Human Body Pose Estimation Framework for Team Sports Videos Integrated with Tracking-by-Detection

January 2016

A thesis submitted in partial fulfilment of the requirements for the degree of
Doctor of Philosophy in Engineering

Keio University

Graduate School of Science and Technology
School of Integrated Design Engineering

Masaki Hayashi

Abstract

Multitarget tracking in team sports videos has been widely studied for the purpose of analyzing tactics. Since the basic and most important information estimated from those tracking systems are the player locations, such systems have been applied in professional team sports, such as soccer. In addition to the player trajectories, knowledge of their poses would provide higher-level cues for analyzing sports videos. If the poses of the players can be automatically recognized from team sports videos, more-detailed analysis of the sports plays, such as recognizing actions or understanding the focus of attention, could be achieved. However, there are no robust methods for estimating the poses of team sports players, because the previous techniques only work for restricted patterns.

In this thesis, a novel human pose-estimation framework for team sports videosis proposed; the framework is integrated with the standard tracking-by-detection approach, and it is able to estimate most of the poses of the players in such videos. After tracking either the player windows or the position of the head of a player in a monocular input video, two independent modules are applied: the first estimates the lower-body pose (the locations of the four lower-body joints), and the second estimates the upper-body pose (upper-body orientation and spine pose). An integrated version of those two pose-estimation modules is also presented. Each chapter empirically shows the proposed module can estimate more types of poses than the previous methods, while achieving competitive results to the previous methods.

The framework does not use any temporal information, but only the per-frame player window from a monocular camera. It can use both tracked windows from the videos, or it can use the detected windows from a single image. Moreover, it can disregard the effect of local deformations and local changes in pose (including unknown poses in the training dataset), because it is based on randomized features,

trained by random forests, and obtained from the histograms of oriented gradients (HOG) features within the global (whole- or half-body) region.

Chapter 1 provides an introduction to this thesis. This chapter introduces the methods of data analysis that are currently used in professional sports; the chapter defines the types of poses and their meaning in a team-sport context, and the proposed framework is summarized. Chapter 2 discusses previous work on pose-estimation techniques (classical motion capture and the recent developments based on machine learning). Chapter 3 proposes a lower-body pose-estimation module that uses label-grid classifiers to estimate the locations of the joints on a grid and the grid-structured features (HOG features) within a tracked player window for which the center is aligned with the pelvis location. Chapter 4 proposes an upper-body pose-estimation module that uses poselets-regressors to estimate the location of the pelvis relative to the head center, by using the HOG features in the upper-body region, which the head tracker aligns with the head center. This chapter also proposes a method to estimate the orientation of the body by selecting from five classifiers that are trained independently using only the images that have a spine angle within a shared range. Chapter 5 proposes an integrated version of the modules presented in Chapters 3 and 4; it uses the poselets-regressor twice: once to estimate the location of the pelvis center relative to the head center, and the second time to estimate the location of each of the lower-body joints relative to the pelvis. Chapter 6 presents the conclusions of the proposed human pose-estimation framework and also discusses areas of future work.

Acknowledgements

I would like to thank Yoshimitsu Aoki, my advisor, for his help. His idea of the 3D model-based method for human recognition was of great help to me in achieving a novel and simple human pose-estimation method, and by reminding me to consider future extensions using a 3D human model and a 3D camera model from the monocular video inputs. I am also thankful to my colleagues in our laboratory, Taiki Yamamoto, Junji Kurano, Hirokatsu Kataoka, and Kiyoshi Hashimoto; those from Panasonic Corporation and AVC Networks, Masamoto Tanabiki, Kyoko Oshima, Junko Furuyama, Daisuke Ueta, and Yuji Sato; and also the former Panasonic member, Takuya Nanri. I also would like to thank Dr. Kiyoshi Matsuo, with whom I had many discussions about my study. The American football dataset was provided by the Panasonic Corporation and the Japan American Football Association, and the women's soccer videos were provided by the Keio Soccer Club. I am also grateful for the cooperation of the technical staff of Panasonic and the AVC network members for capturing team sports videos for this research. I would also like to thank all of the members of our laboratory. All of the research being carried out in our group is related to human recognition with vision techniques, so my colleagues have all offered inspiration for my work and helped me to come up with new ideas.

I would like to thank Prof. Eiji Okada, Prof. Masaaki Ikehara, Prof. Hideo Saito, and Prof. Hironobu Fujiyoshi, for agreeing to be part of my defense committee, for their useful comments on my thesis. I also would like to thank other researchers in this field, especially Toru Tamaki and Hironobu Fujiyoshi. The experience of being a lecturer provided by them boosted my confidence and deepened my understanding of computer vision. I would also like to thank a talented young researcher, Yusuke Sugano, who always motivated me to work hard.

I thank my family and friends for understanding that my work was challenging and for offering me their open-minded support. Special thanks go to my mother, Kazumi, who has offered me continual support and inspiration ever since I decided return to Keio University to pursue doctoral studies.

Contents

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

1

Introduction

Human pose provides visual information about the physical states, actions, group activities, emotions, intentions, and various other attributes of humans. Since the human body is articulated, we can consider the skeletal representation of the entire body, or we can consider the behavior of each of the parts. Evaluation of the motion of the skeletal pose is very useful for understanding human actions, and thus researchers have been trying to decode skeletal information from images of humans engaged in various activities. If a skeletal pose can be determined from an image, it can be used to provide mid-level cues to the understanding of that human activity [1, 2]. In addition, the relative positions of the head and the body can be used as cues to predict the social relations between individuals or to predict their attentional focus [3, 4, 5, 6, 7].

In recent years, computer vision technology has been widely used to further our automatic understanding of human poses. Human pose estimation attempts to determine the location of each part or joint of a subject person, from either a single image or from a video sequence. For many years, motion capture systems using multiple cameras and optical markers have been used for kinematic research; however, recent advances in human pose estimation techniques and computer vision have enabled consumer-level applications that are based on poses estimated from monocular images and videos. The appearance of Kinect (Microsoft Corporation, Redmond, WA, USA) drastically changed the human pose estimation techniques that were available at the consumer level. Shotton et al. [8] established a real-time human pose estimation technique; this technique uses as an input a single-depth image captured by Kinect, and it then uses random decision forests [9] to classify each pixel into a part class. In addition, rapid advances in the pictorial structure framework [10] and its extensions [11, 12] have en-

abled more robust estimations of 2D human poses from a single image. If the skeletal pose of a person can be extracted from a single image, it becomes possible to understand the activity of the subject person from a single image [13].

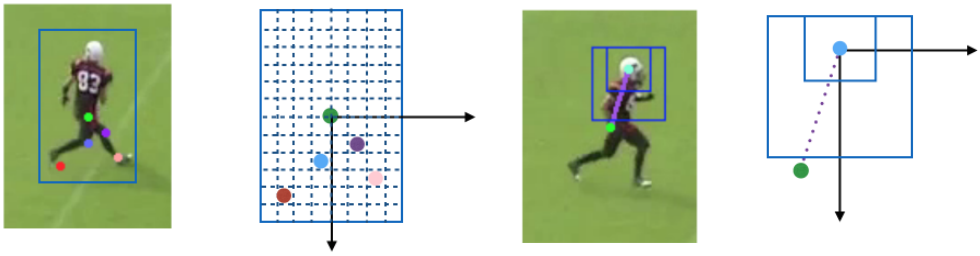
Most skeletal estimation techniques for use with RGB images are restricted to frontal tree-like poses [11, 14], and either the skeletal parts can be identified, or the prior gait cycle can be determined [15, 16]. In addition, computer vision techniques have been used to estimate the relative orientation of the head and body [3, 5, 17, 18, 19], although those studies have considered only standing and pedestrian poses.

The goal of this thesis is to estimate the various types of human poses observed in videos of team sports; the use of such videos to estimate poses has been difficult, because previous methods in computer vision have been unable to deal with the number and variety of poses seen in these videos. In other words, the goal of the thesis is to solve the problem of estimating unconstrained human poses, by establishing a new human pose estimation framework that is explicitly suited to team sports videos captured in sport stadiums. The proposed framework is very simple, and it is integrated with the tracking-by-detection framework, which is the standard approach for tracking the locations of players [3, 20, 21] or the heads of players [22, 23], that uses object detector responses in each frame are used as the confidence for the object tracking in video sequences.

1.1 Proposal

The key idea of the proposed method for estimating human pose is the *use of visual features observed within a global person-appearance window, whose center is aligned by tracking* (see Figure 1.1). The proposed method uses the center of the pelvis (*pelvis center*) or the center of the head (*head center*) to align the global person-appearance rectangles that are used in the proposed joint location estimators, the *label-grid classifier* (Chapter 3) and the *poselets-regressor* (Chapter 4). Since the centers (pelvis or head) of each training image are aligned to a local grid, the estimates of the location of joints need to consider variations in appearance only in a sense relative to the center of the local grid.

The method uses only a few selected dimensions of the Histogram of Oriented Gradients (HOG) [24] features vector observed in each aligned global person window (the blue rectangles in Figure 1.1); random decision forests [25] are used to select the hierarchical feature dimensions. Since there are wide variations in the poses of sports players, I choose random decision



(a) Chapter 3 (classification): Estimated (discrete) locations of joints (knee and foot) relative to the center of the pelvis (green circle).

(b) Chapter 4 (regression): Estimated position of the pelvis center (green circle) relative to the center of the head (blue circle).

Figure 1.1: Key proposal of this thesis: per-frame estimation of the position of joints relative to an aligned center, using the aligned global visual features obtained from person tracking or using detection windows (blue rectangles). (a) Each grid square is classified according to the estimated location of each joint relative to the center of the pelvis; this is done independently and by using random classification forests. (b) A poselets-regressor estimates the position of the pelvis relative to the center of the head; this done by using random regression forests. This framework, which uses the apparent alignment for both training and testing (via tracking-by-detection), allows the selection of more compact and more discriminative visual features (i.e., the feature dimensions used in the histograms of oriented gradients), which can be used to form a robust estimate of the relative location of each joint.

forests to estimate the positions of joints; random decision forests has also been used for per-frame head- or body-orientation classification methods [3, 26, 27], which typically use tracked or detected head or body windows to determine the visual features to be used to estimate the pose. During training, random decision forests can select discriminative features by splitting datasets based on the purity of the random subsets at each node in a supervised manner. With this splitting training procedure, random decision forests can construct hierarchical nonlinear representations that can be used for effectively classifying or regressing the target variables (see details in [25]).

The proposed alignment-based pose estimation strategy was inspired by the use of a detector to align the input rectangle in standard facial-recognition protocols [28, 29]. It is also based on common articulated pose-estimation methods that use part-specific local classifiers with graph-based 2D constraints; these are often called pictorial structures (e.g., the flexible mixture-of-parts method [11]); another common method is to use pose-specific global appearance detectors (e.g., *poselets* [30, 31]). In addition, the proposed framework uses global HOG

[24] features for which the rectangle window is aligned to the local gird; these can be also regarded as *pose-indexed features*[32]. Pose-indexed features are often used with cascaded pose regression [32], which is a framework for estimating the shape (or pose) update of objects from the previous shape estimate; it uses a cascade of random fern regressors to generate 2D shape deformations, such as a face landmarks configuration [33]. In a cascaded pose regression, there is a gradual change in the relative center of the pose-indexed feature space. However, in my framework, a fixed center origin is used for the global person window coordinate, which is determined by tracking-by-detection.

In the proposed framework, the windows used for estimating the pose are acquired by using tracking-by-detection with center-keypoint-aligned detectors; the object tracked is either the person (Chapter 3) whose pelvis is aligned or the head (Chapter 4) whose center is aligned. Although most multi-target tracking studies have used a rigid pedestrian detector to make associations between frames, my framework assumes that the tracked window has already been aligned by using the detector responses or the results of tracking-by-detection. The alignment of the windows enables us to select the features that will be most useful for using random decision forests to estimate the positions of the joints or the direction in which the body is moving. In addition, my framework can work with any method of tracking, and it can separate the pose-estimation procedure from the tracking algorithm.

The proposed framework can extract various types of information about body poses from *monocular* videos of team sports; no previous method can adequately estimate this information, which is as follows:

Pelvis Center Position The head tracker and the poselets-regressor combine to estimate the position of a player's pelvis center. Another option for determining this is to use a person detector that has learned with its center aligned to the pelvis center.

Lower-Body Pose The label-grid classifier (or the poselets-regressor) estimates the locations of the lower-body joints; these are the 2D locations of the left/right knees and the left/right feet.

Upper-Body Pose The body orientation classifiers are conditioned with range of the angle of the player's spine. A line between the head center and the pelvis center is estimated by the head tracker and the poselets-regressor; I will call this the *spine pose*.

Poses during Hard Occlusions between Parts Since my estimation method is based on randomly selected features from the appearance of the entire body, the location of each joint can be estimated, even when the parts form hard occlusions.

Side-view Pose While the popular approaches based on pictorial structures, e.g., [11], or part classifiers [8] have difficulty estimating side-view poses from monocular images, the proposed framework can accomplish this because of the alignment between the visual window and the selection of features within the window.

Many methods for estimating poses from monocular surveillance videos are based on computer vision. However, most of them are constrained to the typical poses of pedestrians, as seen in surveillance poses; for example, see [15, 16]. Most researchers did not use all types of poses to challenge their methods; some methods [11, 14, 34] that use pictorial structures [10] showed good results for only *frontal* or *star-shaped* articulated poses. Pose-estimation methods based on pictorial structures, such as the flexible mixture-of-parts method [11], cannot provide a good estimate of *side-view poses*, poses with hard occlusions, or many types of sports poses (see Chapter 2 for more details).

