angle representation (yaw, pitch and roll) for the comparison phase.

2.3.2 Comparison phase

A table-tennis forehand drive of $n$ frames can be defined by the following sequence $Y_1Y_2Y_3...Y_n$, where $Y_i = (x, y, z, \text{yaw}, \text{pitch}, \text{roll})$ is the paddle state at frame $i$ with $(x, y, z)$ representing the paddle center position and $(\text{yaw}, \text{pitch}, \text{roll})$ representing
the paddle orientation. We use two sequences, namely $S_c$ and $S_s$, to describe the coach’s forehand drive and the student’s forehand drive respectively. Note that $S_c$ and $S_s$ can be different in length.

\[
S_c : Y_1 Y_2 Y_3 \ldots Y_n \\
S_s : Z_1 Z_2 Z_3 \ldots Z_m
\]  

We name the distance between $S_c$ and $S_s$ as SWING distance and our goal is to quantify the distance. As we mentioned in previous sections, a standard forehand drive is characterized by its velocity trajectory and paddle impact orientation. Therefore, the comparison of $S_c$ and $S_s$ can be divided into two parts: (a) the velocity trajectory comparison and (b) the impact orientation comparison. We define two measurements, namely Distance of Velocity Trajectory (DVT) and Distance of Impact Orientation (DIO), as two components of the SWING distance. After the comparison phase, both the coach’s and the student’s forehand drive paddle trajectories and orientations, along with their SWING distance are animated in the rendering stage to provide training advices on how to improve the student’s performance. The DVT and DIO are described in detail below.

$kitten \rightarrow sitten$ (substitute 's' for 'k')

$sitten \rightarrow sittin$ (substitute 'i' for 'e')

$sittin \rightarrow sitting$ (insert 'g' at the end)

Figure 5: An example of Levenshtein distance
2.3.2.1 Distance of Velocity Trajectory (DVT)

Since the velocity trajectory is only related to paddle center positions, we left aside paddle orientations in $S_c$ and $S_s$ and define $T_c: A_1A_2A_3...A_n$ and $T_s: B_1B_2B_3...B_m$ as the coach’s and the student’s paddle trajectory sequences respectively. Vectors $A_i$ and $B_i$ are paddle center positions at frame $i$ (i.e. $A_i = (x, y, z)$). The goal of DVT is to compare and quantify the distances of sequence $T_c$ and $T_s$. The comparison can be difficult because the frame correspondence of the two sequences is not known in advance, and they usually do not have equal length [33].

Levenshtein distance provides a way of comparing two strings, which is to find the minimum distance based on the number of character substitution, deletion and insertion operations [34]. For example, the Levenshtein distance between string “kitten” and “sitting” is 3 (2 substitutions and 1 insertion in Figure 5).

Similarly, if we consider paddle positions in sequence $T_c$ and $T_s$ as characters in strings, the number of substitution, deletion and insertion operations can be used to measure the distance between two paddle trajectories. We use insertion and deletion operations to mark the coach’s and the student’s swing velocity difference (as it can be reflected by the length difference of the trajectories), and substitution operations to mark the paddle position difference.

For two paddle positions $A_i$ and $B_j$, we consider $A_i = B_j$ ($A_i$ and $B_j$ are matching) if their distance $d$ is smaller than a matching threshold $\theta$ (i.e. $d < \theta$). If $A_i \neq B_j$ (i.e. $d \geq \theta$), we should consider a substitution from $A_i$ to $B_j$. However, the substitution in Levenshtein distance does not take account the relative distance between two characters. For example,
two substitutions $a \rightarrow b$ and $a \rightarrow z$ will have the same cost regardless of ‘a’ and ‘b’ are close while ‘a’ and ‘z’ are far away alphabetically.

In the coach’s and students’ paddle trajectories comparison, we need to distinguish one student’s trajectory that only need “close substitutions” and another need “far away substitutions”. Therefore, we introduce a substitution threshold $\sigma$, and our DVT distance will allow substitution only if $A_i$ and $B_j$ are closer than $\sigma$ (i.e. $\theta \leq d < \sigma$). If $A_i$ and $B_j$ are far away (i.e. $d \geq \sigma$), the original substitution operation have to be achieved by a deletion followed by an insertion, which doubles the number of operations needed. Table 1 demonstrates the Pseudo-code of the DVT distance formally.

Figure 6 demonstrates the DVT distance visually. The coach’s and the student’s paddle velocity trajectories ($T_c$ and $T_s$) are represented by two sequences of paddle positions, which are the colored circles in Figure 6. The color of each circle is used to show the applied DVT operation.
Table 1: Pseudo-code of the DVT distance

**Algorithm 1: DVT distance**

**Input:** Two forehand drive trajectories \( T_1: A_1 A_2 \ldots A_n \), \( T_2: B_1 B_2 \ldots B_n \), where Vectors \( A_i, B_i \) are the 3D paddle center coordinates at frame \( i \).

**Output:** A distance between \( T_1 \) and \( T_2 \).

1: Declare a \((m+1) \times (n+1)\) sized distance matrix \( d[0, \ldots m, 0, \ldots n] \)

2: Initialize \( d \) as the following:

- Loop for \( i \) from 0 to \( m \): \( d[i, 0] := i \) End \( i \)
- Loop for \( j \) from 1 to \( n \): \( d[0, j] := j \) End \( j \)

3: Update the remaining elements of \( d \):

- Loop for \( j \) from 1 to \( n \):
  - Loop for \( i \) from 1 to \( m \):
    - if Distance\( (T_1[i], T_2[j]) < \theta \) then
      \( d[i, j] := d[i-1, j-1] \) Matching
    - else
      - if Distance\( (T_1[i], T_2[j]) < \sigma \) then
        \( d[i, j] := d[i-1, j-1] + 1 \) Substitution
      - else
        \( d[i, j] := \text{minimum} \)
        \( (d[i-1, j] + 1, \quad \text{Deletion} \)
        \( d[i, j-1] + 1) \quad \text{Insertion} \)
    - End \( i \)
  - End \( j \)

4: Return \( d[m, n] \)

*Procedure Distance\( (C_1, C_2) \) will return the Euclidian distance between center \( C_1 \) and \( C_2 \).*

For example, gray circles represent the matched paddle positions from \( T_s \) to \( T_c \), black or red circles represent the deleted or inserted paddle positions and green circles represent the substituted paddle positions. The operation Delete1 deletes one of the redundant paddle positions from \( T_s \), and keeps \( T_c \) and \( T_s \) at the same swing velocity. The operation
substitute marks the point where $T_c$ and $T_s$ start to diverge. When $T_c$ and $T_s$ start to have some major differences, the substitution has to be achieved by Insert and Delete2.

With the DVT results, we can derive constructive training advices for the student. For example, if several deletion operations are mixed with a sequence of matching operations in the student’s trajectory (“match – delete – match” pattern as shown in Figure 7), the student’s drive motion has a much lower velocity than the coach’s drive motion. Therefore, the student needs to adjust the speed up and strength burst timing. Similarly, several insertion operations are mixed with a sequence of matching operations
in the student’s trajectory (“match – insert – match” pattern as shown in Figure 8) will tell the student to slow down and put more control to the paddle. Moreover, DVT is able to locate the important turning points in the student’s trajectory where the trajectory starts to diverge from the coach’s trajectory. Those turning points can be found by monitoring a successive substitution and insertion operations (Figure 9), and they are the key points the student needs to prioritize in training. We will demonstrate the training advices found in our experiment section.

2.3.2.2 Distance of Impact Orientation (DIO)

Besides the paddle trajectory, the impact orientation plays an important role in the training because it determines the direction and rotation of the ball. Usually the impact frame will not be the last frame of the trajectory since the drive motion will carry on after
the contact point. We manually record the impact frame of the coach \((k)\) and the student \((l)\), and compare the paddle orientation difference. DIO can be estimated as follows:

\[
DIO = \sum_{p \in Q} |Y_k(p) - Z_l(p)|, \quad Q = \{\text{yaw}, \text{pitch}, \text{roll}\}
\] (2.5)

The final SWING distance = DVT + DIO

2.4 Experiment results

2.4.1 Capturing and comparison results

The performing subjects of our experiment are professional table tennis athlete and his student from Beijing Sport University (Jiangchuan Yu and Xiaoyi Wang). We first demonstrate our capturing phase in Figure 10.