Computer vision has rarely been used to estimate the spine pose from monocular videos with unconstrained variations in human poses. Unlike the spine poses seen in surveillance videos, where most people are walking or standing, there is a wide variety of spine poses seen in sports players, because, for example, they may bend their upper bodies during defensive moves or stand upright to run at high speed during offensive moves. Estimating the orientation of the body also becomes more difficult for videos of team sports, because it is easier to classify aligned images of pedestrians than it is to classify unaligned images of sports players who have spine poses at many different angles relative to the frame (*spine angles*; see Figure 4.2 in Chapter 4).

If a single generalized framework can allow us to use *monocular* sports videos to estimate those more difficult poses and views (e.g., side views, hard occlusions, spine pose, bent over), then that framework can be used to automatically recognize all types of poses seen in sports videos. However, there have been only a few studies that have attempted this.

There are various methods for estimating human poses based on multiview cameras, such as for surveillance [35] or team sports [36]. While it would seem that these approaches could be applied to any pose, they only work for the star-shaped configurations, because they depend

on either a pictorial structure framework [10] or a part-classification approach, such as those in [8, 11] (in Chapter 2, I will discuss recent methods that use pictorial structures).

In the first section, I will define what I mean by *human pose* in this thesis from the team sports video perspective, and in Section 1.2, I will discuss how it is modeled. In Section 1.3, I will consider the requirements of the vision-based human pose estimation techniques for the analysis of sports data (the background to my framework will be provided in Chapter 2). Section 1.4 summarizes the meaning of the pose of each part in order to clarify how they contribute to the final estimated pose of the whole person. Finally, Section 1.5 provides an overview of the proposed framework and the overall structure of this thesis.

1.2 What Is Human Pose?

In motion capture and kinematics, the human body is typically modeled as a collection or tree structure of parts, in which cylinders are used to represent each body part. This model is sometimes referred to as an *articulated* human body model. For example, [37] uses a 3D model of a human shape and then the parameters of the model are tracked when it is fit to multiple camera images. In those models, adjacent sections are regarded as being connected by a joint. The transformations between two parts (the joints) are defined in the local 3D coordinate system, where the center of the parent joint is at the origin; the centers of child joints can be transformed from a global coordinate system to the local system. By using this relative transformation representation between each pair of adjacent joints (or parts), the global location of each joint can be calculated from the root joint by traversing along the tree structure.

In this representation, human joints have various numbers of degrees of freedom (DOF), depending on the possible rotations of the joint relative to the three orthogonal rotations (roll, pitch, and yaw). While some joints, such as the knee, have only one DOF, others, such as the shoulder, have three. We often represent an articulated model as the sum of the DOF of each of the parts (e.g., 31 DOF). In the graphics or video game industries, this articulated skeleton representation of humans (or human-like creatures or robots) is used to render the animations from the skeletal motions that have been manually designed or from data obtained by a motion capture system. This model is frequently used for articulated robots.

Classical motion capture systems use image processing techniques to track the markers on the surface of the subject; these markers are placed at the joints. Since real joints are within the human body and thus cannot be directly observed, three or four markers are placed on

the surface around each joint, and their trajectories are used to represent the trajectories of the actual joint [1]. For about twenty years, motion capture systems have been used for purposes such as animated movies, video games, and kinetic/kinematic experiments. We can use the real-time motion capture results of Kinect or other RGB-D sensors, although they cannot capture precise joint locations but only estimate them to within a centimeter or two (see Chapter 2 for more details).

Classical motion capture methods and Kinect infer the locations of joints, but they do not infer the degree of *twist* between parts. The orientation of the body can be approximated on a plane by estimating the location of the shoulder joints and the pelvis center, but the original estimation framework of the Kinect sensor only infers the locations of the joints.

Another important issue is that variations in the shape of the human body must be modeled in order to capture a pose from color images or depth images without using markers. In order to use 3D point cloud data obtained by a scanning system to measure the shape of a person, statistical shape models [38, 39, 40, 41] are usually fit to the data. Markerless motion capture [42, 43, 44, 45, 46, 47] can also be achieved by fitting a 3D model to a silhouette image or to edge-based features that (are assumed to) represent the shape of the person. These methods use a 3D body model that has almost the same shape as that of the subject. Even depth-based methods, such as the Kinect method [8], depend on the use of random forests to select the body shape features. Hence, abstracting the contour or the shape of a person is important for markerless shape-based pose-estimation methods.

Dealing with variations in clothing is another key to achieving markerless shape-based pose estimations. In team sports, most players wear simple, close-fit clothing of similar colors. For this reason, it is easier to use visual features to estimate human poses in sport videos than it is to estimate them in general surveillance videos. For this reason, my previous work in motion capture used the foreground region, contours, or other edge-based features to acquire the features for representing body shapes [48]. This contour-based feature approach is based on the same idea for detecting object features that is used in HOG for detecting the presence of people [24] and the deformable part models [49] for detecting the presence of objects (these models use HOG to represent both the local and global appearances). These approaches [24, 49] ignore the color and calculate the orientation histograms from (gray-scale) edge images, in order to learn the variations in the contour-like global appearance of the target object class. In

[1] A joint in an articulated model can be regarded as a functional joint, because the real (anatomical) joint is at a different location and does not always match the exterior behavior.

a previous study, head and body orientations were estimated by integrated tracking techniques (see Chapter 2).

1.3 Motivation: Necessity for More-Detailed Computer Vision Sensing of Human Data and Data Analysis for Sports

Following the current trend in the analysis of big data, vision-based sports data sensing, analysis of measured data, and manually tagged higher-level data are being used for tactical analysis by professional and Olympic sports teams. These data include basic information for sports plays, such as 2D player trajectories on the field, manually tagged action information (e.g., ”shoot,” or ”pass”), and other higher-level group tactics (e.g., ”formations” and ”one-two pass”).

Although in some soccer and basketball stadiums, player tracking is performed automatically with multi-camera tracking products (e.g., TRACAB [50] and SportVU [51]), tracking players with videos is rarely performed. Moreover, in American football or soccer, tagging of the action or even the trajectories of the players is not automatic, but is performed by humans. For example, a Japanese sports data analysis company, Data Stadium, analyzes the high-level meta information for various professional sport teams and broadcast companies, and these labels (such as ”shot,”, ”pass,” and other various types of shots) are applied *manually*. This manual labeling of actions is expensive and requires the labelers to have a professional level of knowledge of the sport. For this reason, technologies enabling the automatic labeling of player actions or trajectories using computer vision techniques will be in high demand in the sports industry; there is an existing demand to be able to analyze digital sports data without the need for labor-intensive labeling of the input data.

In a study that used computer vision techniques to explore the processing of social signals, Bazzani [52] stated, *”The automatic analysis of data is becoming mandatory for extracting summarized high-level information (e.g., John, Sam and Anne are walking together in group at the playground near the station) from the available redundant low-level data (e.g., an image sequence).”* Although that study [52] was written from the point of view of surveillance, the analysis of team sports videos faces the same challenges. However, most methods of recognizing sports plays and most data-mining techniques use only the trajectories and locations of the players and do not use information about their poses [53, 54, 55]. Hence, because it can interpret appearances in terms of postural changes, the proposed framework will enable

the pose-based analysis of team sports. Since sports players tend to vary their pose far more than do people in surveillance videos, we cannot easily use the same approaches for them as for more common group activities (see, e.g., [5, 56]). For this reason, it is necessary to use videos of individual players in order to produce the appropriate pose and activity labels, prior to labeling group activities in sports videos. One of the biggest motivations for this study is the need to estimate the body poses of players in team sports.

In the field of computer vision, action recognition using skeleton information estimated with human pose estimation or key-pose detector as features has been studied with RGB videos alone [1, 2, 57] and with RGBD videos [58, 59, 60]. The RGB-based methods have also begun to achieve good results for action recognition, although they are unable to recognize complex actions or large varieties of human poses, since they simply classify the semantic class of the simple action (e.g., ”jumping” or ”punching”) with only constrained camera views. If Kinect-like human pose information can be extracted even from RGB videos, we can attempt to capture and analyze natural human actions from team videos. Although previous human pose-estimation techniques for a single image with pictorial structures such as [11, 12, 61] have obtained good results for frontal human poses, they are weak when estimating *side-view* and *parts-occluded* skeleton poses; note that in soccer, American football, and many other team sports, side views of players are often captured. This limits the usefulness of the previous methods when it comes to analyzing videos of team sports. Hence, this thesis proposes a new approach to estimate the skeletal pose from team sports videos as well as estimate the positions of joints even when the parts are occluded or the person is viewed from the side.

1.4 What Do the Partial Poses Contribute to Our Understanding of Team Sports?

In order to strengthen the contributions of this study, this section will briefly review the meaning and interpretation of each pose (or motion) of human parts that were identified by the team sports analysis of the proposed method. Note that I will only consider this from the perspective of sports action; I refer the reader to [62] for the social signal perspective.

Figure 1.2 summarizes the meaning of each pose identified for players in team sports. The proposed framework regards the *pelvis center joint* as the center of the entire body and hub connecting the upper and lower halves of the body. The following subsections will discuss the meaning of each of these poses in the context of team sports.

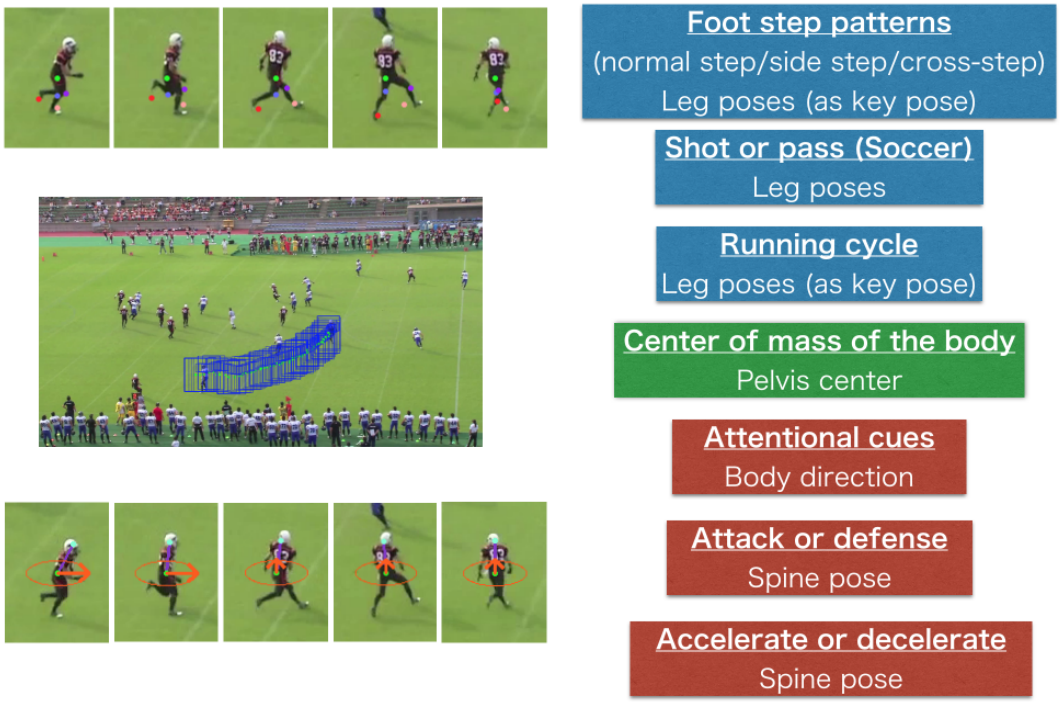


Figure 1.2: The meaning of each type of pose in the context of team sports. In the left column, the tracking summary and the five key frames of this sequence are shown, with the estimated body pose information indicated in red. In the right column, blue rectangles show the pose types (underlined) and the corresponding pose of the lower body. Red rectangles show the types of upper-body poses. The green rectangle shows the most common pose type (pelvis center).

1.4.1 Leg Pose

1.4.1.1 Key Pose during Running

Running is a rapid cyclic action of the two legs. In team sports, players primarily run forward, but they sometimes run left, right, or even backwards, while keeping their (upper) body frontal to the opponents, goal, or some other target that requires their attention. If the leg pose is observed in a video, we can recognize the state change of the foot steps (e.g., normal-step, side-step, and cross-step) during running, and thus we can classify the action by using the leg poses as features.

1.4.1.2 Shot and Pass Actions

In soccer and some other sports in which the leg pose is important, the skeletal leg parts can serve as direct cues for recognizing the lower-body actions, such as shot and pass. If the lower-body joints can be measured from videos, those lower-body (leg) actions can be easily recognized from the measured information about the pose.

1.4.2 Head and Body Direction as Attentional Cues

The head and body direction of team sports players are attentional cues, and they are important for determining the intention or the coverage area of each player. I note, however, that since the intention of the play can be automatically recognized in this way by the opponents, players may deliberately hide their real intentions.

In the context of surveillance, there have been many proposals for estimating the head and body directions of pedestrians [4, 17, 18, 26]. Although in the surveillance context, the direction of movement and the orientation of the body tend to be aligned when people are walking, those of sports players are often very different, because the players must frequently pay attention to the actions of their opponents.

There are previous studies that have used 2D trajectories of the players, such as motion fields [63], to predict the players' trajectories or the group's common intention. In contrast, the orientations of the head and body show the other kind of attention or intention of the players or they show the same attention or intention of moving directions (2D player trajectories).

1.4.3 Spine Angle as an Indicator of Offensive or Defensive State

The angle of the upper-body offers a clue as to whether a player is in an offensive or defensive state. In general, offensive players tend to stand straight, and defensive players tend to lean forward, particularly in team sports that require running to a goal. In American football, for example, running players, such as wide receivers and running backs, do not lean forward, since their goal is to move toward the goal as quickly as possible. On the other hand, defensive players tend to lean forward, and they move slowly; this position allows them to use more power to block their opponents.

1.4.4 Estimated Pose as the Key Pose for the Mid-level Features and Activity Recognition

The poses in each frame of a video can serve as *key poses* of the motion, or they can show a static condition of a person or group. From a multimedia perspective, the key frame or key pose of a video can be used to extract the most important shot or to segment the video or action into multiple (semantic) segments or temporal components. If we segment a video or an action into several temporal segments [64, 65, 66, 67], the key frame and key pose serve as *boundaries* between the neighboring segments or between the representative shot of each segment. If we use key frames or key poses as the inputs for summarizing or identifying a video or action, the key poses can be regarded as *prototypes* of the video or action [2, 68, 69, 70, 71].

Since human actions consist of motions of various body parts, low-level visual motion features and low-level visual features around the spatio-temporal key point or tracked key point trajectories have been used as cues for activity recognition; examples of this include cuboid features around the (STIP) key points [72], optical flows [73], and dense trajectories [74]. However, if the skeletal poses are estimated beforehand, such as by Kinect [8], the mid-level poses can be estimated more easily, and simpler pose-based gesture can be used as cues for activity recognition. The key pose also assists in the recognition of actions from a single image or from a video [1, 2]. Since the proposed framework can estimate the locations of the lower-body joints, the spine pose, and the orientation of the player's body, these estimated poses will be able to be used as cues for person/group activity recognition or multimedia applications, such as video retrieval or video summarization (this will be discussed as an area of future research in Chapter 6).

1.5 The Outline and Contributions

The outline of this thesis is summarized in this section (it was also discussed in Section 1.1). The challenges and the contributions will be summarized below in Sections 1.5.2 and 1.5.3.

1.5.1 Outline of the Thesis

The human pose-estimation framework proposed herein is a per-frame pose estimator that is integrated with object trackers (the player tracker in Chapter 3 and the head tracker in Chapter 4) of the subject player. After the tracker provides a rectangular window image of the player, each of two modules independently infer the lower-body and the upper-body poses of

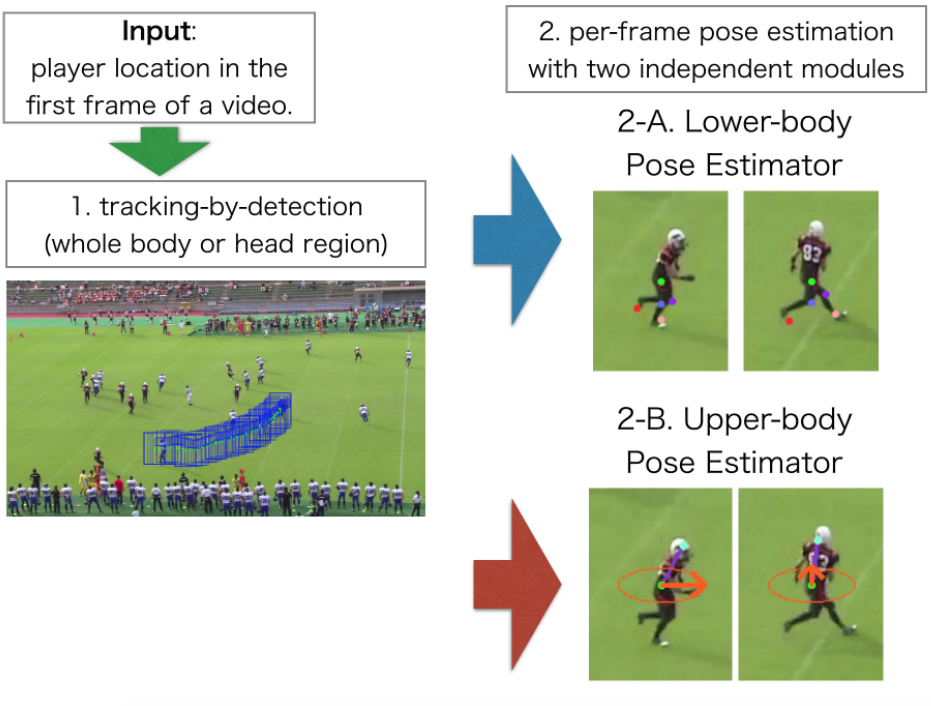


Figure 1.3: Overview of the framework. The left side of the figure shows the tracking part of the framework, and the right side of the figure shows the two pose estimators: the lower-body pose estimator (Chapter 3) and the upper-body pose estimator (Chapter 4). Both pose estimators utilize center-aligned global person appearances in each frame to estimate the corresponding output poses.

the player. The two modules provide the visual features within the tracked window in each frame. The per-frame and aligned whole-body image strategy also enables a alignment-based pose estimation, although here I primarily consider using the tracked windows to obtain an estimation.

Section 1.5.2 describes the target poses seen in some specific target scenes, and it discusses the difficulties in estimating them. Section 1.5.3 summarizes the contributions of this thesis.

1.5.2 Problem Setting: Target Pose and Appearance Variation

There are two main targets of the framework: (1) the 2D positions of the joints in an image, which are used to estimate the skeletal pose of the player, and (2) the body orientation and the spine pose, which is the 2D line between the head center and the pelvis center. That is, this study explores the estimations of the two main variables of each part, which are: (1) the 2D

location of each part and (2) the twist and tilt of each part (these were defined in Section 1.2). The module for the lower-body pose (Chapter 3) estimates the location in an image of each of the lower-body joints, and the module for the upper-body pose (Chapter 4) estimates the orientation of the head and body (output 1) and the spine pose (based on the locations of the head and pelvis centers; output 2).

The proposed framework divides these estimates into two subproblems: estimating the locations of the four lower-body joints (Chapter 3) and estimating the upper-body spine pose and the orientation of the body (Chapter 4). Those two challenges become too difficult to solve when I include variations in clothing for which the visual features are too widely distributed. However, the most important goal in pose estimation is to be able to deal with many types of poses (and appearance) for sport-specific actions. It is not necessary to be able to deal with varied clothing in sports scenarios, because the players in a giver sport wear similar uniforms. Therefore, the primary changes in appearance are due to postural changes during the various actions.

For the above reasons, this thesis focuses on solving the human pose estimation problems by training a sport-specific appearance model with HOG and random forests; that is, the model only knows the postural appearances for a specific sport. For instance, I used only images of American football to train a poselet-regressor, and I used only images of soccer to train body-orientation classifiers. This sport-specific approach allows the pose-estimation model to focus on only the variations in pose and appearance (feature distribution) that occur in the single sport for which the model is used. In the following subsections, I will discuss the two target poses that must be considered when using a sport-specific model.

1.5.2.1 Skeletal Pose Estimation for Nonfrontal Views

The proposed framework focuses on estimating the skeletal pose of nonfrontal and nonrear views, which were not explored in previous studies. As will be summarized in Chapter 2, in previous work, side-view skeletal poses have only been estimated, but the estimation of frontal upper-body poses has been explored using parts detectors and pictorial structures.

The main reason for this is that the focus was primarily on estimating frontal skeletal poses by using pictorial structures that assume that a person can be viewed as a tree structure comprising local parts [11, 61], and with this approach, it is difficult to deal with the various types of inter-part occlusions. The proposed pose-estimation approach is intended to be able to overcome the difficulties with occlusions by using the global and aligned HOG features.

1.5.2.2 Estimating Body Orientation When the Upper Body Is Bent

The proposed framework (Section 4) can be used to estimate the upper-body orientation even when the person is bending over. As I will discuss in Section 2.4, previous estimates of the upper-body direction or orientation have mostly considered pedestrians with an upright upper body. This assumption greatly reduces the variation in the appearance data, and it is much smaller than the variation in my unconstrained setting for the poses of players of team sports. Moreover, note that players often perform sport-specific actions that cause the upper-body poses to vary much more than they do in pedestrians.

On the other hand, previous work on upper-body pose estimation did not consider estimates of the body direction. The reason for this is that they mainly assumed that people would appear with frontal poses in most high-resolution images on the Web or TV (See Section 2.4 in Chapter 4).

1.5.3 Contributions

The contributions of this thesis are summarized as follows:

- Estimation of the *relative* location of human joints, using the center-aligned global person appearance (HOG features from the whole body window or half body window); most previous work relied on local part detection [11] or part classification [75].
- Spine pose estimation using a poselet-regressor and from the upper-body HOG appearance, with its center aligned to the head center; this is done *without* using the pictorial structures framework, which has been the standard strategy for human parsing from images.
- The first method for estimating the location of joints in side-view poses during running for monocular videos. This is critical for estimating the poses of players of team sports, since much of the time, they are running in the direction of the goal.
- Estimation of body orientation, even when the spine is at a steep angle; previous work considered only the body orientation of standing pedestrians.
- The proposed center-aligned visual feature space with feature selection is robust against *occlusions* of parts, because the space captures the selected global appearance for estimating the relative positions of joints with random decision forests.s

- Detailed estimations of human pose, using *only* monocular videos; previous methods include multiview and part-based pose estimation methods [75, 76]. With this ability, the proposed framework can be used for the analysis of sport video archives.

In addition to the above contributions, the proposed framework has the following characteristics for both the upper-body and lower-body estimators:

- Per-frame estimation of the position of each joint is done independently using the appearance features for the whole body (global HOG features are selected by training weak classifiers of random forests). Whereas the structured prediction approaches that use pictorial structures need to infer or search for the best configuration of the joints, my approach does not rely on inference from pictorial structures as priors, because it depends on the aligned appearance of the whole body while it is being tracked, and it uses the poselet-regressor to estimate the pelvis center.

- Training of the estimators must be performed only once for any given type of team sports clothing, providing that the appearance of the people in the test videos is sufficiently similar to that of the people in the training images. The HOG features and the random forests also contribute to this by creating local deformation invariance and by ignoring the uninformative features during feature selection.

- Rough alignment of the tracker-based center and the HOG features make the method robust against some degree of drift of the tracker. This enables us to use the tracker results to determine the alignment-based skeletal pose and to estimate the orientation. The computational cost is also limited due to the dependence on the tracking-by-detection results.

These characteristics will be discussed again in Chapter 5, along with the presentation of further experiments using the integrated method. Naturally, the above characteristics (or my novel approach of estimating the pose of moving people) also have some disadvantages and limitations. I will discuss these along with the experimental results in Chapters 3–5. In Chapter 6, I will discuss directions that can be pursued to correct these limitations, but that are beyond the scope of this thesis.

2

Related Work

This chapter will discuss work related to that presented in this thesis. In Section 2.1, I will provide a brief overview of classic motion capture systems, which depend mainly on image processing and multiview geometry techniques. The early systems depended on optical-marker tracking, and they were constrained to people wearing black costumes or to chroma-keying in order to simplify figure-ground segmentation. Later, techniques were explored that were based on fitting human-shaped models to a human foreground region [48, 77]. These techniques did not use machine learning procedures to directly train the poses that were output from the visual features.

From Section 2.2, I will summarize recent advances in machine-learning techniques for estimating human pose; these can be viewed as the foundation of the proposed framework. Section 2.3 discusses previous papers on estimating skeletal joint positions that are related to the proposed lower-body pose-estimation method (Chapter 3). Section 2.3 discusses previous papers on estimating head and body directions that are related to the proposed upper-body pose-estimation method (Chapter 4) in Section 2.4. The integrated method (Chapter 5) is related to Sections 2.3 and 2.4.

There are two important considerations in the following discussion on related work. The first is to note that most of the previous work for estimating joint poses employed part detectors that output configurations of graphical models (e.g., [11]), or they used pixel-wise segmentation with a part classifier (e.g., [8]). Instead of performing these per-pixel segmentations or part detections in a sliding window, my work builds a visual feature space within the center-aligned global person appearances. This enables a one-shot estimation with the per-frame global HOG

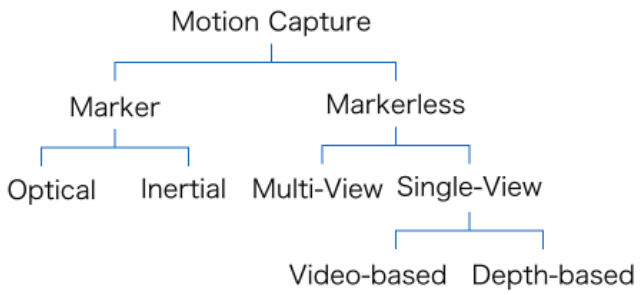


Figure 2.1: Categories of motion capture and human pose estimation.

appearance using random forests, while the previous methods must perform per-pixel classifications with a sliding window.

The other important consideration is that the proposed framework was inspired by previous methods for estimating the *head and body orientation* that use the global appearance provided by person detectors [3, 26, 27]. The global pose representation is the same as the poselets detector [30], but the proposed label-grid classifier and poselets-regressor assume that a continuous global appearance feature space with an aligned center is provided by the tracker of the central joint. In my framework, the center-aligned global windows in each frame are provided by the person tracker (Chapter 3) and the head tracker (Chapter 4).

2.1 Classic Human Motion Capture Techniques

This section briefly discusses the history of motion capture techniques and the estimation of human pose in (mainly) the field of computer vision.

Human pose-estimation techniques can be categorized as follows:

- Marker-based versus markerless motion capture using multiple cameras.
- Multiview versus single-view cameras.
- Video based (RGB image) versus depth based (depth image).
- Image processing and multiview geometry versus machine learning.

These pairs can be also summarized as a hierarchical tree, as shown in Figure 2.1. Although it does not include the image processing/ machine-learning pair, Figure 2.1 shows a bird's-eye view of human motion capture techniques [1].