The forehand drive motion is captured and visualized based on its paddle center trajectory and orientation. In order for the student to visualize his/her forehand drive in detail, an animation has been presented to show how fast the drive motion is and what leads to the final trajectory as well as the impact orientation. Spheres in Figure 10 correspond to the paddle center positions of each frame. For the comparison of two drive
motions from the coach and the student, we capture them in the same setting and visualize their paddle trajectories together (Figure 11, 12). As one can see, the two drive motions have similar paths at the beginning, but they diverge in later stage with different swinging velocities, trajectories, and paddle orientations. We quantify the differences of the two forehand drive motions follow the SWING distance. The Distance of Velocity Trajectory (DVT) has been visualized in Figure 13, where gray, green, red, and black spheres correspond to matched, substituted, inserted, and deleted paddle positions of the two forehand drive motions.
Figure 11: Two drive motions with different swing velocity and paddle orientation

Figure 12: The two trajectories from a different view point
Figure 13: The visualized DVT distance between the coach’s (left) and the student’s (right) trajectory
2.4.2 Generate training advices

During the training process, the comparison is shown step by step so that the student will be able to get training advices along the way. As we’ve mentioned before in the comparison phase section, when visualizing the trajectory distance (DVT), one can easily get training instructions. For example, in Figure 14(b), the trajectory pattern at the beginning is “match – delete – match” which shows that the student’s drive motion (right) has a lower velocity than the coach’s (left). It is reasonable to accelerate and use the racquet head to generate speed during this stage of the swing. Figure 14(c) shows that the comparison pattern in later stage becomes “match – substitute” or even “substitute – insert”. These are the characteristics of two turning points (P1, P2) where the coach’s and the student’s drive motions no longer match (P1) or they start to have major divergence (P2). The student needs to prioritize those key points in training.
Figure 14: Training advices generation
2.4.3 Training case study

We conduct a training case with a rookie student and a professional coach for 13 weeks. The student will perform under our training system every 3 weeks and then practice and revise his motions based on the analytical results and training advices given by the system. The visualization, comparison results and SWING distances are listed in Figure 15. From both visualization results (Figure 15) and distance timeline (Table 2) we can see with the help of our training system the student is able to learn from the coach’s velocity trajectory and stabilize his drive motion around week 10 while at least 20 weeks is needed based on past teaching experience.

In the training of paddle impact orientation, the student is able to reduce the DIO gradually during 13 weeks of training. In the future, we will seek to provide additional training advices to help the student improve the paddle impact orientation faster.

From the experiment result above, we conclude that we have introduced a new training system that can measure the difference between a student’s forehand drive motion from an expert’s and giving training advices. The significance is that the student can receive quantitative measurements (SWING distance) of the differences between her/him and the expert coach, so that s/he can constantly improve her/his movement to be as good as the coach’s. The student can learn and stabilize the forehand drive velocity trajectory in much shorter time than using the tradition methods, which in general only takes 10 to 13 weeks, while the traditional methods take at least 20 weeks. In the future, we would like to find some additional training advices on how to improve paddle orientation faster.
Figure 15: Visualization of 13 weeks training

Table 2: SWING distance timeline

<table>
<thead>
<tr>
<th></th>
<th>week1</th>
<th>week4</th>
<th>week7</th>
<th>week10</th>
<th>week13</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVT</td>
<td>111</td>
<td>91</td>
<td>66</td>
<td>36</td>
<td>20</td>
</tr>
<tr>
<td>DIO</td>
<td>12.56</td>
<td>11.36</td>
<td>11.61</td>
<td>11.18</td>
<td>10.89</td>
</tr>
<tr>
<td>SWING</td>
<td>123.56</td>
<td>102.36</td>
<td>77.61</td>
<td>47.18</td>
<td>30.89</td>
</tr>
</tbody>
</table>
Also, for a fine student who already mastered some individual techniques, we plan to build a second stage of our training system which will take a set of her/his consecutive motions and analyzing the transitions among them. It is important for the student to have a smooth transition between different techniques (e.g. transition from a forehand drive (offensive) to a block (defensive)) in real game scenarios. Our future effort in individual player training will focus on compare the student’s motion transition to a professional player and identify the student’s transition weaknesses.
CHAPTER 3: BROADCAST SPORTS VIDEO SEGMENTATION

In this chapter, we are going to introduce our new tactics training system, which is able to analyze the state of the game in detail and find new tactics for the team. Before the actual tactic analysis, we need to segment the sports video in order to eliminate tactic irrelevant shots while maintaining the integrity of the tactics relevant segments and the structure of the video. We will go over the related work of broadcast sports video segmentation and our new segmentation method, namely Segmentation of Chinese Restaurant Process (S-CRP) in the following sections.

3.1 Related work

Team sports training, such as basketball or soccer training, mainly involves players’ individual mechanics, team synergy, and tactics training. Tactics training studies the current state of the game (including players’ positions, ball procession, etc), to determine a good tactic move such as where to attack or which player to defend. Tactics training requires the team to watch past broadcast matching replays. However, broadcast sports videos are designed for the best viewing experience of the TV audience, and they are commonly synthesized by several camera views composed of both tactic relevant and irrelevant shots.
Many efforts have been made to automatically segment broadcast sports video into different types of shots [35-50]. Despite numerous research efforts in semantic sports video analysis, it is hard to develop a generic approach to sports video analysis. Currently, most works focus on specific sports games in order to investigate the roles of different information sources or statistical learning algorithms in structure analysis and semantics extraction. Although it is possible to achieve promising results on limited dataset by adopting an advanced learning approach or strong domain knowledge, it is hard to extend the approach for one kind of sports game to another, and even to the same kind of game but for different matches. Examples of methods that use domain knowledge are listed in the reference [39,43,45,51].

For example, the Field Dominant Color (FDC) is one of the most common segmentation method used in sports video analysis [51]. In most sports, the field is characterized by one distinct dominant color. The statistics of this dominant color, in the selected color space, are learned by the proposed system at start-up and automatically updated to adapt to temporal variations that may be caused by weather and/or lighting changes. The FDC (e.g. the grass color of a soccer field or the wooden color of a basketball field) is modeled as the mean color around the peak bin of the color histogram of the field, and the FDC ratio is used to determine whether a frame belongs to a long view shot (high ratio, meaning mostly field), a close-up view shot (medium ratio), or a non-field view shot (low ratio).
As another example, temporal models were introduced to segment “goal” shots in a basketball video [45]. There are some temporal relationships among the goal, crowd cheers (audio), scoreboard change, and players’ direction change.

For instance, there will be crowd cheers after a goal. In a basketball game, the most exciting moments occur when a player shoots a successful goal, thereby increasing the score of the offense team. The game becomes particularly interesting and gains momentum when there is a successful field goal which increases the score by two or three points, depending on the position from where the offense team scores the goal. Such exciting field goals are marked by high excitement from the audience, which in turn result in loud cheers from the crowd after each successful field goal. It is also noticeable that while field goals elicit a loud response from the crowd, a successful free throw has a milder response, which is understandable as the score increases by just one point in such cases.

Also, the scoreboards are updated after each valid goal. In television broadcasts, the latest score tally of the two teams are displayed at frequent intervals in order to keep the viewers up to date with the current status of the game. The scoreboard is normally displayed as separate embedded text after each legal goal. Hence, the displayed scoreboard can be used as a second key event in television broadcasts of basketball games. Most of them show the change in score after goals. Embedded text is normally inserted in the video during the post-processing time. The score change event depends on the production styles. Each valid goal is reflected in embedded text in most of the existing production styles, some of them embedded text appears at the center of the
screen after each valid goal and others display a scoreboard is displayed at the left-top or right-bottom corner of the screen throughout the game and the color of the text is changed for a few seconds after each valid goal.

Moreover, typically, a field goal attempt starts when a player begins the motion that precedes the actual shot. It continues until the shooting effort ceases and the shooter returns to a normal floor position. Soon after a field goal attempt, most of the players will start to move away from the basket where the field goal was attempted. That is, there is a change of direction of most players in the field. Change in direction of motion also occurs after other game events such as fouls, free throws and throw-ins. This change is normally reflected in camera motion as the camera focuses on players.

Those temporal models in the basketball game can be used to help video segmentation. For example, crowd cheers often follow the goal within 3 seconds and the scoreboard often updates after the goal within 10 seconds. Using these temporal models, the segmentation of goal shots is converted to the key event detection of crowd cheers, scoreboard changes, or players’ direction changes (Figure 16).

Despite their effectiveness in segmenting specific sports videos, domain knowledge methods (e.g. FDC and Temporal model) use the knowledge of the specific sports, and they are not able to be applied to other sports activities.
Some other systems use supervised learning approach [40,42,52], which does not require domain knowledge. Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (feature vector) and a desired output value (label). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a reasonable way.

This approach is more general and can be used in different types of sports videos. They define some semantic classes (long camera view, close-up camera view, audience camera view, etc.) and select a few sample frames from each class to train a classifier. The trained classifier can be used to classify each frame into a semantic class, and a shot is defined as a list of consecutive frames of the same class [40,42,49]. To better illustrates the supervised learning segmentation methods, we denote a broadcast sports video by a sequence of alphabets, where each frame is represented by a lower case alphabet and

- a. Goal → [3 sec] → Crowd cheer (audio)
- b. Goal → [10 sec] → Scoreboard
- c. Goal → [10 sec] → Change in direction

Figure 16: Temporal models
Table 3: Notations of frames and shots

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Examples</th>
<th>Representations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower case ‘n’</td>
<td>n</td>
<td>An unclassified frame</td>
</tr>
<tr>
<td>Lower case alphabet</td>
<td>a, b, ...</td>
<td>A classified frame</td>
</tr>
<tr>
<td>Upper case alphabet</td>
<td>A, B, ...</td>
<td>A shot (e.g., A = aa…a)</td>
</tr>
</tbody>
</table>

Each shot is represented by an upper case alphabet. More specifically, we use the lower case ‘n’ to represent an unclassified frame and the other lower case alphabet (a, b, …) to represent a frame after being classified into a semantic class (Table 3).