Commercial marker-based motion capture systems (such as the products of VICON [78] and OptiTrack [79]) depend on the 3D reconstruction of the path of markers attached to the subject. They also require the wearing of black suits to make it easier to track those markers with classic image-processing techniques with multiple IR cameras. Although they can track the 3D positions of the markers precisely, this technique can be applied only to a person wearing a black suit in an experimental room, and it requires a long preparation time to mark the suits. Moreover, the marker-based motion capture systems tend to be expensive, because multiple infrared cameras are needed. They are sometimes combined with inertial sensors (such as the products of Xsense [80] and SYNERTIAL [81]). The motion capture systems that use inertial or gyro sensors can provide clear motion data without requiring a studio or particular clothing. However, those systems are still expensive, because the inertial sensors are precision instruments.

Markerless motion capture using multiple cameras (e.g., Organic Motion [82]) has also been used for the same purposes as that for maker-based motion capture. Although this method does not require markers, it can be only used with a monocolor background to the capturing area (this is also called *chroma keying*). To estimate the 3D skeleton of a subject in each frame, this method fits a silhouette of the 3D human-shaped model onto the foreground region that was extracted with background subtraction of the chroma key monocolor. The human-shaped model represents limbs with cylinders, and the model is fitted to the foreground contour of the subject. The fitting error is viewed as a likelihood, and the human skeletal pose is tracked by a Kalman filter or by a Bayesian tracker.

Both marker-based methods and markerless methods require some restrictions on the images captured from multiple cameras. For marker-based methods, monocolor suits are needed if the marker position is to be tracked using only classic image-processing techniques. On the other hand, markerless methods require a monocolor background to calculate clear and accurate foreground regions. Since their target is a real-time and precise capture of motion, they use monocolor suits or a monocolor background. However, this restriction had been the main issue

[1] This figure is taken from the tutorial "Motion Capture from RGB-D Camera" in the CVPR 2014 tutorial "Towards Solving Real-World Vision Problems with RGB-D Cameras " (http://www.iai.uni-bonn.de/ gall/tutorials/visionRGBD14.html).

preventing motion capture in real environments. See [48, 77] for more details on the classic motion-capture techniques that do not use machine learning.

There are also markerless motion capture methods that use a complex background with generative tracking techniques and a human-shaped model for evaluating *monocular* RGB videos [42, 43, 44, 45, 46, 47, 83]. These approaches depend on a sampling-based tracker, such as an annealed particle filter [83], to sample the candidate silhouettes from the 3D human-shaped model, and they use the distance between the (sampled) model and the foreground in each view as a likelihood for the annealed particle filter. Since a 3D human-shaped model typically has about 20–30 DOF for joint movements, it is necessary to sample many silhouettes in each frame or to use an iterative method so that the cost function converges in the large search space. This is because the annealed particle filter requires a long time to estimate the pose in the next frame. See [44] for a comparison of the tracking-based markerless techniques.

Since 2010, Kinect and other RGB-D sensors have become available [84], and we can now perform markerless motion capture techniques or tracking-based techniques *without* monocolor environments or suits and *in real time*, because the data captured by the depth sensors can be easily segmented into the foreground human region and the other regions. For example, [85] uses three handheld Kinect sensors to acquire the foreground point cloud of a subject, and performs markerless motion capture (this method also performs camera pose estimation and computes the registration between the point clouds of each view). At the same time, we also have OpenNI [86] and Kinect for Windows SDK [87], which use machine learning techniques to capture the human pose in real time. These methods and products suggest that we can easily capture the skeletal pose with inexpensive RGB sensors in real time, and we can use Kinect to capture the depth data.

However, Kinect can only capture depth data in indoor environments and in regions near the sensor (within 5 to 10 meters). For team sports videos, we also need to capture image data in outdoor fields, and these cannot be captured by the Kinect depth sensor. Moreover, even in indoor environments, such as basketball courts and hockey rinks, Kinect cannot capture the depth data of the players, since the distances are too great for it to capture a broad region at one time.

2.2 Previous Work Using Machine Learning: Overview

This section provides a brief overview of the human pose-estimation techniques that use machine learning.

In Section 2.3, I will discuss papers that are related to the lower-body pose estimator (Chapter 3), and in Section 2.4 I will discuss those related to the upper-body pose estimator (Chapter 4).

Current state-of-the-art human pose-estimation techniques that use machine learning have the following disadvantages:

- Most of the body parts must be shown in the image. Hard occlusions of parts makes it difficult to estimate the human pose, because a tree-based representation of human parts is assumed, and it has difficulty handling a large number of parts.
- For the upper-body pose estimation, previous methods that used pictorial structures [10, 11, 14] relied heavily on detection of the arm in order to estimate the spine pose. Hence, the spine pose cannot be estimated if most of the arm is occluded, because arm parts are hard to detect with part detectors.
- They can estimate the head and body orientation only when the person is standing, because the previous methods [4, 17, 18, 26] only dealt with (standing) pedestrians. For this reason, it is difficult to use the previous methods, which are for predicting the orientation of pedestrians, to estimate the body direction of a bending player.
- They cannot estimate the locations of joints in side-view poses. The pedestrian skeletal pose-estimation methods [15, 16, 88] can only estimate the side-view poses by using trained priors and assuming that the subject will only appear in a side view for a single gait cycle.

Since consumer depth sensors, such as Kinect, are intended to capture the natural pose or behavior of a person, they are customized for estimating the front of the body of the person who is interacting with the display, the PC, or some other device. Thus, they are not good at estimating side-view poses, because this view is not one of its targets.

As I will discuss in the next section (Section 2.3), previous human pose-estimation methods that use the popular pictorial structures framework can only estimate frontal and star-shaped poses. The estimation of side poses has barely been explored in the field of computer vision.

The proposed method will disregard pictorial structures and estimate side poses by using a label-grid classifier or a poselets-regressor.

As I will discuss in the following section (Section 2.4), human body orientation methods are restricted to estimating the pose of standing pedestrians. They cannot deal with bending poses. Moreover, previous upper-body pose-estimation methods that used pictorial structures require a frontal view, because they are mainly targeted at estimating the skeletal 2D pose of a person with a frontal pose in photos, web images, sit-coms, or movies. In these images, people tend to appear to the camera in a frontal direction.

2.3 Previous Work Related with the Proposed Lower Body Pose Estimation Method

First, I review human pose estimation methods using the classical silhouette-based template matching technique for team sport videos (Section 2.3.1), which is not robust and is just for graphics visualization. Next, I review two types of human pose estimation techniques: human pose estimation from a single image using pictorial structure (Section 2.3.2), and motion model regression (Section 2.3.3). Finally, I review instance template detection techniques such as poselets [89] and Exemplar-SVMs (Section 2.3.4).

2.3.1 Exemplar-based Pose Search Methods

Germann et al. [90] proposed silhouette-based database matching methods for soccer players. These methods first construct an exemplar-pose database with silhouette images of players. At test time, their method first finds the most similar silhouette from the database in each frame, and then applies optimization through multiple temporal multiple frames. The resulting poses are not very accurate because the exemplars cannot include every type of human poses, and are sensitive when extracting the silhouette via background subtraction. Moreover, this approach involves a high computational optimization cost.

2.3.2 Single Image Pose Estimation Using Deformable Part Models

The FMP [11] is the extension of the deformable part models (DPM) [49] that are used to estimate the pose of the human by inferring from the best part configuration. DPM was originally a weakly-supervised model using latent support vector machines (latent SVM) to learn

the appearances and locations of each part detector automatically from the labeled whole object window. On the other hand, FMP is a supervised version of DPM and uses mixture-of-parts to represent the discrete changes of each part appearance. FMP [49] accurately estimates frontal poses of subjects opening their legs and arms. However, FMP cannot precisely estimate the poses that have feet and arms occluded, because it depends on the part-detection scores and depends on the tree-graph where each subnode (arms and feet) is widely open.

For this reason, pictorial structure model, such as FMP, tends to estimate the pose of a person who looks at right or left and even frontal poses incorrectly, because it is hard to detect each arm or leg part in those poses owing to their ambiguous and incomplete part appearances. Moreover, it is difficult to model the configuration with one tree-structure model for frontal, side and bending poses. The model needs to learn those models separately.

More recently, Poselets-based [89] pictorial structure approaches have been studied [12, 61]. Although those methods overcome the weakness of FMP by representing the relationship between parts using poselets (which are larger parts than parts of FMP), they are still poor at hard occlusion cases because they still depend on the pictorial structure. A multi-camera approach [75] helps to deal with part-occlusions, but even this methods is not able to tackle with side running scenarios where part-occlusions occur frequently.

There is also a method involving an occlusion handling scheme using an occlusion detector and part-based regression [91]. While this method provides good results with small occlusions between parts, it also cannot estimate side poses because it still depends on the pictorial structure.

2.3.3 Motion Model Regression and Pose Manifold Methods for Pedestrians

If the human pose model only includes the pose types within one action class, such as walking or swinging a golf club, pose estimation can be solved using regression techniques with fixed-view training images. The tracking method using Gaussian Process Regression [15] is popular for learning a (latent) 3D pose manifold from cyclic pedestrian images from one camera view or multiple views.

For pedestrian pose estimation, Gammeter et al. [88] proposed a people tracking and pose estimation method for pedestrians using Gaussian Process Regression. Rogez et al. [92] proposed a per-frame pose estimation of human pose estimation based on Random Decision Forests [93] and pose manifolds of a gait sequence with HOG features. [92] is close to my method in Random Decision Forests for pose classification. However, their Random Decision

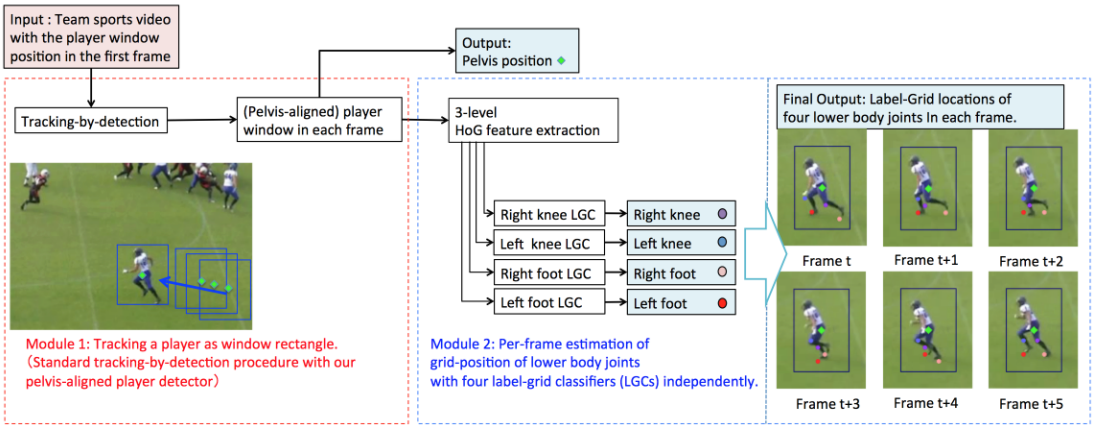


Figure 2.2: Flowchart of the proposed lower body pose estimation framework.

Forests class is based on camera views and gait manifold cycle, while my Label-Grid class is the grid of HOG features. Additionally, they do not predict the joint positions precisely because they just find the most similar walking pose exemplar on the gait manifold. These methods only investigated the pose distribution of walking people. They can learn the latent transitions between the poses of pedestrians, but cannot afford to include every types of poses.

2.3.4 Poselets and Exemplar-SVMs

My pelvis-aligned detector and Label-Grid classifier are both inspired by the poselets [89] framework. Poselets are the *detector* of one specific *pose* of a middle-level human part detector, which can be learned from training images with the same aligned pose and same scale images but from different subjects (e.g., upper body Poselets with their arms crossed).

Exemplar-SVMs [94] is an object detection method using per-exemplar detectors. It detects exemplars using each exemplar-SVM to detect multiple appearance types of an object class. Exemplar-SVMs separately train each training instances independently by regarding the other all positive examples as negatives. By using all exemplar-specific SVMs, this frameworks not only can detect per-exemplars but can even detects sub-category level with exemplar-SVMs with in-category exemplars (e.g, with 100 frontal car exemplar detectors).

On the other hand, my pelvis-aligned detector learns multiple scales and poses of an object class altogether. This one-detector solution makes it easy to integrate with a tracking-by-detection scheme, while the goal of Exemplar-SVMs is robust object detection even with only one image using multiple SVMs that know the hard-negatives.

2.4 Previous Work Related with the Upper Body Pose Estimation Method

The estimation of head and body orientation from low resolution videos has been studied for the purposes of video surveillance [17, 18, 26, 95, 96]. The body orientation of the subject, which the proposed framework predicts, has also been used for group activity recognition [5, 56, 97] as context features between interacting people.

There are two main approaches for body pose estimation from a fixed camera view: 1) human pose/orientation estimation from a *single image*; and 2) human body pose or orientation estimation from *videos* based on the position of the person/head *tracker*. Inferring the skeletal body pose or body orientation is quite useful for many applications, such as searching for semantic key poses from TV shows [98], recognizing pose and clothing attributes of a person [99], recognizing the interaction between two people [7, 19], and recognizing the interaction between an object and a person [100, 101]. I will review human pose estimation methods from a single image in Section 2.4.1, and human pose estimation methods from videos in Section 2.4.2.

Another categorization of related work is the human pose estimation for (1) the *skeletal* pose and (2) the head/body *orientation*. The proposed poselet-regressor is the skeletal pose estimator of the (simplified) 2D spine line while the proposed spine-conditioned body orientation estimators are the body orientation estimators. The single image methods in Section 2.4.1 are related to the poselet-regressor, while the tracker-based orientation estimation methods in Section 2.4.2 are related to my spine-conditioned body orientation estimators.

While skeletal pose estimations are mostly performed using a part-detector strategy, the proposed strategy in Chapter 4 employs the global aligned window to estimate the skeletal spine pose. This tracker-based joint location estimation of the poselet-regressor is inspired by my lower body pose estimation work [102] (Chapter 3), which also uses the body tracker for the window alignment and used random classification forests to estimate the lower body joint locations.

2.4.1 Human Pose Estimation from a Single Image

There are many papers that try to estimate the skeletal body pose from a *single image* [7, 19, 30, 61, 98, 99, 100, 101, 103]. These approaches mainly estimate the *frontal* 2D skeletal bone of the subject and the regions of each part of the body.

The most popular approach to skeletal human pose estimation uses pictorial structures [10, 104], where the body parts configuration of the person in the image is represented as a graphical tree model of body part region appearances. After the success of the Deformable Part Models (DPM) [49], the pictorial structure framework for people detection is extended with DPM to jointly detect the part locations with structured prediction, such as a flexible mixture-of-parts model (FMP)[11].

FMP [11] is robust for the frontal pose images because it employs a tree-structured graphical model of part detectors, which are trained as the mixtures of the part appearances from the training dataset using k-means. While FMP [11] gives very good results for frontal human poses, it cannot infer the partially-occluded side poses accurately because the tree-structure of the graphical model does not appear when the arms and legs partially occlude each other. In team sports videos, there are many side-view pose appearances in which it is difficult to estimate the parts locations using part-based models.

Part-model approaches find it difficult to estimate the pose: (1) when multiple parts are occluded; (2) when the image is low-resolution and the parts have similar appearances, making it hard to discriminate each part correctly; and (3) when the person is in non-frontal side poses, which are hard to model with tree part-models of pictorial structure.

There are also per-frame classification or regression methods for estimating each part location from a single depth image [8, 105]. Additionally, there are part-model approaches using multiview images [75]. In contrast, my approach does not use depth image or multiview inputs, but only use monocular RGB images.

Poselets: Detection of One Specific Pose.

Bourdev and Malik [30, 31] proposed human (partial) pose detectors, poselets, which can be also regarded as a human (partial) pose silhouette detector. The poselets have been also used for the middle-level parts for human pose estimation methods with part-based models [61]. Pishchulin *et al.*[12] proposed the poselet-conditioned pictorial structures approach, which uses the poselets as a mid-level representation of multiple body parts. While pictorial structures using poselets [12, 61] can cover the pictorial structures with only local part detectors [11], classification still depends on the pictorial structures and cannot cover many types of articulation (with only a few specific poselets). Particularly when the arms or legs are partially occluded

or hidden behind other parts, we need more poselets (exemplars) to represent those person appearances. This makes it more difficult to represent a huge number of part configurations, even when occlusion or part-disappearance occurs.

The poselets can be also used for key-frame extraction such as in [2] for activity recognition via key-frame responses. However, poselets cannot detect detailed or in-between poses because poselets are discretized key-pose exemplar detectors (but they can detect key-poses as attributes or actions, see [99, 103]). Maji *et al.*[103] proposed multiview poselets for the purpose of single image action recognition from the detected pose with action-specific poselets.

Pose Regression.

Recently, appearance-based regression of some kinds of human poses has been studied [106, 107]. Classic approaches for estimating skeletal poses, such as [15, 106, 107], try to train regression models (with low dimensional latent variables) of typical human movements for individual activities (e.g., walking, jumping). [15, 108] estimated the joint locations of a target walking person using a fixed side-camera view in each single frame of a video. In these papers, cameras are set to capture the person from side views on the road or street so that people can be captured in only side-view poses. This side-view camera setting is often used in gait recognition [109]. In contrast, the proposed method can estimate the body orientation and the spine pose from any camera view, because the poselet-regressor learns the various types of human upper body pose appearances from all camera views using only one model.

Conditional regression forests [105, 110], which divides the visual feature space into each-view spaces or some other subcategories, is the closest approach to my proposal. For facial images, when the view of the face is restricted, visual patterns of fiducial points become smaller because the facial parts cannot move so much. However, the whole-body appearance has a wide variety of patterns (body orientation or spine angle, in my case) even when the camera view is restricted. Also gaze direction estimation from the eye-region appearance is explored with conditional regression forests conditioned on the head pose [111].

For non-articulated objects, regression-based pose estimation with Cascaded Pose Regression [32] can be done. However, articulated-pose estimation from RGB images has not been explored with Cascaded Pose Regression, while some depth-based methods have proposed the regression of part locations [8, 105]. Recently, articulated human pose regression methods using Convolutional Neural Network (CNN) have been proposed [112, 113]. However, those papers perform experiments on the datasets that only contains frontal poses or star-shaped

poses, such as Buffy dataset [114], FLIC dataset [115]. Hence, we cannot know whether those methods can also work with the side-view poses or part-occlusion poses in team sports videos.

2.4.2 Human Orientation Estimation from Low-Resolution Videos

Head and body orientation estimation approaches during *people tracking* have been proposed, mostly for surveillance videos [3, 4, 17, 18, 26, 116]. Another popular scene setting is the frontal video of automobiles [95, 96]. These methods train scene-specific or clothing-specific head or body orientation classifiers, which typically classify the horizontal eight directions into eight classes, by combining those classifiers with the head or body trackers to jointly estimate the orientations and the location with filtering techniques. They mainly estimate the head and body orientations of *pedestrians* and cannot deal with poses where the subjects are bending their upper bodies. While I would like to review only body orientation estimation methods, since the proposed method estimates the body orientation in each frame, I will also review head orientation estimation methods. The latter are closely related and adopt very similar approaches to classifying the direction, and my method uses the same head tracker as used in this work [3, 4, 17, 116]. Those head orientation methods are also combined with body orientation estimation [18, 117, 118].

Benfold and Reid [116] proposed the first approach that adopts feature-selection for learning a per-frame head orientation classifier using randomized trees with a color feature. Benfold and Reid extended [116] in [3] by introducing a HOG features [24] with a color feature to create a robust head pose classifier and proposed a multi-target tracking scheme using a HOG-based head detector and an optical-flow tracker for surveillance videos. This per-frame classification with tracking-by-detection proposed in [3] has been a standard approach for recent head or body orientation estimation methods [18, 26, 119].

For team sports videos with low-resolution settings, Hayashi et al.[119] proposed a head and body orientation estimation and spine pose estimation of American football players in videos. This work performs head and body orientation classification independently with tracking-by-detection with a player tracker, as well as head detection within each frame. In sports videos, it is easier to train a robust head orientation classifier than training it for surveillance videos, because the visual appearance of the head region is similar between different players wearing similar uniforms, especially helmeted players in the experimental American football videos. However, since head appearance is versatile in other team sports (e.g., basketball or soccer) and in surveillance videos such as [3], the head orientation classifier needs higher dimensional

or discriminative features than for sports players. Conversely, body features in sport videos have more pose variations than in pedestrians. In this sense, tackling various types of postural appearances of sports players is the main focus and contribution of this work.

Schulz *et al.* [95] proposed a joint head pose estimator and head localizer for pedestrians for the risk assessment of car drivers. Later, Schulz *et al.* [96] proposed a sequential Bayesian tracking extension of [95] with a particle filter. In [96], they use a head pose classifier result as the per-frame likelihood of a particle filter, and jointly predict and update the head location and head pose by tracking a pedestrian on video. Benfold *et al.* [17] used conditional random fields to train the head pose estimator of pedestrians in an unsupervised manner in a new video scene. These papers make use of the temporal transition constraints on head location and also the temporal continuity of the head orientations of pedestrians via filtering. While the head locations can exist only in the upper region of the detection window because pedestrians are always standing and walking upright [95, 96], the head locations of sports players have larger variation because they often bend their bodies and sometimes dive into opposing players.

Compared with the Town Dataset setting of Benfold *et al.* [3, 17], players in team sports videos have much more random transitions of the head/body orientation between frames and it is harder to assume the smoothness of the head/body orientations between consecutive multiple frames. For this reason, I will investigate *per-frame* classification of body orientation in this paper without using a temporal connection while the head tracker is performed with a Kalman filter, which assumes temporal smoothness of (only) head locations.

Cheng *et al.*[117] proposed a temporal framework for joint estimation of body orientation and location of the subject pedestrian using a particle filter with sparse codes of multi-level HOG features. Baltieri *et al.* [26] proposed body orientation estimation for pedestrians using the mixture representation of Extremely Randomized Trees classifiers. These methods have only been tested for standing pedestrians, while my method covers even non-standing bending poses owing to the flexibility of the poselet-regressor. In addition, this work and our previous work [119] estimate the upper body orientation using only upper body HOG features, while previous papers estimate the (whole) body orientation using the appearance of the whole body.

2.4.2.1 Joint Estimation of Head and Upper Body Orientation from Videos

Chen *et al.*[118] proposed joint tracking of head and body pose in surveillance videos. They used a particle filter to jointly estimate the head and body orientation combined with the movement direction. Later, in [18], they extended their work to the semi-supervised learning setting

with their own kernel learning scheme by learning the relationship between the parameters governing head orientation, body orientation, and movement direction.

Different from [18] and [118], which leverage the assumption of the lower velocity of pedestrians and combine the velocity with the body orientation, the proposed method tries to deal with the upper body appearances of sports players at higher speeds, which are already aligned by the head tracker without needing to make a connection between movement direction and body orientation. For this reason, good alignment by the head tracker is key in the proposed method, because the proposed method does not depend on the relationship or a prior from the movement direction and perform only per-frame body orientation classification.

Poselets Detector as Pose Category Classfier.

[30] proposed human (partial) pose detectors, poselets, which can be also regarded as a human (partial) pose silhouette detector. Poselets have been also used for the middle-level parts for human pose estimation methods with part-based models [61]. The poselet framework has an advantage in creating detectors for side-view poses, which are hard to deal with for part-based whole-person models [11].

Poselets can be also applied for key-frame extractor such as [2] for activity recognition via key-frame responses. However, poselets cannot detect detailed or in-between poses because they are discretized detectors. Poselets can only detect discretized rough poses (See Figure 2 in [2] for poselets examples), while poselets can detect side-view poses that occur often in activity recognition videos.

3

Lower Body Pose Estimation with Label-Grid Classifier and the Center-Aligned Player Tracker

In this chapter, I propose the joint locations estimation method integrated with the standard tracking-by-detection technique using the whole body detector. This work has previously published as [102].

The method of this chapter does have a restriction that it can only track and estimate the lower body joints of the standing players because of the dependency to the whole body detector trained from standing players images. In the next Chapter 4, this restriction will be resolved by replacing the tracking module with the combination of the head tracker and the poseletes-regressor.

3.1 Introduction

Video-based player tracking has drawn interest in computer vision. Since video-based object detection and tracking techniques have shown rapid improvement [120], the applications of tracking *team sports players* are becoming increasingly more attractive for professional sports. Body pose information would be a middle-level feature for classifying the detailed action of each player. Like activity recognition methods [1, 121] using a pose estimated by single image pose estimation techniques [11] suggest, the pose (joint location of the person in 2D or 3D) can be a stable and clear cue for detailed and fine-grained activity recognition. While action

recognition methods using spatio-temporal local features [122, 123] can estimate the semantic action class (e.g., running or standing) of the player, inner-class action difference (e.g., how widely the person moves his or her legs while running) cannot be easily estimated. Even for semantic action recognition, an action classifier using a pose feature (annotated joints) performs better than one using low-level feature (dense flow) as [124] illustrated in their experiments.

In particular, using the lower body pose (or lower body joint positions) from team sports videos would be a new way of recognizing each action of a player in detail. Since *running* is the very basic and most frequent action in all team sports, leg movements are one of the most important cues for recognizing player actions. For example, when the subject player is running, the joint pose can precisely measure the number of steps of the player, which can be the cue for classifying the step types (normal step or cross step), and which can be the contact point with the foot in soccer. However lower body pose estimation has rarely been investigated in computer vision.

Human pose estimation from *video* is still an open problem in computer vision while depth-based methods using RGB-D sensors has already been realized as a highly robust system [8]. I cannot yet estimate the pose of the sports players in all types of sports videos, while pose estimation of some limited periodic actions (such as walking) has only been solved using non-parametric regression techniques [15]. On the other hand, the frontal human pose of sports players can be robustly estimated from an image with methods using part detectors and pictorial structures such as the flexible mixture-of-parts model (FMP)[11]. However, part-based methods usually fail to estimate non-frontal poses in team sports videos where players tend to frequently be displayed as side poses. The reason is that these methods need enough space between parts to become a star-shaped tree configuration of body parts. Even multiview part-based pictorial structure techniques [75] with pan-tilted cameras cannot robustly estimate the side and part-occluded poses in team sports videos because they still depend on the discriminative part classification as [8] does. If the side poses of sports players could be estimated with monocular videos, a broad range of possibilities of vision-based human motion and behavioral understanding would be opened up.

In this chapter, I propose a novel grid-wise joint location classifier *the Label-Grid classifier* for monocular team sports videos, which is integrated with a standard tracking-by-detection framework such as [125]. Label-Grid classifier estimates the lower body human joint location with Label-Grid resolution using the whole body appearance of the tracked window estimated by the player tracker (Figure 4.1). In other words, the player tracker first tracks the player

window in each frame, then the Label-Grid classifier estimates the joint location (grid position in the player widow (See Figure 3.2).