For a sixteen-frame broadcast basketball video clip $n \ldots n$, each frame is first classified to a semantic class (a, b, or c) as: $n \ldots n \rightarrow a \ldots a b b a a a a a a$. Consecutive frames of the same semantic class form a shot as: $a \ldots a a a a a a b b a a \rightarrow A B A$.

Because classification is performed at each frame, we call this setting frame level segmentation (F-L for short). One of the noticeable problems of this setting is that some misclassified (noisy) frames can easily break the original video structure. For instance, if there is a misclassified frame ‘c’ in the above video clip, the segmentation result is then as follows: $a \ldots a a a a a a a b b a a a a a a \rightarrow A C A B A$. The original 3-shot basketball clip is mistakenly segmented into a 5-shot clip. Even though a few noisy frames will not affect the frame classification accuracy very much, it may have a huge effect on the segmentation quality for tactic analysis. For example, if $A$ represents a half-court long-view shot and $C$ represents a player’s close-up view shot, the first half-court offense at the beginning of the clip will be misinterpreted as two different half-court offense rounds.

Some other systems defer the classification to shot level, which refer to as shot level segmentation (S-L for short). They first group unclassified frames into different
shots, and then each shot is identified to a semantic class [52,53]. For example, the above basketball video clip are grouped into shots as: $\text{nnnnnnnnnnnnnnnn} \rightarrow \text{nnnnnnnnnnnnnnnn}$, and then each shot is classified to a semantic class as: $\text{nnnnnnnnnnnnnnnn} \rightarrow \text{ABA}$. The most common method of grouping frames into shots is based on frame similarity – similarities between two consecutive frames are measured and shot boundaries are detected if the similarity of the two frames is below a certain threshold. In this setting, the quality of the segmentation result is mainly dependent on the correct shot boundary detection, which requires the accurately measured frame similarity and the properly chosen threshold. Noisy frames can still result in missed or falsely detected shots, and will break the original video structure in the same way as misclassified frames in the previous setting.

Many methods tried to reduce the effect of noisy frames and improve the frame classification or shot detection by building more sophisticated classifiers or use more complex features.

For examples, Zhang et al. use Fisher criterion to build their segmentation classifier [52]. Duan et al. use the power of Support Vector Machine [54] and motion vectors feature for their classification [42]. Adjeroh et al. used both color and edge based features to ensure the segmentation robustness [53]. Mohanta et al. combine several global features and local features in order to achieve classification accuracy [55].

### 3.2 The approach

We present a new method that improves the quality of the segmentation, which we group frames into shots similar to the Chinese Restaurant Process (CRP) [56] to
overcome the potentially misclassified noisy frames or false detected shots. CRP is a clustering algorithm which solve the task of grouping a set of objects in such a way that objects in the same group (cluster) are more similar to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings (including the distance function to use, the density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure. It will often be necessary to modify data preprocessing and model parameters until the result achieves the desired properties.

There are lots of different clustering algorithms in the literature. For example, K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. K-means clustering aims to partition n observations into K clusters in which each observation belongs to the cluster with the
nearest mean, serving as a prototype of the cluster. The commonly used initialization methods are Forgy and Random Partition [57]. The Forgy method randomly chooses \( K \) observations from the data set and uses these as the initial means. The Random Partition method first randomly assigns a cluster to each observation and then proceeds to the update step, computing the initial mean to be the centroid of the cluster's randomly assigned points. The Forgy method tends to spread the initial means out, while Random Partition places all of them close to the center of the data set. According to Hamerly et al. [57], the Random Partition method is generally preferable for algorithms such as the k-harmonic means and fuzzy k-means. For expectation maximization and standard k-means algorithms, the Forgy method of initialization is preferable. From the discussion above, we can see that one crucial problem of K-means clustering is the choice of the number of clusters \( K \) and the initial value of the centroids.

Another clustering algorithm, namely self-organizing map is a type of artificial neural network that is trained using unsupervised learning to produce a low-dimensional, discrete representation of the input space of the training samples. Self-organizing maps are different from other artificial neural networks in the sense that they use a neighborhood function to preserve the topological properties of the input space [58]. Like most artificial neural networks, self-organizing map operate in two modes: training and mapping. Training builds the map using input examples, while mapping automatically classifies a new input vector. A self-organizing map consists of components called nodes or neurons. Associated with each node is a weight vector of the same dimension as the input data vectors and a position in the map space. The usual arrangement of nodes is a
two-dimensional regular spacing in a hexagonal or rectangular grid. The self-organizing map describes a mapping from a higher dimensional input space to a lower dimensional map space. The procedure for placing a vector from data space onto the map is to find the node with the closest (smallest distance metric) weight vector to the data space vector. While this type of network structure seems related to feed-forward networks where the nodes are visualized as being attached, it is fundamentally different in arrangement and motivation. It has been shown that while self-organizing maps with a small number of nodes behave in a way that is similar to K-means, larger self-organizing maps rearrange data in a way that is fundamentally topological in character. Similar to K-means, one of the problems of self-organizing map is to choose the number of output neurons which determines the number of clusters.

Different from other common clustering algorithms such as K-means or self organizing map, CRP describes the table assignments of a sequence of customers arriving at a restaurant. Each customer can choose to sit at a new (unoccupied) table or sit at an occupied table with other customers. At the end of the process, a partition has been made among all customers as they sit at different tables.

The advantage of CRP is that the number of tables is not known in advance and it is only decided by the customers. Similarly, in the sports video segmentation, the number of shots is unknown in advance and it is only decided by the frames. If we consider each frame as a customer and each shot as a table, CRP provides an ideal solution to video segmentation. We extended the idea and developed a new segmentation method, S-CRP (namely Segmentation based on distance dependent Chinese Restaurant Process), which
is able to segment broadcast sports video into high quality shots for further tactic analysis.

Moreover, we observed that the common segmentation evaluation metrics are not able to reflect the quality of the segmentation result properly. They include: a) frame classification accuracy = (no. of correctly classified frames) / (no. of total frames); b) shot boundary detection standard recall = (no. of correct shot boundaries) / (no. of correct shot boundaries + no. of missed shot boundaries); c) precision = (no. of correct shot boundaries) / (no. of correct shot boundaries + no. of false shot boundaries). For example, let’s assume that the segmentation and classification ground truth of a video sequence is $aaaaa|bbb|aaaaaaa \rightarrow ABA$. Consider two different results: 1) $aaaa|bbb|aaaaaaa \rightarrow ABA$ and 2) $aaaaa|bbb|aaa|bb|aa \rightarrow ABABA$. Both segmentation results have two misclassified frames and therefore have the same frame classification accuracy (14/16 = 87.5%). The metric is not able to show preference of the two results. In the shot boundary detection standard recall and precision metrics, the first segmentation result has zero correct shot boundaries, 2 missed shot boundaries, and 2 false shot boundaries, whereas the second segmentation result has 2 correct shot boundaries, 0 missed shot boundaries, and 2 false shot boundaries. Shot boundary detection standard recall and precision metrics even favor the second result, but clearly the first result is able to reflect the structure of the original video better. We introduce a new performance metric, L-Ratio (namely Levenshtein distance Ratio), which is a more accurate measure of how well the result can match the real video structure.
3.2.1 CRP: the Chinese Restaurant Process

Before we discuss CRP in detail, let us first introduce the background of CRP. In probability theory, a Dirichlet process is a random process that is a probability distribution whose domain itself is a set of probability distributions. It has a hierarchical structure of distribution over distributions. Given a Dirichlet process $DP(H, \alpha)$, where the base distribution $H$ is an arbitrary distribution and the concentration parameter $\alpha$ is a positive real number, a draw from DP will return a random distribution over some of the values that can be drawn from $H$. That is, the support of each draw of the output distribution is always a subset of the support of base distribution. The output distribution will be discrete, meaning that individual values drawn from the output distribution will sometimes repeat themselves even if the base distribution is continuous. The extent to which values will repeat is determined by $\alpha$, with higher values causing less repetition. If the base distribution is continuous, so that separate draws from it always return distinct values, then the infinite set of probabilities corresponding to the frequency of each possible value that the output distribution can return are distributed according to a stick breaking process. The conditional distribution of one variable given all the others, or given all previous variables, is defined by the Chinese Restaurant Process (CRP).