To the best of my knowledge, my method is the first human pose estimation method that can estimate the pose of side-running players with scale changes, which part-based methods [11, 49] cannot estimate very well. As the example results in Figure 4.1 show, my method can robustly estimate the poses of the side running sequence. Similar to (frontal) facial recognition techniques [126], the Label-Grid classifier embeds all types of *aligned* player window image into only one multi-class classifier, which enables pose estimation when they are running sideways.

Since my framework employs Histograms of Oriented Gradients (HOG) [24] as whole body gradient histogram features, it can estimate the joint location even from a low-resolution videos owing to the deformation invariance and contrast invariance of the HOG feature. In addition, it can estimate poses that have similar appearances between parts (e.g., pants with only one color) that part-based methods find it difficult to estimate, because part appearances become too ambiguous to detect (e.g., while crossing the legs).

The contributions of this chapter can be summarized as having the following advantages:

- Label-Grid classifier whose 2D unit blocks (Label-Grid) are synchronized with the resolution of Grid-Histogram features via multi-class classifier (in this chapter I use HOG features randomized by Random Decision Forests).

- Align the tracking window with a pelvis-aligned detector to provide center-aligned visual features (selected from HOG by Random Decision Forests) to easily classify the class of Label-Grid classifiers.

- Can also estimate the pose of side running sequences, which frequently appear in team sports videos.

- Per-frame estimation without temporal pose motion models of a specific action, such as [15], which uses a walking pose manifold or temporal pose prior.

- Per-frame pose estimation for videos *without* pictorial structure and part detectors. The ignorance of pictorial structure framework in my framework achieved fast pose estimation (about 1 fps computational time).

Figure 3.1: Example result of the lower body pose estimation framework. Images in the upper row show the tracked player images in each frame of the input video with the estimated joint position as colored circles in each frame. The lower image shows an input frame of the video. The rectangle is the tracked player window, and the green circle is the center of the window.

The rest of this chapter is organized as follows: the proposed framework is presented in Section 3.2. Section 3.3 describes how to learn the Label-Grid classifier with a prepared dataset. Section 5.3 illustrates the experimental results and evaluates the performance of my framework. Finally, Section 3.5 is the conclusion. Related work of this chapter was already discussed in Section 2.3.

3.2 Proposed Framework

The proposed framework (Figure 4.4) is composed of two modules: a tracking player with tracking-by-detection technique with the pelvis-aligned detector (Section 3.2.1), and estimating the four joint positions on the grid structure independently using Label-Grid classifiers (Section 3.2.2).

These two modules share the tracked player window as HOG features [24] to estimate the pose (locations of four joints) in each frame of the video (Section 3.2.2). At test time, the only

input of the framework is the player window position (rectangle) of the subject player in the first frame of the video. All the lower body poses in each frame are estimated automatically by tracking the player and are estimated by Label-Grid classifiers in each frame.

I train four Label-Grid classifiers of each lower-body joint separately; left-knee $L^{lk}(\boldsymbol{x})$, right-knee $L^{rk}(\boldsymbol{x})$, left-foot $L^{lf}(\boldsymbol{x})$, and right-foot $L^{rf}(\boldsymbol{x})$. Note that left or right means left in the image and right in the image respectively. The Label-Grid classifier learns the position of the left and right joints in the image plane just like the other pose estimators such as FMP [11][1].

3.2.1 Player Tracking with Pelvis-Aligned Detector.

The first module of the proposed framework is the player tracking method with standard tracking-by-detection, such as [125], to provide an aligned player window for the second module. I also used this pelvis-aligned detector in my upper body pose estimation framework [119], the explanation in [119] was very short because of the page limit. Hence, this thesis provides a more detailed procedure to prepare the dataset of my pelvis-aligned player detector in Section 3.2.3.

For tracking-by-detection, I use the player detector learned from the dataset $\mathcal{D}_{all}$ (Section 3.2.3). I use a Kalman Filter (instead of Particle Filter in [125]) to track the player whose likelihood in each frame is a non-maximum suppression result of the detections within the local area around the predicted player location. Since the proposed method mainly targets at estimating side poses during running or walking, which occurs very often in team sports videos (as mentioned in the introduction), I choose a Kalman Filter by assuming the simple and monotonous trajectory of the subject players in typical team sports videos on large fields.

This tracking procedure provides the smoothed and the center of the tracked window in each frame, and these tracked windows are expected to be aligned to the pelvis position. This first module provides the aligned window that works well for the Label-Grid Classifier in the second module. Since I use HOG feature for classifying the pose, I expect around 1 or 1.5 grid errors in this tracker to ensure that the Label-Grid classifiers can use the aligned HOG feature leaned from the pelvis-aligned dataset.

[1]This is the typical limitation of the two-dimensional human pose estimation methods. To overcome this, I will use the three dimensional information inferred by the other approaches in the future.

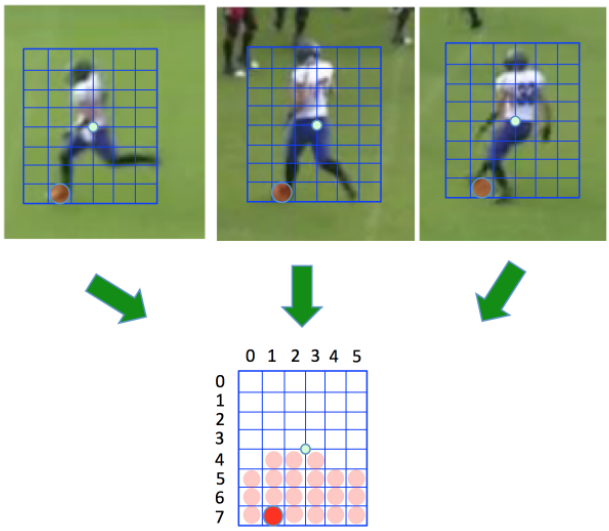


Figure 3.2: Label-Grid Classifier. The red circle on the grid is the classified joint location $\boldsymbol{l}_t^j \in \mathbb{N}^2$ on each player window from the learned Label-Grid class candidates (pink circles). In this example, the Label-Grid classifier for the j-th joint is on the (6×8) grid structure, and the estimated Label-Grid is on $\boldsymbol{l}_t^j = (1, 7)$ on all three images. The number of the class of the Label-Grid classifier of the left foot is 21 (sum of pink circles and red circles).

3.2.2 Label-Grid Classifier for Estimating Joint Grid Position.

The second module estimates the four joint locations using four Label-Grid classifiers. The proposed Label-Grid classifier is a multi-class classifier whose label classes are assigned to the grid locations of grid histogram feature such as HOG features [24]. The Label Grid classifier $L^j(\boldsymbol{x}_t) = \{F^j(\boldsymbol{x}_t), M^j(\hat{\boldsymbol{y}}_t^j)\}$ (for the j-th joint) consists of a multi-class classifier $F^j(\boldsymbol{x}_t)$ and the class-to-grid mapping function $M^j(\hat{\boldsymbol{y}}_t^j)$, where I denote the input visual feature vector (in my case, normalized three-level HOG) as $\boldsymbol{x}_t \in \mathbb{R}^D$ at frame t, and the estimated Label-Grid class label of $F^j(\boldsymbol{x}_t)$ is $\boldsymbol{y}_t^j \in \{l = 1, 2, \ldots, L\}$.

Each class l of $F^j(\boldsymbol{x}_t)$ learns the appearances (or poses) of players with its j-th joint is on the same grid (See Figure 3.2). At test time, the classifier $F^j(\boldsymbol{x}_t)$ of $L^j(\boldsymbol{x}_t)$ first estimates the class $\boldsymbol{y}_t$ from the input visual feature vector $\boldsymbol{x}_t$:

$$\hat{\boldsymbol{y}}_t^j = F^j(\boldsymbol{x}_t) \tag{3.1}$$

The reason why I use not only the lower body but also the full body window for the HOG feature is that I aim to leverage the whole upper body appearance for classifiers, which result

in capturing a wide variation in upper body appearances in each lower body joint position. I expect that this strategy of including upper body appearance makes the Label-Grid classifier easier to discriminate the pose of a specific joint position from the other poses even when the pelvis is not aligned, while the HOG of only the lower body could cause too much sensitivity to the mis-registration of the tracker[1].

After estimating the class label $\hat{\boldsymbol{y}}_t^j$ from the input feature vector, I map $\hat{\boldsymbol{y}}_t^j$ to the corresponding 2D grid location with $M^j(\hat{\boldsymbol{y}}_t^j)$:

$$\boldsymbol{l}_t^j = M^j(\hat{\boldsymbol{y}}_t^j) \tag{3.2}$$

where M^j is the dictionary function for the j-th joint to map each class $\hat{\boldsymbol{y}}_t^j$ to the corresponding grid location $\boldsymbol{l}_t^j \in \mathbb{N}^2$, which I call Label-Grid. This mapping dictionary M^j is built during training. I typically assign each class $\boldsymbol{y}_t^j$ to the grid from left to right and from top to bottom if there is more than one sample labeled on the grid (see Figure 3.2 for the example grid index assignment). Note that I need the inverse mapping of $M^j(\hat{\boldsymbol{y}}_t^j)$ during training, because I first have to assign each Label-Grid $\boldsymbol{l}_t^j$ in the $\mathcal{D}_{all}$ to the l-th class in L-class classifier. However, at test time, I need only $M^j(\hat{\boldsymbol{y}}_t^j)$ for converting $\hat{\boldsymbol{y}}_t^j$ to $\boldsymbol{l}_t^j$.

The training dataset for the Label-Grid classifier must include most of the types and scales of players' appearances that could occur in the target videos. Hence, my system can estimate the lower body pose in any location of the image (in this theis, the sports field) where player scale varies according to the position and the pose of the camera.

3.2.3 Dataset Preparation

I share the same data-augmented dataset $\mathcal{D}_{all}$ between two modules to learn the pelvis-aligned Detector and four Label-Grid classifiers. To share the player window with its center aligned to the pelvis of a player, the player detector and the Label-Grid classifier are learned from the same images and labels from $\mathcal{D}_{all}$. To create $\mathcal{D}_{all}$, I perform data augmentation with different scales and mirrored images from the original dataset $\mathcal{D}_{ori}$ (Figure 3.4).

I first prepare a training dataset $\mathcal{D}_{all} = \{\mathcal{D}_{sca}, \mathcal{D}_{mir}\}$ from realistic team sports videos to learn both player detector and Label-Grid classifiers. $\mathcal{D}_{sca}$ (Figure 3.4(a)) is the resampled player window images and their labels from the original dataset $\mathcal{D}_{ori}$ for which I should only need to prepare the labels. $\mathcal{D}_{mir}$ (Figure 3.4(b)) is the mirrored dataset of $\mathcal{D}_{sca}$ whose images

[1] If the estimation of the pelvis is perfect with any other methods, I need to use only the lower body appearance.

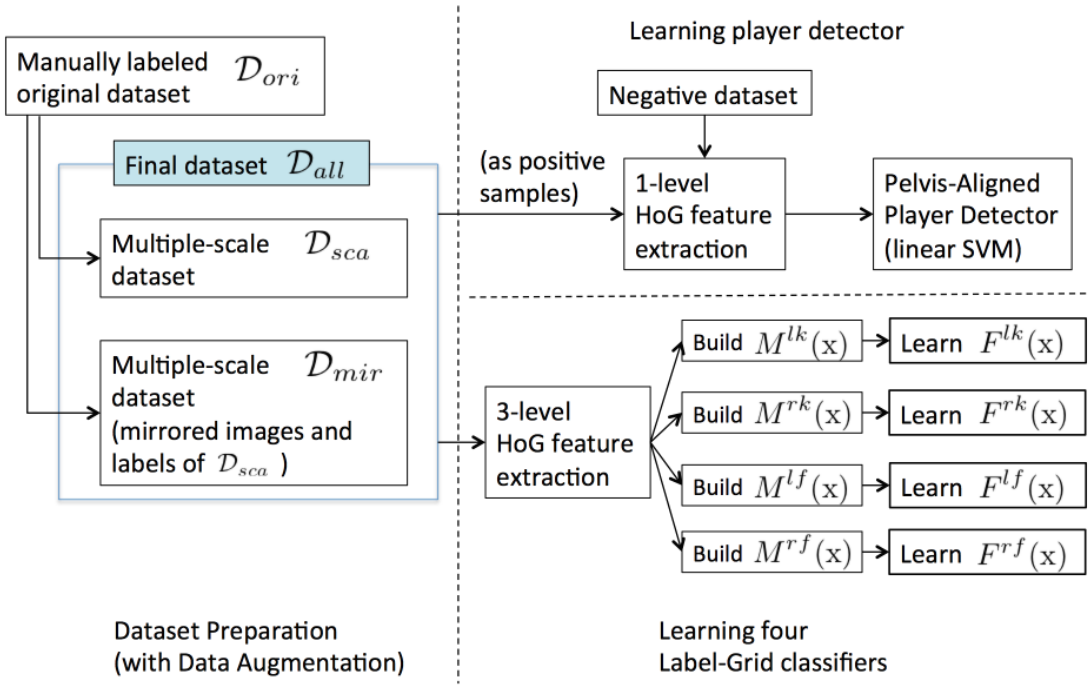


Figure 3.3: Learning procedure.

and labels are flipped horizontally. See the left half of the Figure 3.3 for this dataset preparation procedure.

First, I prepare a dataset $\mathcal{D}_{ori}$ with N images $\mathcal{I} = [I_1, I_2, \ldots, I_N]$ and labels of each image I_i so that the images includes various types of poses of players in one specific sport. Each player window image I_i in $\mathcal{D}_{ori}$ has labels $\boldsymbol{L}_I = [\boldsymbol{p}_i^1, \boldsymbol{p}_i^2, \boldsymbol{p}_i^3, \boldsymbol{p}_i^4, \boldsymbol{p}_i^{pel}, h_i^{pla}]$, where $\boldsymbol{p}_i^j \in \mathbb{R}^D$ is the j-th joint location of the i-th image I_i on the image plane, and h_i^{pla} is the player height for resampling the original images. $\boldsymbol{p}_i^{pel} \in \mathbb{R}^D$ is the location of the pelvis of the i-th image I_i on the image plane, which is always at the center of the player window and becomes the information of the location of the player window. Images $\mathcal{I}$ are all clipped from the team sports videos so that their window centers $\boldsymbol{p}_i^{pel}$ are all aligned to the center of the window, and the labeling person manually inputs h_i^{pla} as a length between the top of head and the bottom of the foot of the player in I_i. This labeling procedure determines the scale of the player height h_i^{pla} to the fixed size window in each sample.

For resampling the original image of $\mathcal{D}_{ori}$ to multiple scales, I resize all the images and labels in $\mathcal{D}_{ori}$ to the resampled player scales $s = h_i^{pla}/h^{win}$, where h^{win} is the height of the Label-Grid widnow. $\mathcal{D}_{sca}$ includes several player scales with regular intervals (e.g., 0.80, 0.85,

(a) Example images from scaled dataset $\mathcal{D}_{sca}$ with different player scales $s = \{0.7, 0.8, 0.9\}$. Note that all player windows for Label-Grid (purple grid) have same fixed window size.

(b) Example images from mirrored dataset $\mathcal{D}_{mir}$ created from images and labels $\mathcal{D}_{sca}$.

Figure 3.4: Data Augmentation. Using h_i^{pla} (height of the blue window), images are scaled to the scale s so that the center position $\boldsymbol{p}_i^{pel}$ keeps to the center of the Label-Grid even in the scaled images. By performing this aligned image sampling of the training dataset, the feature space of the Label-Grid classifier can be augmented to the multi-scale player sizes within the Label-Grid window.

..., 1.00) by resizing $\mathcal{D}_{ori}$ (Figure 3.4(a)).

Finally, I acquire $\mathcal{D}_{mir}$ by flipping the images and labels of $\mathcal{D}_{sca}$ horizontally to learn the mirrored features and labels (Figure 3.4(b)).

The player detector uses only the images of $\mathcal{D}_{all}$ because the centers of the players' window images in $\mathcal{D}_{all}$ are all aligned to the pelvis of the players. In contrast, images and Label-Grid classes of $\mathcal{D}_{all}$ are used to learn the classifier. Any types of multi-class classifier can be used as a Label-Grid classifier. However, for its high precision and fast computational time, I use Random Decision Forests [93] as a Label-Grid classifier in the experiments.

3.3 Learning Label-Grid Classifier

The Label-Grid classifier is a multi-class classifier with its class labels (Label-Grid) assigned to the 2D grid location of a feature type with a grid structure such as HOG features. The Label-Grid classifier can be any types of multi-class classifier (e.g., Support Vector Machine, Random Decision Forests, etc.), but preparing the dataset for a Label-Grid classifier to classify the lower body grid-position of the player is the original approach to using a general multi-class classifier.

Every one class (Label-Grid) of the Label-Grid classifier learns the HOG features whose joint is on the same grid position (Figure 3.2). Given the grid feature of a player in a $W \times H$ window, classifying the Label-Grid of a specific joint (e.g., left-knee) can be regarded as an

L-class grid classification problem, where the task is to choose a grid position (i, j) from L candidate positions (pink circles are marked as candidate positions in Figure 3.2). The other N grids in the grid feature are just ignored from label-grids to learn. The number of Label-Grids L is decided when building $M^j(\boldsymbol{y}_t^j)$ (see Section 3.3.1). For instance, if you use HOG features with 6×10 cells in a player window and if there are 35 classes where the joint label-grid exists more than one joint position in the training dataset, the Label-Grid classifier becomes 35-class classifiers. The other 25 ($= 6 \times 10 - 35$) grids, which have no joint labels in the training dataset, are ignored for the classification of the joint.

3.3.1 Learning Procedure

I will explain how to learn a Label-Grid Classifier for classifying the j-th joint (e.g., the left knee joint). Figure 3.3 shows the whole procedure, used to learn the Label-Grid Classifiers ($L^{lk}(\boldsymbol{x})$, $L^{rk}(\boldsymbol{x})$, $L^{lf}(\boldsymbol{x})$, and $L^{rf}(\boldsymbol{x})$) and the pelvis-aligned detector by preparing a dataset $\mathcal{D}_{all}$.

Given a data-augmented dataset $\mathcal{D}_{all}$, I first calculate the grid location $\boldsymbol{l}_i^j$ from the j-th joint $\boldsymbol{p}_i^j$ of the i-th image in $\mathcal{D}_{all}$ using pelvis position and the size of Label-Grid (e.g. each grid is 8×8 and the window size is 64×128).

After calculating all $\boldsymbol{l}_i^j$ in $\mathcal{D}_{all}$, I then build the mapping function $M^j(\boldsymbol{y}_i^j)$ to decide the number of the class N of the Label-Grid Classifier and all the Label-Grid indices $\boldsymbol{y}_i^j$ of the i-th image in $\mathcal{D}_{all}$. After $M^j(\boldsymbol{y}_i^j)$ has been built, I can finally learn $F^j(\boldsymbol{x}_i)$ (using Random Decision Forests, in this theis) with Label-Grid s and the calculated feature $\boldsymbol{x}_{all}$ that I will explain in the next subsection.

3.3.2 Multi-level HOG Feature and Feature Selection

I use a three-level image pyramid from the player window for calculating three-level HOG features $\boldsymbol{x}_t^1, \boldsymbol{x}_t^2, \boldsymbol{x}_t^3$ for making the feature vector $\boldsymbol{x}_t = [\boldsymbol{x}_t^1 \boldsymbol{x}_t^2 \boldsymbol{x}_t^3]$ for a Label-Grid classifier. Learning multiple resolution of the HOG appearance makes the Label-Grid classifier restrict the label-grid candidates at each resolution level, which helps to avoid the bad classification result far from the true position.

To decrease the effect of the difference of feature scales between the three levels, I normalize the feature vector $\boldsymbol{x}_t$ to L_2 unit vector both at training time and test time.

I use L-class Random Decision Forests as the L-class Classification Forests [25] as $F^j(\boldsymbol{x}_t)$ of the Label-Grid classifier, which results in performing feature selection from these normalized

three-level HOG features $\boldsymbol{x}_t$. After learning the L-class, each split function uses two randomly selected values of the multi-level feature vector $\boldsymbol{x}_t$ to estimate the class label $\boldsymbol{y}_t^j$.

3.4 Experiments

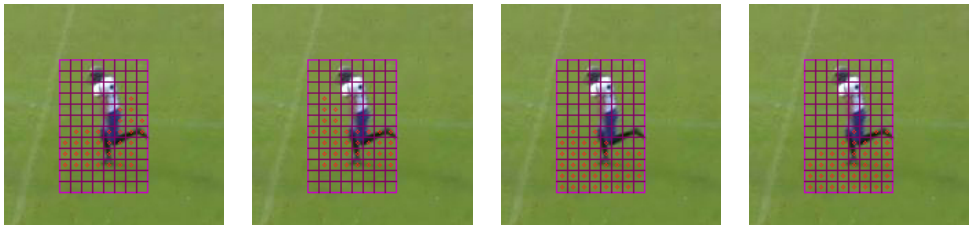
I tested my framework in two scenarios: frontal pose sequences (Section 3.4.3.1), which part-based pose estimator [11] can also estimate robustly, and side pose sequences (Section 3.4.3.2), which part-based pose estimator *cannnot* predict properly as argued in Section 2.3.2.

3.4.1 Experimental Setup

I performed experimental evaluations on my system with American football videos in professional league matches. The size of each video is 1280×780. The videos are taken from the matches of Panasonic IMPULSE[1]. All videos were captured from the high place in the stadium with fixed cameras. All videos were converted to 29 fps videos while the original videos were recorded at 59 fps. These videos include players from a team with a white-colored uniform and players from the other team with a black-colored uniform. Although I captured high-resolution videos, motion blur of moving legs and arms sometimes occurs and the players are captured with a relatively low resolution. I created 10 test sequences (test(1)–(10)) for the five frontal pose tests and five side pose tests from these videos (see Figure 3.6 to see the player trajectories on each sequence). Each test video is composed of 40 frames. I will show the detail behavior (pose) on each sequence in Section 3.4.3.1 for the frontal pose sequences and Section 3.4.3.2 for side pose sequences.

I manually clipped player windows from video frames and assigned labels to create the original dataset $\mathcal{D}_{ori}$ for training both Label-Grid classifiers and the player detector. I tried to include as many pose patterns (and also views of the pose) as possible in the dataset $\mathcal{D}_{ori}$ to make the Label-Grid classifiers learn the whole possible appearance patterns in American football videos. I randomly selected the images from all the videos so that the original dataset includes more versatile player poses, and the original dataset becomes 977 images and its labels. Note that approximately 10 % of the original training images shares the same images with test dataset sequences of test(7) and test(8). Then I resampled 977 images and labels of $\mathcal{D}_{ori}$ with 13 scales $s = \{0.70, 0.725, 0.75, \ldots, 0.975, 1.0\}$ and finally prepared 25402 images ($25402 = 977 \times 13 \times 2$) for training four Label-Grid classifiers independently.

[1] http://panasonic.co.jp/es/go-go-impulse/



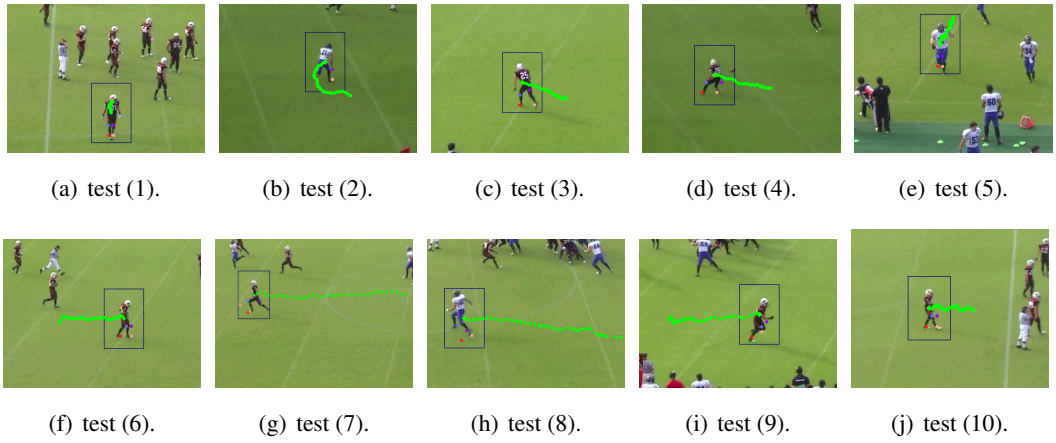
(a) left knee Label-Grid classifier. (b) right knee Label-Grid classifier. (c) left foot Label-Grid classifier. (d) right foot Label-Grid classifier.

Figure 3.5: Four Label-Grid classifiers with 8×12 Label-Grids, which I use in the experiments. Each red circle shows the candidate Label-Grid class of the classifier.

HOG window size was 64×128 pixels (width $\times$ height), and the cell size was 8×8 both for my pelvis-aligned player detector and the four Label-Grid classifiers. For learning the pelvis-aligned player detector, I only used 64×128 HOG and labels of $\mathcal{D}_{all}$. For the Label-Grid classifiers, I also created pyramid images 48×64 and 24×32 from the tracked 64×128 window image in one frame. I then calculated three-level HOG $\boldsymbol{x}_t^1, \boldsymbol{x}_t^2, \boldsymbol{x}_t^3$ from each level pyramid image with 8×8 cell size and combined them. Finally, I obtained a 2268-dimensional L_2-normalized feature vector $\boldsymbol{x}_t$ as the input of each Label-Grid classifier.

I learned four Label-Grid classifiers as Random Decision Forests [25] with the feature vector $\boldsymbol{x}_t$ for each lower body joint independently (right/left knee and right/left foot) from the training dataset $\mathcal{D}_{all}$. Consequently, I had 34-class left knee Label-Grid classifier, 34-class right knee Label-Grid classifier, 38-class left foot Label-Grid classifier, and 39-class right foot Label-Grid classifier (see Figure 3.5 for the class assignment).

To apply FMP [11] as a baseline, I used PartsBasedDetector software [127]. I used 26-parts frontal person models as FMP and regard center positions of the 4 part-detector as four lower body joints to compare with my joint location classifiers (index 12 as left knee joint, index 13 as left foot joint, index 24 as right knee joint, index 25 as right foot joint). I assume that the center of the rectangle of each detected part is the corresponding joint position in the image.



(a) test (1). (b) test (2). (c) test (3). (d) test (4). (e) test (5).

(f) test (6). (g) test (7). (h) test (8). (i) test (9). (j) test (10).

Figure 3.6: Tracked results of all tests (1)–(10). Test (1)–(5) are the results of the frontal pose sequences while tests (6)–(10) are the results on the side pose sequences. Green dots show the player window center locations in each frame.

3.4.2 Evaluation Manner

Pixel Error of the Joint Position.