CRP simulates the scenario in a restaurant that a sequence of arriving customers take turns to choose tables to sit at. When a customer arrives, s/he can choose to sit at the first unoccupied table (alone) or an occupied table (with arrived customers). When the
$n$th customer arrives, the probability of s/he sitting at the first unoccupied table is $\frac{\alpha}{n-1+\alpha}$

and the probability of sitting at an occupied table $k$ is $\frac{n_k}{n-1+\alpha}$, where $\alpha$ is a parameter that determines how likely a customer chooses to sit at a new table and $n_k$ is the number of customers already sit at table $k$. From here we can see that in the traditional CRP, the probability of a customer sitting at the first unoccupied table is proportional to $\alpha$, and the probability of a customer sitting at the occupied table $k$ is proportional to $n_k$. The distribution of the customer $n$’s table assignment $t_n$ can be described as:

$$P(t_n) \propto \begin{cases} \alpha & \text{unoccupied table} \\ n_k & \text{occupied table } k \end{cases}$$  \hspace{1cm} (3.1)
CRP simulates table assignment based on one rule: customers are likely to be assigned to the table that has more people. However, in most clustering problems, we want similar customers to be assigned to the same table. Blei et al. developed a distance dependent Chinese Restaurant Process (DCRP) [59], where the customer table assignment depends on the distances between the customers. The key of DCRP is that instead of being assigned to a table directly, a customer is first connected to \( N \) other customers (or her/himself) based on the distances from the customer to the other customers. Table assignments are then derived from customer connections, where two customers will be assigned to the same table if they are connected or eventually reachable through other customers’ connections (Figure 17).

In calculating customer assignments, two customers who are closer will have a higher probability to be connected to each other. Similar to traditional CRP, parameter \( \alpha \) will determine the probability of a customer connect to him/herself. The probability distribution of customer \( i \)'s connections \( c_i \) can be described as:

\[
P(c_i) \propto \begin{cases} 
\alpha & \text{self connection} \\
 f(d_{ij}) & \text{connection to } j 
\end{cases}
\]  

(3.2)

where \( d_{ij} \) is the distance between customer \( i \) and \( j \), \( f \) is a decay function that usually chosen between exponential \( f(d) = e^{-\alpha d} \) or logistic decay \( f(d) = \frac{e^{-\alpha d}}{1 + e^{-\alpha d}} \).
The exponential decay will drop very fast as the distance increases, whereas the logistic decay will decrease slowly and then drop quickly when the distance increases to a certain point.

3.2.2 S-CRP: a new segmentation method

In CRP, choosing parameter $\alpha$ is always difficult and often based on experience. However, since a shot in a broadcast sports video is always consisted of multiple frames (more than one frame), it is safe to assume that each frame will connect to another frame. Therefore, in our S-CRP, we eliminated the consideration of a frame connecting to itself, which avoided the uncertain parameter $\alpha$.

Let us go back to the broadcast sports video segmentation problem. We want the frame sequence to be grouped into unknown number of shots and they can accurately reflect the video structure for further tactic analysis. Noisy frames will make things difficult since they can be easily misclassified into other semantic classes, and shot boundaries are easily detected around them. For example, let us consider a frame sequence of the same class $a$ with some noisy frames $a'$ in the middle that are supposed to be the same class as well (Figure 18). Incorrect shot boundaries are likely to be detected around the noisy frames $a'$ and the entire shot will be divided into three shots. On the other hand, if we reduce the sensitivity of the classifier and/or boundary detection algorithm to deal with the noisy frames, shot boundaries may easily be missed between two similar but different shots. For example, let us consider a sequence consisting of frames from three different classes $a$, $b$, and $c$ (Figure 19). If $a$ and $b$ are quite different
and $b$ and $c$ are similar, the shot boundary between $b$ and $c$ is likely to be missed because of the reduced classification/detection sensitivity.

As mentioned in Section 3.1, many methods have been trying to improve the segmentation result by utilizing more sophisticated classifiers and/or domain knowledge. We take advantage of the customer (frame) assignments in DCRP to further improve the segmentation result. In DCRP, a frame has a higher probability to connect to other frames with shorter distances. Here the distance represents the appearance similarity between the two frames. Therefore, frames of the same class on the two sides of the noisy frames are
Figure 20: Bridge over the noisy gap

- a – Regular frame
- a’ – Noisy frame

Figure 21: Frames connect to their own class side

- a – Class a frame
- b – Class b frame
- c – Class c frame
- Detected shot boundary

Figure 22: Connections cross over another class

- a – Class a frame
- b – Class b frame
likely to connect to one another, and the connections will bridge over the noisy frames (Figure 20 vs. Figure 18). On the other hand, frames from two similar but different classes have lower probability to connect to one another than frames of the same class. For example (Figure 21), based on the appearance distances, even though \( b \) and \( c \) are similar, frames in \( b \) have a higher probability to connect to frames in \( b \) on the left side, and frames in \( c \) have a higher probability to connect to frames in \( c \) on the right side (Figure 21 vs. Figure 19).

There is a problem with the above method. If we calculate the distance based on frame appearance only (e.g. color, texture, or other image features), there may be connections crossing over frames that belong to a different class or classes. As in Figure 22, frames in \( b \) will be considered as noisy frames belonging to \( a \) by the algorithm. To avoid this mistake, in addition to the frame appearance distance, we added a frame time distance, which is the time elapsed between the two frames in the video sequence, to restrict frame connections. Two frames that have larger time distance will have lower probability to connect to each other.

Finally, in the S-CRP, we combine the probability distributions of appearance and time distances together using the logarithmic opinion pool \([60]\). The result is that the frames close in appearance and time are connected together. The probability distribution of frame \( i \)’s customer assignments \( c_i \) in S-CRP is described as:

\[
P(c_i) \propto f(d_{ij}).f(t_{ij}) \quad \text{connection to } j
\]

(3.3)
where $f(d_{ij})$ is the decay of the appearance distance and $f(t_{ij})$ is the decay of the time distance.

### 3.2.3 Levenshtein distance Ratio

In this section, we introduce our new performance metric Levenshtein distance Ratio (L-Ratio). As we briefly mentioned in Section 3.1, conventional metrics such as the frame classification accuracy and the shot boundary detection standard recall and precision are unable to reflect the quality of the segmentation result properly for further tactic analysis. We need some additional metric to measure the structure difference. A sports video sequence is a sequence of semantic shots, which is called the ground truth string. The segmentation result may not be exactly the ground truth string, so it has errors. For example, a broadcast basketball video clip is consisted of one long field view shot, three player close-up shots, and then another long field view shot. If we use the notation in Table 3, the ground true string is $ABBA$, where $A$ is a long field view shot and $B$ is a player close-up shot. Now the problem is how to compare and measure the difference between the segmentation result and the ground truth string and find their edit distance.

Edit distance is a measurement of quantifying how dissimilar two strings are to one another by counting the minimum number of operations required to transform one string into the other. Edit distances are widely used in applications related to natural language processing, where automatic spelling correction can determine candidate corrections for a misspelled word by selecting words from a dictionary that have a low
distance to the word in question. There are several definitions of edit distance exist, and they use different sets of string operations.

For example, the Hamming distance measures how many corresponding symbols of two strings are different. In another way, it measures the minimum number of substitutions required to change one string into the other, or the minimum number of errors that could have transformed one string into the other. However, the Hamming distance allows only substitution operations between the two input strings. Therefore, it only applies to strings of the same length.

The comparison of the video segmentation result and the ground truth string can be difficult because the correspondences of the two strings are not known in advance, and the two strings usually have different lengths. Levenshtein distance provides a way of measuring the difference between two strings—finding the minimum number of character substitution, deletion, and insertion operations [34]. Levenshtein distance is able to reflect the structural difference between the ground truth string and the segmentation result.

L-Ratio is defined as the Levenshtein distance divided by the length of the ground truth string, so that the measurement is length-irreverent. L-Ratio is normalized to the range of 0–1 using the decay function \( f(d) = e^{-\alpha d} \) for easier comparison. The L-Ratio metric represents the effort needed to edit the segmentation result to the ground truth string, and the smaller the L-Ratio, the better the segmentation result. Table 4 shows the pseudo-code describing L-Ratio formally.
**Algorithm 2: Levenshtein distance Ratio**

**Input:** The ground truth string \( S_1: x_1, x_2, \ldots, x_m \) and the segmentation result string \( S_2: y_1, y_2, \ldots, y_n \), where \( x_i, y_i \) represent individual shots.

**Output:** The Levenshtein distance Ratio between \( S_1 \) and \( S_2 \).

1: Declare a \((m+1) \times (n+1)\) sized distance matrix \( d[0\ldots m\ldots n] \)

2: Initialize \( d \) as the following:
   - Loop for \( i \) from 1 to \( m \):
     \[ d[i, 0] := i \] End i
   - Loop for \( j \) from 1 to \( n \):
     \[ d[0, j] := j \] End j

3: Update the remaining elements of \( d \):
   - Loop for \( j \) from 1 to \( n \):
     - Loop for \( i \) from 1 to \( m \):
       - if \( S_1[i] == S_2[j] \) then
         \[ d[i, j] := d[i-1, j-1] \] Matching
       - else
         \[ d[i, j] := \text{minimum} \]
         \[ (d[i-1, j-1] + 1, \text{Substitution}) \]
         \[ d[i-1, j] + 1, \text{Deletion} \]
         \[ d[i, j-1] + 1) \text{Insertion} \]
     End i
   End j

4: Calculate L-Radio based on \( d[m, n] \).
   \[ \text{L-Ratio} := \exp \left( -\frac{a}{d[m, n]} \right) \]

5: Return L-Ratio

---

### 3.3 Experiment results

The video data used in our experiment are video sequences from Man’s basketball Beijing 2008 Summer Olympic Games in the elimination round (8 games). We present the results of our S-CRP together with F-L and S-L segmentation. The classifiers used in
our experiment include a simple Nearest Neighbor (NN) classifier and a more sophisticated Support Vector Machine (SVM) classifier.

For F-L, each frame is directly classified into a semantic class. Consecutive frames of the same semantic class form a shot (e.g. \textit{aaabaaabb} \rightarrow \textit{ABAB}).

For S-L, unclassified frames are first grouped into shots according to frame similarities and a similarity threshold (e.g. \textit{nnnnnnnnnn}). The class of each shot is voted by the frames within (e.g. \textit{aabaaabb} \rightarrow \textit{aabaaabb} \rightarrow \textit{ABABA}).

For S-CRP, unclassified frames are first connected to \(N\) other frames according to appearance and time distance distributions and shots are formed based on frame connections. A good candidate of \(N\) can be determined using a validation set. We choose \(N = 4\) in our experiment. We defined five different classes to cover the semantic meaning of the video data and use upper case letters (A-E) to represent each shot of a specific class. The five different semantic classes are: Long Field (A), Medium Field (B), Team-1 player Close-up (C), Team-2 player Close-up (D) and Non-field (E) (Figure 23(A-E)). We use color histogram (in HSV space) as the image feature to represent each frame. Thirty frames of each semantic class are randomly selected from the target video as training samples. For NN classifier, we represent each class by the average vector of the 30 training samples, and each frame is classified to the nearest class according to Euclidian distance. For SVM, we use linear kernel with penalty parameter \(C = 10\) to train the classifier. The quality of the segmentation results are evaluated using the frame classification accuracy and the L-ratio metrics.
Figure 23: Examples of the four semantic classes

In Table 5 we have shown an example of our generated segmentation strings from the Gold Medal game USA vs. Spain 2nd half first three minutes using different methods. We can see that the structure of the segmentation results from S-CRP is the most similar to the ground truth.

In Table 6 we present the average frame classification accuracy and L-ratio of all the 8 games in the elimination round. S-CRP has the best frame classification accuracy and L-ratio, and even if a trivial classifier (NN) is used, S-CRP can still provide very good segmentation quality. S-CRP also has huge L-ratio improvements compare to other methods which will significantly reduce the effort needed to edit the S-CRP segmentation result to the video ground truth for further tactics analysis.
Table 5: Segmentation result strings

<table>
<thead>
<tr>
<th>Method</th>
<th>Result strings</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-L + NN</td>
<td>ABADCDCDCDADADACDADADACCDADAB</td>
</tr>
<tr>
<td></td>
<td>ABABACACDCBABADADADADACADACDCDCD</td>
</tr>
<tr>
<td>F-L + SVM</td>
<td>ABDACDCDCDADADACACACABA</td>
</tr>
<tr>
<td></td>
<td>CACDADADADACADCDADCDADCDADCDADCDABABABABA</td>
</tr>
<tr>
<td>S-L + NN</td>
<td>AAAACCDACAADDACADACDA</td>
</tr>
<tr>
<td>S-L + SVM</td>
<td>ADACCDACDADDACADADCA</td>
</tr>
<tr>
<td>S-CRP + NN</td>
<td>ABACDADACDADDACADACADADACADCA</td>
</tr>
<tr>
<td>S-CRP + SVM</td>
<td>ABACCDDACDADAACADDACA</td>
</tr>
<tr>
<td>Ground Truth</td>
<td>ABACDADDACDADAACADDADA</td>
</tr>
</tbody>
</table>

Table 6: The frame classification accuracy and L-ratio

<table>
<thead>
<tr>
<th>Method</th>
<th>Frame acc</th>
<th>L-ratio (0~1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-L + NN</td>
<td>85.3%</td>
<td>0.79</td>
</tr>
<tr>
<td>F-L + SVM</td>
<td>89.1%</td>
<td>0.70</td>
</tr>
<tr>
<td>S-L + NN</td>
<td>90.1%</td>
<td>0.21</td>
</tr>
<tr>
<td>S-L + SVM</td>
<td>92.9%</td>
<td>0.16</td>
</tr>
<tr>
<td>S-CRP + NN</td>
<td>96.7%</td>
<td>0.04</td>
</tr>
<tr>
<td>S-CRP + SVM</td>
<td>97.0%</td>
<td>0.04</td>
</tr>
</tbody>
</table>

Significant effort has been put on automatically segmenting broadcast sports videos. However, during the segmentation process, noisy frames can easily break the original video structure and lead to miss or falsely detected semantic shots. We have introduced a novel method (S-CRP), which uses two segmentation criteria, namely appearance and time distances to find the connections between individual frames and reduce the negative effect of noisy frames without using domain knowledge. It takes advantage of the customer (frame) assignments in DCRP and is able to generate high quality segmentation results even with a trivial classifier.
Compared to the segmentation results of the traditional methods, S-CRP’s has higher frame classification accuracy and more structural similarity to the video ground truth. Moreover, we have introduced a new performance metric, Levenshtein distance Ratio. Compared to the conventional metrics, L-Ratio is more accurate on measuring how well the result matches the ground truth.
CHAPTER 4: TEAM TACTICS TRAINING

In this chapter, we introduce a novel tactics training system that is able to assess the state of the basketball game in detail, and find better defense tactics. Instead of analyzing only player/ball positions, our system studies players’ attributes together with their positions to accurately estimate each player’s offense threat or defense ability for a certain applied defense tactic. We then formulize an optimization problem to find the defense tactic that minimizes the offense team’s threat. The optimal defense tactic will be suggested to assist tactics training.

4.1 Related work

Team sports training, such as basketball or soccer training, mainly involves players’ individual mechanics, team synergy, and tactics training. Tactics training studies the current state of the game (including players’ positions, ball procession, etc), to determine a good tactic move such as where to attack or which player to defend. The most common way for tactics training is to study from past games. For example, we can study the opponent team’s offense tactics and figure out the flaws in our own defense tactics. Traditionally, this requires the coach to analyze past matching replays of the team, study different opponents’ offense tactics, evaluate our team members’ defense
performance, and find if there could be any improvement. The quality of tactics training is limited by the coach’s available time and game knowledge.

Many efforts have been made to assist tactics training [61-78]. For example, Fu et al. develop a screen action detection system for a basketball game [61]. A screen action is a blocking move performed by an offense player to a defense player, in order to create shooting or passing opportunities for an offense teammate. They use Kalman filter to track the players’ trajectories, and detect screen actions based on players’ relative distances. Zhu et al. develop a goal detection system in a soccer game [62]. They extract goal events based on game time and web-casting text alignment. For those detected goal events, they further divide the soccer field into different regions to analyze if the opponent team is likely to initiate side attacks or center attacks.

Despite their good performances in screen detection, goal detection, or other type of event detections, the related systems provide only low level assistances in tactics training. They are able to capture certain events in the game and may save the coach some analyzing time, but they provide little help towards finding a better tactic. For example, successful screen detections will help the coach to make a statistical reference, but provide no insights on how to rotate defense to deal with the opponent’s screen actions. Also, being able to detect goal events and find out the opponent prefers side-lane attacks or center attacks will be helpful, but such information does not lead to any suggested defense tactics. Moreover, the tactics analysis information provided in such systems is not enough and sometimes incomplete. For instance, the opponent team may prefer side attacks because their star player roams to the side-lane (the defense tactic
should be build around the star player no matter which lane he goes) or may be due to a weak side-lane defender on our team (the defense tactic should focus on helping our weak defender on the side-lane).

It will not be sufficient for tactics analysis without considering players’ attributes. Player’s attributes are vectors that able to reflect players’ mechanics in the target sports. For instance, in soccer, players’ attributes may include speed, ball control, volley, etc, and in basketball, we may consider medium shot, block rate, ball handle, etc. Different offense players should be weighted differently in a defense tactic based on their attributes. For example, when making defense decisions, star players are likely to be weighted higher because they have higher attribute values compare to normal players.

Furthermore, besides players’ attributes, their positions should also be considered when making defense decisions. Let us consider an example in a basketball game. An offense player in the opponent team is extremely good at inside shot and layup (or dunk), but performs poorly outside the 3-point line (i.e. he has high attribute value on inside shot and layup but low attribute value on 3-point shot like traditional center in NBA such as Shaquille O’Neal). He will have a large scoring potential when inside the restricted area while having minimal threat outside of 3-point line. It will be the best to double-team the player inside the restricted area while leave him unattended outside the 3-point line.

We present a new tactic analysis system that is able to provide high level assistances in basketball tactics training. Different from existing systems, our system is able to assess the state of the sports game, which considering players’ positions along with their attributes, and find out better defense tactics for the team. Each player’s
attributes can be initialized based on the statistics of past seasons, or they can be customized by the coach according to his/her game knowledge and emphasis. We formulize an optimization problem, where the goal is to minimize the threat of the offense team after applying our defense tactic. Finally, the optimal defense tactic that yields the minimal remaining threat will be suggested to assist tactics training.

4.2 Players and ball tracking

In order to analyze and evaluate tactics in a basketball game, we need to track positions of each player and the ball.

Object tracking is the process of locating a moving object over time using a camera. It has applications in numerous of fields: human-computer interaction, surveillance, traffic control, medical imaging and video editing [79-81]. Object tracking can be time consuming due to the amount of data in the target video.

The objective of object tracking is to associate target objects in consecutive video frames. The association can be especially difficult when the objects are moving fast relative to the frame rate. Another situation that increases the complexity of the problem is when the tracked object changes orientation over time. For these situations object tracking systems usually employ a motion model which describes how the image of the target might change for different possible motions of the object.

To perform object tracking an algorithm analyzes sequential video frames and outputs the movement of targets between the frames. There are a variety of algorithms, each having strengths and weaknesses. Considering the intended use is important when choosing which algorithm to use.
There are two major techniques used in tracking algorithms: (1) target representation and localization and (2) filtering and data association.

Target representation and localization is a bottom-up process. Methods using this technique give a variety of tools for identifying the moving object. Locating and tracking the target object successfully is dependent on the algorithm. For example, blob tracking is useful for identifying human movement because of the articulate and dynamic natures of a human figure [82]. The computational complexity for these algorithms is low. Common target representation and localization algorithms include blob tracking, kernel-based tracking and contour tracking.

Filtering and data association is a top-down process, which involves incorporating prior information about the scene or object, dealing with object dynamics, and evaluation of different hypotheses. These methods allow the tracking of complex objects along with more complex object interaction like tracking occluded objects behind obstructions [83]. Additionally the complexity is increased if the video tracker is not mounted on rigid foundations but moving from time to time, where typically an inertial measurement system is used to pre-stabilize the video tracker to reduce the required dynamics and bandwidth of the camera system. The computational complexity for these algorithms is usually much higher. Common filtering algorithms include Kalman filter and particle filter.

The state of the art object tracking methods in the literature [84-89] are different in aspects such as object representation, search mechanism and model update. For example, Zhong et al. propose a robust object tracking algorithm using a collaborative
model [84]. They deal with object drastic appearance change challenge with a robust appearance model that exploits both holistic templates and local representations. They develop a sparsity-based discriminative classifier to compute the confidence value that assigns more weights to the foreground than the background. They also develop a sparsity-based generative model where a histogram-based method has been used to take the spatial information of each patch into consideration to handle occlusion.

Hare et al. present a framework for adaptive visual object tracking based on structured output prediction [85]. They explicitly allow the output space to express the needs of the tracker and avoid an intermediate classification. They use a kernel-structured output support vector machine (SVM) to provide adaptive tracking. The highlight of their method is their budgeting mechanism which prevents the unbounded growth in the number of support vectors to allow for real-time applications.

Kalal et al. apply their P-N learning method to on-line learning of object detector during tracking [86]. The idea of their P-N learning method is based on a novel paradigm for training a binary classifier from both labeled and unlabeled examples. They have shown that the performance of a binary classifier can be significantly improved by the processing of structured unlabeled data, i.e. data are structured if knowing the label of one example restricts the labeling of the others. The learning process is guided by positive (P) and negative (N) constraints which restrict the labeling of the unlabeled set. With P-N learning, an accurate object detector can be learned from a single example and an unlabeled video sequence where the object may occur.
Jia et al. develop a simple and robust tracking method based on the structural local sparse appearance model [87]. Their model representation exploits both partial information and spatial information of the target based on an alignment-pooling method. The similarity obtained by pooling across the local patches helps to locate the target more accurately and handle occlusion. As opposed to most sparse representation based trackers, which only consider the holistic representation and do not make full use of the sparse coefficients to discriminate between the target and the background, they employ a template update strategy which combines incremental subspace learning and sparse representation. This strategy adapts the template to the appearance change of the target with less possibility of drifting and reduces the influence of the occluded target template to handle multiple similar objects or occlusions in the scene.

In addition to the common challenging aspects in visual tracking such as unconstrained environments, object changes in appearance, varying lighting conditions, cluttered background, frame-cuts, and similar appearance, a major factor causing tracking failure is when the target leaves the field of view leading the tracker to follow another similar object, and not reacquire the right target when it reappears.

Dinh et al. present a method to address this problem by introducing two terms: Distracters and Supporters [88]. In their work, both of them are automatically explored using a sequential randomized forest, an online template-based appearance model, and local features. Distracters are regions which have similar appearance as the target and consistently co-occur with high confidence score. The tracker must keep tracking these distracters to avoid drifting. Supporters, on the other hand, are local key-points around
the target with consistent co-occurrence and motion correlation in a short time span. They use Supporters to verify the genuine target.

All of the state of the art methods have their relative strength in different performing factors, such as illumination variation, occlusion, as well as background clutters, and there exists no single tracking approach that can successfully handle all scenarios. Wu et al. presented a detailed evaluation of the state of the art tracking methods using various testing video sequences and evaluation metrics [90]. They also annotated their test sequences with different properties and challenging aspects, such as illumination variation, scale variation, occlusion, deformation, etc.

Our target video, the broadcast basketball footage, has some challenging aspects such as illumination variation, player/ball occlusion and non-rigid object deformation. Moreover, the background near a player sometimes has the similar color or texture of that player (i.e. background clutters). Based on the performance comparison reported [91], we use Sparsity-based Collaborative Model (SCM) [84] as our tracker, as it performs fairly well in those challenging aspects. The player’s tracking and ball tracking results are presented in Figure 24 and Figure 25.

Figure 24: Player tracking
4.3 Camera calibration

Players’ positions derived from the tracker are 2D coordinates in the image coordinate system which is the projection of the real basketball court coordinate system. Before conducting any tactic analysis, we need camera calibration to calculate the camera projection matrix in order to transform a 2D image coordinate to a 3D court coordinate. The transformation of a 2D image coordinate \((u, v)\) to its 3D court coordinate \((x, y, z)\) can be described by the following equation:

\[
(u, v, 1)' = M(x, y, z)', \quad M = \begin{pmatrix} m1 & m2 & m3 & m4 \\ m5 & m6 & m7 & m8 \\ m9 & m10 & m11 & 1 \end{pmatrix}
\] (4.1)
The projection matrix $M$ has 11 unknown parameters, which can be calculated from six non-coplanar reference points whose positions are known in both image and court coordinate systems [92, 93]. We choose the two top corner points of the backboard and four corner points of the restrict area as our reference points (named from a-f). The origin of the court coordinate system is set at the top left corner of the court with X axis overlapping with the sideline, Y axis overlapping with the baseline, and Z axis perpendicular to the court plane (Figure 26).

The court positions of the 6 reference points (a-f) are listed in Table 7. The image positions of reference points across frames can be located by object tracking. Figure 27 shows the result of reference point (point b) tracking using SCM tracker.
Table 7: Court positions of the 6 reference points (in meters)

<table>
<thead>
<tr>
<th>Point</th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1.2</td>
<td>6.72</td>
<td>3.95</td>
</tr>
<tr>
<td>b</td>
<td>1.2</td>
<td>8.52</td>
<td>3.95</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>5.18</td>
<td>0</td>
</tr>
<tr>
<td>d</td>
<td>5.79</td>
<td>5.18</td>
<td>0</td>
</tr>
<tr>
<td>e</td>
<td>5.79</td>
<td>10.06</td>
<td>0</td>
</tr>
<tr>
<td>f</td>
<td>0</td>
<td>10.06</td>
<td>0</td>
</tr>
</tbody>
</table>

4.4 Optimal defense tactic

After players/ball tracking and camera calibration, we are able to compute players/ball positions in the court coordinates. In this section, we introduce players’ attributes, which are vectors describing how good the player is at the sport. Players’ attributes, together with their court positions, are used to estimate their offense threats or defense abilities.

4.4.1 Player attribute vector
Table 8: Offensive and defensive attributes correspondences

<table>
<thead>
<tr>
<th>Offensive</th>
<th>Defensive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inside Shot</td>
<td>Block Rating +</td>
</tr>
<tr>
<td>Close Shot</td>
<td>Contest shot Rating</td>
</tr>
<tr>
<td>Medium Shot</td>
<td></td>
</tr>
<tr>
<td>3-Point Shot</td>
<td></td>
</tr>
<tr>
<td>Layup</td>
<td></td>
</tr>
<tr>
<td>Ball Handle</td>
<td>Steal Rating</td>
</tr>
<tr>
<td>Drive the Lane + Quickness</td>
<td>Take Charge + Quickness</td>
</tr>
</tbody>
</table>

Each player needs to have an attribute vector to represent his/her in-game mechanics so that the offense threat or defense ability can be accurately estimated. The attribute vectors are consisted of attributes related to fundamental basketball techniques, including Inside Shot, Close Shot, Medium Shot, 3-Point Shot, Layup, Ball Handle, Drive the Lane, Block Rating, Contest shot Rating, Steal Rating, Take Charge, and Quickness. Some of the attributes are offensive attributes, and can be affected by their defensive counterparts. As an example, an offense player’s 3-Point Shot will be affected by a defender’s Block Rating or Contest shot Rating. In Table 8, we list the correspondences of offensive attributes and defensive attributes.

Each attribute has a 0-100 numerical value with 100 the highest. We initialize the value of each player based on the statistics of past seasons (or they can be customized by the coach according to his/her game knowledge and emphasis). Also, as briefly mentioned in Section 1, some offensive attributes will be effective only when the player is at a desired court position. For example, 3-Pointer Shot will only be effective when the player is around the 3-point line and Layup will be effective only when the player is near
Table 9: The effective range (distance to the hoop in meters)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Inside Shot</th>
<th>Close Shot</th>
<th>Medium Shot</th>
<th>3-Point Shot</th>
<th>Layup</th>
<th>Drive the Lane</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>0~1.5</td>
<td>1.5~3</td>
<td>3~6.75</td>
<td>6.75~8</td>
<td>0~1</td>
<td>3~10</td>
</tr>
</tbody>
</table>

the hoop. Therefore, we further introduce an effective range vector $E$ (Table 9), which describes offensive attributes’ effective range by the distance to the hoop.

### 4.4.2 Player offense threat

The offense threat of a player depends on his/her offensive attributes and court position. We estimate the offense threat based on the effectiveness of offensive attributes. An offensive attribute will contribute a lot to the threat if the player is located within the effective range of that attribute. On the other hand, if the player is far away from the effective range of a certain offensive attribute, the attribute will be less important and attenuated by some decay function $f$.

More formally, for an offender $k$, let $A^k$ be its offense attribute vector and $A_{i}^{k}$ be the $i$th offensive attribute. Let $E$ be the effective range vector and $E_i$ be the effective range of the $i$th offensive attribute. The player’s offense threat $O^k$, which measures the effectiveness of $A^k$, can be calculated as the following equation set, where $H^k$ is the distance from the player to the hoop, and $f$ is the exponential decay function.

$$O^k_i = A_{i}^{k} \cdot f(H^k, E_i), \quad f(H^k, E_i) = e^{-\alpha|H^k - E_i|}$$

(4.2)

Furthermore, we put some emphasis on ball procession player (player *) by an amplifier $\mu > 1$ as he/she is able to initiate the attack directly. The offense threat for the ball
procession player \( * \) is estimated as:

\[
O_i^* = \mu A_i^* \cdot f(H^*, E_i)
\]  

(4.3)

### 4.4.3 Player defense ability

The defense ability of each player depends on its defensive attributes as well as its position relative to the assigned offender. An offender cannot be well defended if the defender is at a far away position. Moreover, if an offender has the ball possession, s/he cannot be well defended if the defender is at a bad defense angle (e.g. behind the offender). Therefore, we attenuate the defender’s attribute values based on the relative distance and defense angle to the assigned offender with the decay function \( F \). Figure 28 shows different examples of defense distance \( d \) and angle \( \alpha \) between an offender (red circle) and a defender (green circle).

More formally, for a defender \( l \) with assigned offender \( k \), let \( B_i^l \) be the defense attribute vector and \( B_i^{l_i} \) be the defensive attribute corresponds to the \( i \)th offensive attribute \( A_i^{l_i} \). The player’s defense ability \( D_i^{l,k} \), which is the attenuation of \( B_i^l \), can be calculated as the following equation set, where \( d^{l,k} \) and \( \alpha^{l,k} \) are the relative distance and defense angle between defender \( l \) and offender \( k \) respectively. The form of decay function \( F \) depends on if the assigned offender \( k \) has the ball possession.

\[
D_i^{l,k} = B_i^l \cdot F(d^{l,k}, \alpha^{l,k}), \text{ where}
\]

\[
F(d, \alpha) = \begin{cases} 
  f(d) = e^{-b/d} & k \text{ without ball possession} \\
  f(d) \cdot f(\alpha) = e^{-b/d} \cdot e^{-c/\alpha} & k \text{ with ball possession}
\end{cases}
\]

(4.4)
4.4.4 Optimal defensive tactic

In a basketball game, there are five different roles for the five players in a team, namely Point Guard (PG), Shooting Guard (SG), Small Forward (SF), Power Forward (PF) and Center (C). We will follow the convention and represented each role with a number from 1 to 5 (Table 10).

<table>
<thead>
<tr>
<th>Role</th>
<th>PG</th>
<th>SG</th>
<th>SF</th>
<th>PF</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
A man-to-man defense tactic (defense assignment) can be abstracted by a set of connections between the defense and offense player, where multiple defenders can connect to a single offender (e.g. double team), but not vice versa (Figure 29).

Therefore, we represent each defense tactic by a vector $q$, which contains the connections from each defender to its assigned offender. For example, the defense tactic in Figure 30 can be represented as $q = [(1,1), (2,2), (3,2), (4,3), (5,5)]$. Because each defender will have 5 different assignment options, the set of all possible defense tactics $Q$ will have $5^5$ different tactics in total.

Our goal is to find the optimal $q \in Q$ that could minimize the offense team’s remaining offensive threat after apply the defense tactic. From the previous discussions, we are able to estimate the offense threat $O^i$ of offender $k$, and the defense ability $D^{(i,k)}$ of
defender \( l \) to offender \( k \). If we consider the whole offense team, the remaining offense threat after applying a defense tactic \( q \) can be represented by the following equation:

\[
R = \sum_{k} (O^k - \sum_{(l,k) \in q} D^{(l,k)})
\]

(4.6)

Because an offender’s remaining offense threat should not be negative, we revise the equation to make sure every attribute in \( R \) is greater or equal to zero as follows, where \( \text{abs}(P) \) is the absolute value of vector \( P \) (i.e. \( \text{abs}(P) \) will take absolute value of each attribute of \( P \)).

\[
R' = \sum_{k} (O^k - \sum_{(l,k) \in q} D^{(l,k)}) + \text{abs}(O^k - \sum_{(l,k) \in q} D^{(l,k)}) / 2
\]

(4.7)

The remaining threat of the offense team will be represented by \( |R'| \) where \( |P| \) is the \( L_1 \)-norm of vector \( P \). Our optimization problem of finding the best defense tactic can be formulated as:

Minimize \(|R'|\)  

(4.8)

Where \( R' = \sum_{k} (O^k - \sum_{(l,k) \in q} D^{(l,k)}) + \text{abs}(O^k - \sum_{(l,k) \in q} D^{(l,k)}) / 2 \), and \( q \in Q \)

The optimal defense tactic will be found after iterate through all \( 5^5 \) possible tactics.
Figure 30: The screen action between SAS Duncan and Parker

(a)                                                                 (b)

(c)                                                                 (d)

Figure 31: The defense tactic of MIA (first footage)
4.5 Experiment result

The video footage in our experiment are selected from the 2012-2013 NBA playoffs finals, San Antonio Spurs (SAS) vs. Miami Heat (MIA). We first focus on situations where the defense team made some bad tactic choices that cause the offense team to score easily. We demonstrate the optimal defense tactic found by our system, which provides a more reasonable defense strategy, to help training the defense team to make some better decisions and avoid giving such easy score opportunities to the opposing team in the future. Secondly, we demonstrate the optimal tactics found is indeed helpful by analyzing situations where the defense team’s tactics actually matches the optimal tactics and results in a successful defense. In addition, the tactics generated by our system have been evaluated and commented by three professional basketball coaches from Beijing Sports University. We report their opinions about our training system.

The first footage can be summarized as follows. It starts off with offense team SAS’s Center player Duncan (#21) coming out of the restricted area to take screen action for the Point Guard Parker (#9), who has the ball possession. The current defense tactic for MIA is Chalmers (#15) on Parker, Wade (#3) on Green (#4), James (#6) on Ginobili (#20), Miller (#13) on Leonard (#2) and Bosh (#1) on Duncan (Figure 30).

Because the screen action from Duncan to Parker, MIA’s Chalmers is in a bad angle to have impactful defense (on Parker). MIA uses Bosh as the backup defense in order to keep pressure on the ball (Figure 31(a-b)), while rotating Miller to prevent Duncan (who should be defended by Bosh) from attacking the rim (Figure 31(c-d)).
At the point of Figure 31(c), Chalmers has regained the defense position/angle for Parker, but Bosh is in a bad position to defend any SAS players. The defense rotation ends up with an open shot opportunity when Parker passes the ball to Leonard (who should be guarded by Miller) (Figure 32).

As opposed to the MIA’s defense tactic used in the footage, our system finds the defense tactic that minimizes the threat of the offense team. First, we initialize players’ attributes based on the statistics of past seasons [94], which include players’ name (Name), number on the court (#), players’ team (Team), players’ role (R), and players’ basketball mechanics. We consider the total of 12 mechanics, namely Inside Shot (IS), Close Shot (CS), Medium Shot (MS), 3-Point Shot (3PS), Layup (LU), Ball Handle (BH), Drive the Lane (DL), Block Rating (BR), Contest shot Rating (CR), Steal Rating (SR), Take Charge (TC), and Quickness (QK) (Table 11).

From players’ attributes, we can observe some flaws of the applied defense tactic in the footage. For example, MIA uses Bosh to cover Parker when Chalmers is in a bad defense position/angle. Parker can perform a very fast dribble breakthrough (DL = 79,
BH = 93, QK = 91) while Bosh is a relatively slow moving Center (SR = 54, TC = 80, QK = 64). It will be difficult for Bosh to follow the pace of Parker.

Secondly, MIA rotates Miller to defend Duncan from attacking the rim (Figure 31(c-d)), where Duncan is at one of his most comfortable attacking positions. He can either initiate a Medium Shot attack or a Close Shot attack after one dribble. His corresponding attribute values (MS = 79, CS = 82) are much higher than Miller’s Block Rating or Contest shot Rating (BR = 27, CR = 27). Therefore, the defense tactic in the footage will not only result in an open shot opportunity for Leonard, but may also result in the extremely challenging match up where Miller has to defend Duncan in the restricted area.

Figure 33 demonstrates the optimal tactic we found after estimating offense team’s threat and defense team’s defense ability. When Duncan taking screen action for
Parker, instead of sending Bosh to cover Parker, the team should let him defend Duncan to avoid the rotation of Miller. That is, Wade, Miller and Bosh should stay with their defense assignment (Wade on Green, Miller on Leonard and Bosh on Duncan). On the other hand, the team should send James to cover Parker until Chalmers regain his defense position/angle. This tactics avoids the challenging match up where Bosh defends Parker and Miller defends Duncan in the restricted area, and uses a quicker moving Small Forward James (SR = 74, TC = 88, QK = 78) to cover Parker. Since the distance is relatively close from James to both Parker and Ginobili, even if Parker chooses to pass to Ginobili, it is likely for James to regain his defense position for Ginobili.

All of the three coaches in Beijing Sports University showed their favorable evaluations in the optimal tactics provided by our system. They commented that when the offense team conducts a quick screen action (pick and roll) between the Center and the Point Guard, the defense team normally needs to send their Center to temporarily cover the opposite team’s Point Guard. This defense tactics is more predictable and is likely to result in defense switch between Center and Point Guard which creates more
opportunities for the offense team, such as penetration by the Point Guard or attacking the rim directly by the Center. They feel the optimal tactics found by our system opens up promising defense possibilities. If a more agile player (e.g. a Small Forward like James) is near the pick and roll location, sending such a player will prevent the defense switch and boost the defense quality.

The second footage starts off with offense team SAS’s Small Forward Green (#4) penetrates the MIA’s front line defense. Green’s corresponding defender Allen (#34) is at a bad defense position/angle and he is not able to mitigate the threat of Green effectively. The optimal defense tactic found by our system is that Andersen (#11) will help Allen to defend Green (#4) while James (#6) move back to defend Duncan (#21) as shown in Figure 34.

The rest of the clip shows that due to the pressure of Andersen, Green (#4) cannot attack the rim directly, and he ends up pass the ball to Duncan (#21). Because of the runback of James, MIA is able to mitigate the threat of Duncan and successfully defends this round (Figure 35(a-d)).
Figure 34: The optimal defense tactic (second footage)

(a)                                                                 (b)

(c)                                                                 (d)

Figure 35: The defense tactics of MIA (second footage)
4.6 Conclusion

In team sports tactics training, the coach usually needs to watch the past matching replays in order to find a better tactics for the team. Many efforts have been made to assist tactics training, but without considering players’ attributes. Related tactic analysis systems provide only low-level assistances in tactics training such as game event detection, and are unable to find a better tactics. In this section, we present a novel tactics training system that is able to assess the state of the game, which includes players’ attributes together with their court positions, and estimates the offense team’s threat as well as the defense team’s defense ability. We formulate an optimization problem that minimizes the offense team’s remaining threat, and the optimal tactics that results in the minimal remaining threat is suggested to the tactics training.
CHAPTER 5: FUTURE WORK

Possible future works on our research are:

In our virtual system for individual mechanics training, we are able to generate training advices and significantly reduce the training time to stabilize the forehand drive velocity trajectory. But for the training of paddle impact orientation, the experiment result only showed a slow improvement. We would like to find some additional training advices on how to improve paddle impact orientation faster.

Our future effort will focus on working with coaches that have different anatomy (i.e. height, arm length, etc.) to create a training database. Based on the student’s anatomy, we choose the coach that has the most similar anatomy. We can also synthesis the most similar anatomy coach through interpolation. Also, for a fine student who already mastered some individual techniques, we plan to build a second stage of our training system which will take a set of his/her consecutive motions and analyzing the transitions among them. It is important for the student to have a smooth transition between different techniques (e.g. transition from a forehand drive (offensive) to a block (defensive)) in real game scenarios. Our future effort in individual player training will focus on compare the student’s motion transition to a professional player and identify the student’s transition weaknesses.
In our tactics training system for the basketball game, we are able to estimate the offense team’s threat as well as the defense team’s defense ability and provide the optimal tactics that result in the minimal remaining threat. In our future work, we plan to work with professional basketball analysts and coaches to create a database of half-court defense video clips with manual annotations of the best defense tactics. The database will be used to tweak the parameters and add new factors to our tactics training system.

Moreover, besides the man-to-man defense model in basketball games, we plan to find some mathematical representation of the region defense model which is commonly used in sports such as soccer.
CHAPTER 6: CONCLUSION

This dissertation intends to create novel systems for competitive sports training. We create a virtual system for individual mechanics training and a system for team tactics training.

Our virtual individual mechanics training system is able to improve a student athlete’s performance in activities such as learning a table tennis forehand drive. Unlike other existing virtual training systems that only capture and visualize subjects’ motions in a virtual environment to assist training, our system further quantifies the distance between the student’s and the coach’s motions, and generate training advices to improve performance. A novel measurement – SWING distance – has been introduced to quantify the distance between two forehand drive motions (velocity trajectory and paddle orientation). The significance of our training system is that the student can receive quantitative measurements (SWING distance) of the differences between her/him and the expert coach, so that s/he can constantly improve her/his movement to be as good as the coach’s. The student can learn and stabilize the forehand drive velocity trajectory in much shorter time than using traditional methods, which in our experiment only takes 10 to 13 weeks, while the traditional methods take at least 20 weeks.

Our team tactics training system is able to consider individual player’s attributes and accurately estimates the offense team’s threat as well as the defense team’s defense
ability through broadcast sports videos, and find out better defense tactics for the defense team. Before the tactics analysis, since noisy frames in broadcast sports videos can easily break the original video structure and lead to missed or falsely detected shots and disrupt further tactics analysis, we have introduced a novel method named S-CRP (Segmentation based on distance dependent Chinese Restaurant Process), which can reduce the negative effect of noisy frames during segmentation. S-CRP uses two segmentation criteria, namely appearance and time distances to find connections of individual frames. It takes advantage of the customer (frame) assignments in DCRP and is able to reduce the negative effects of noisy frames without the use of domain knowledge. It also can generate high quality segmentation results without the use of more sophisticated classifiers. In addition, we find that the conventional performance evaluation metrics such as frame classification accuracy and shot boundary detection standard recall/precision are unable to reflect the quality of the segmentation properly. We introduced a new performance metric, namely Levenshtein distance Ratio, which gives a more accurate measure of how well the segmentation result can match the original video structure. Compared to the segmentation results of the traditional methods, S-CRP’s has higher frame classification accuracy and more structural similarity to the video ground truth.

After eliminating the tactics irrelevant shots, our system studies each individual player’s attributes together with his/her court position and position relative to the ball to accurately estimate the offense threat or defense ability. Then we formulate an optimization problem to minimize the offense team’s remaining threat. The optimal
defense tactics found by our system have received favorable evaluations by three professional basketball coaches in Beijing Sports University from a professional point of view, and as opposed to some well known defense conventions, our system opens up new promising defense tactics and possibilities.
BIBLIOGRAPHY
BIBLIOGRAPHY

[1] L. Hodges, M. Nordby, and D. Seemiller, *Instructor’s guide to Table Tennis*, USA Table Tennis, 2000, pp. 3.


<table>
<thead>
<tr>
<th>Reference</th>
<th>Authors</th>
<th>Title</th>
<th>Conference/Journal</th>
<th>Pages</th>
</tr>
</thead>
</table>
CURRICULUM VITAE

Hao Sun has been a PhD student in the Department of Computer Science at George Mason University (GMU) since 2008. He received his BE in Computer Science in 2008, from Beijing University of Technology, China. His research focuses on computer graphics, virtual reality and video analysis.