For measuring the performance of my lower body pose system, I define the Euclidean distance error as below:

$$E_t = d(\hat{\mathbf{p}}_t, \mathbf{p}_t^{\mathbf{GT}}) \tag{3.3}$$

where $\hat{\mathbf{p}}_t = (x, y)$ is the center point of the estimated Label-Grid location and the $\mathbf{p}_t^{\mathbf{GT}}$ is the ground truth position. Since my Label-Grid classifiers are learned with 8×8 Label-Grid, the center position $\hat{\mathbf{p}}_t$ becomes $(4, 4)$ from the left-top point $(0, 0)$ in each Label-Grid.

Note that the running speed of the player is fast in most of the experimental videos because I apply my method to the isolated running players, such as Quarterback, Runningback, and Linebacker. For this reason, the length of each video is very short (40 frames). Another reason is that I cannot collect many sequences of long running isolated play easily, because each American football play is around only 10 seconds and players tend to be occluded and congested frequently.

Although I would like to compare my methods with FMP using the PCP [14] score, which is broadly applied to the evaluation of part-based methods, I cannot calculate the PCP score because my method does not infer the stick area of each part which is needed for calculating PCP scores. This is one of the reasons why I used the Euclidean distance for the evaluation.

Detection Rate of FMP and How to Apply FMP to the Videos.

To test the limitations of FMP for side poses in the videos, I defined the detection rate R as $R = k/N$, where k is the number of detections against the number of frames N in one test video (in my case, $N = 40$ frames).

Since FMP[11] was the object detector (but jointly estimate the pose while detecting the object), I automatically clipped the magnified and margin-added image to apply the FMP detector. I first clipped the player window from the tracking-by-detection tracking module (Section 3.2.2) by adding 40×40 margin, and magnified it 200% to enable FMP to detect the players in the video. Each of images in Figure 3.8 shows the clipped images with this procedure.

Table 3.1: Average estimation error of each joint in the frontal pose tests (1)–(5). All errors are in pixels.

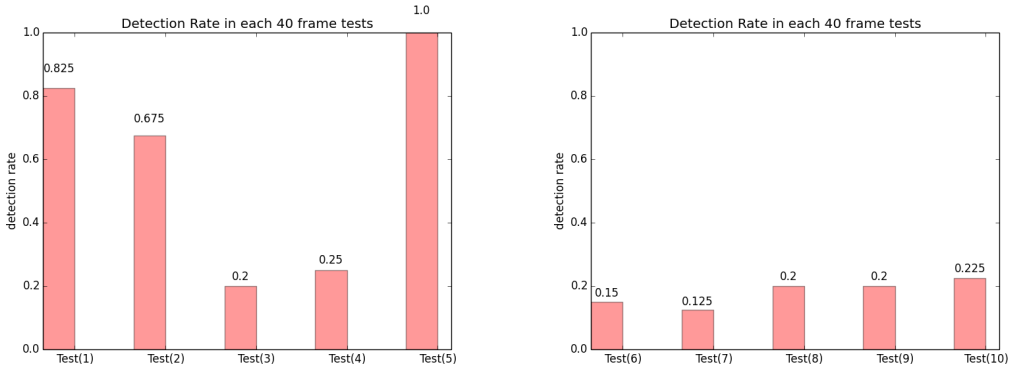
Joints	(1)	(2)	(3)	(4)	(5)
Left knee	15.07	14.92	15.58	14.13	13.66
Right knee	9.29	13.01	22.28	15.71	20.26
Left foot	9.89	9.98	13.60	16.28	11.50
Right foot	10.99	23.28	20.31	16.62	10.83
Pelvis	6.06	4.42	4.91	4.93	6.80

Table 3.2: Average estimation error of each joint in the side pose tests (6)–(10). All errors are in pixels.

Joints	(6)	(7)	(8)	(9)	(10)
Left knee	9.65	14.40	8.08	16.22	5.93
Right knee	9.62	15.99	11.45	8.34	16.93
Left foot	6.81	16.19	24.83	11.56	6.80
Right foot	20.58	19.31	28.67	19.10	21.34
Pelvis	4.26	6.33	13.01	3.50	5.08

3.4.3 Experimental Results

Figure 3.7(a) shows the detection rate of FMP [11] in tests (1)–(5). Since the movement of the players in tests (2)–(4) are diagonal and curved (Figure 3.6), most of their poses were difficult for FMP with hard occlusion and low-resolution, even though I defined those tests as frontal tests. However, since my method does not use any part-models and just use tracked (whole)



(a) Detection rate of FMP[11] in frontal pose tests (1)–(5). (b) Detection rate of FMP[11] in side pose tests (6)–(10).

Figure 3.7: Detection Rate of FMP[11] in each test.

player window appearance in each frame, it can classify the joint position in all frames in test (1)–(5). For example, in the Figure 3.8(d), while FMP could not detect the player in each frame, the Label-Grid classifier estimated the joint positions correctly from the same images. While I wanted to compare the position error between my method and FMP, I abandoned the calculation of the pixel error for FMP since I was not able to get enough detections from even frontal poses (Figure 3.7(a)).

3.4.3.1 Frontal Pose Experiment

I tested my system on the following frontal pose scenarios to compare the performance or detection rate with FMP [11].

I prepared the following five sequences for a frontal pose dataset (Fig 3.6, left column):

- Test (1): The player walks to the start position while facing their frontal upper body to the camera.
- Test (2): The player runs up to the upper side of the field.
- Test (3): The player begins to run from the start position.
- Test (4): The player runs diagonally.
- Test (5): A large player walks to the outside of the field.

(a) test (1). (b) test (2). (c) test (3). (d) test (4). (e) test (5).

Figure 3.8: Example results from frontal pose experiments. The left tow panels in each subfigure (a)–(e) show the results of the Label-Grid classifiers, and the right panels show the result of the FMP, where only the four detected joints are shown (no visualization of joints means that FMP could not detect anyone in the frame.).

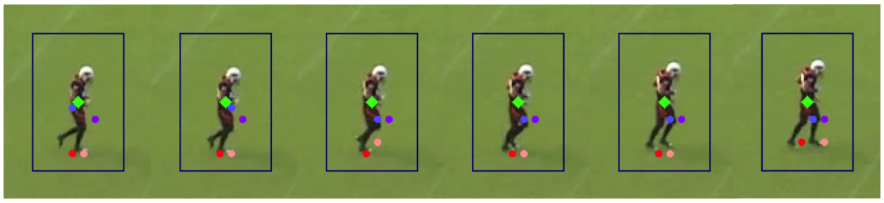
Figure 3.6 shows the tracked trajectory of pelvis position (center of the player window) in each of tests (1)–(10). The panels in the left column show the results of frontal pose tests (1)–(5). Table 5.3.1 shows the average error of my method and FMP [11] for each joint. Note that the Label-Grid is 8×8 pixels for all tests. While FMP sometimes failed to detect a player who had occluded parts, my method could detect non tree-structured poses. Figure 3.8 shows the example results of my method and FMP to compare with each other.

3.4.3.2 Side Pose Experiment

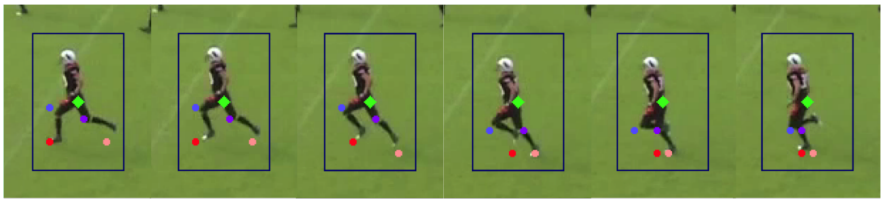
Just as for the frontal pose experiment in the previous section, I also performed evaluation of my method and FMP with the following five side pose scenarios (Fig 3.6, right column):

- Test (6): The player runs straight (namely, almost no scale change) at relatively slow speed from left to right.
- Test (7): The player runs very fast from right to left.
- Test (8): The Runningback player runs diagonally from the starting position.
- Test (9): The Runningback player runs straight from the starting position.
- Test (10): The player walks backward.

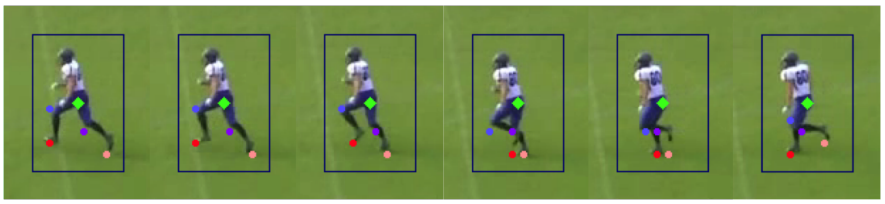
I collected these side pose test videos so that the upper body direction of the player was almost the same as the lower body direction.



(a) test (6).



(b) test (7).



(c) test (8).

Figure 3.9: Example results from side pose experiments. Each row shows the results of side pose test (6), (7), and (8).

Table 5.3.1 shows the average error of my method in these side pose tests and Figure 3.9 shows the example results of tests (6)–(8). Figure 3.7(b) shows the detection rate of FMP [11] in side pose tests (6)–(8). As Figure 3.9 shows, FMP can rarely detect the player in side pose tests. FMP detects player with a detection rate 0.15 to 0.25. Compared with these results of FMP, my method could estimate the pose in all frames via its tracking and classification procedure within about two Label-Grid errors (Table 5.3.1).

3.4.4 Discussions by Topic.

Whole Body Appearance Feature as Multi-Level HOG.

As already argued in Section 3.2.2, I use the whole body appearance to classify the lower body pose. The proposed HOG-based classification approach can be viewed as the modern replacement of the classical silhouette-matching schemes using background subtraction, such

as [90]. I instead use randomized HOG features (learned by Random Decision Forests) to robustly classify the pose with machine learning. The proposed strategy has richer information with which to discriminate Label-Grid classes than using only the lower body appearance. I instead use the whole body HOG appearance to estimate the joint position.

Moreover, owning to the deformation invariance of the HOG features, my Label-Grid classifier can estimate the pose of a larger or slimmer person until the gradient distribution changes from the feature distribution of the dataset. For instance, I performed an experiment using a large player in test (5) (see Figure 3.8(e)). Even though I only included middle-sized and thin players in the original dataset $\mathcal{D}_{ori}$, my classifiers could still estimate the joint location of the large-sized player.

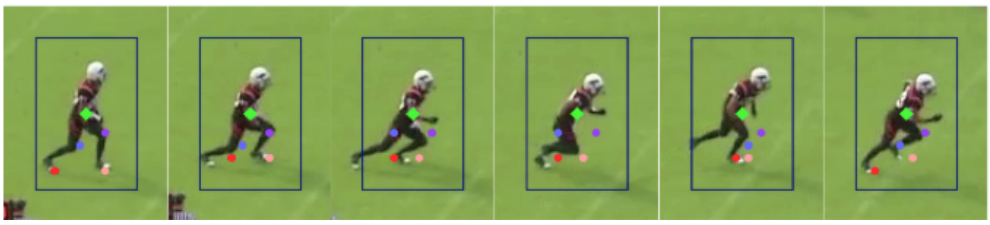
Disregard of Pictorial Structure and Low-Resolution Invariance of The Proposed Method.

Another important part of the nature of my method is that my framework disregards the pictorial structure strategy [104] while it depends on the aligned window appearance. As I showed in the experimental results for side pose tests (Figure 3.9), my method can model any types of pose including hardly occluded side poses, which pictorial picture models cannot infer very well owing to their tree-models. As I already argued in Section 3.3.2, my three-level HOG feature and the Randomized feature selection helps to restrict the error as much as possible. Since the Random Decision Forests technique takes advantage of the spatial grid structure to learn the distance between classes, the error seems to be restricted within neighboring Label-Grid (See Figure 3.8 and Figure 3.9).

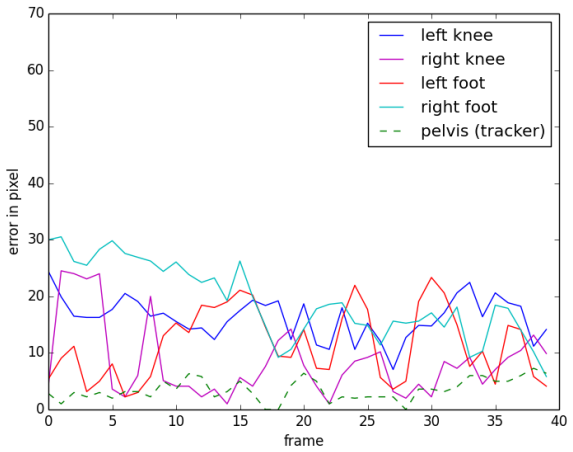
While FMP and the other part-detector techniques assume clear and non-blurred images, multi-level HOG-based Label-Grid classifiers can even classify the poses in low-resolution and motion-blurred images because the HOG feature is robust for contrast change (between image scales) using grid-wise edge histogram pooling and block-wise normalization [24].

Sport-Specific Classifier.

While my method is able to estimate the lower body pose with various types of poses in American football robustly, the Label-Grid classifier is learned from the same clothing type while the pictorial structure includes various types of clothing. In the experiments, welearned Label-Grid classifiers from two American football teams, but the classifiers can classify the lower body pose with the appearances of both teams.



(a) Results from side pose test (9). From left to right, each image is frame 8, 12, 16, 20, 24, and 28 respectively.



(b) Joint position error in each frame of side pose test (9).

Figure 3.10: Temporal analysis of test (9).

Importance of Alignment and Scale of the Window.

As already mentioned, my framework depends on the alignment of the player window between the tracking module and the classification module. In the experiments, the player detector is learned with a HOG of 8×8 cell size, which tracks the player within a cell error. This means that Label-Grid classifiers can only deal with the window patterns within one or two (at most) cell size drift. Hence, Label-Grid classifiers can robustly estimate the grid location if the tracker can provide well-aligned windows.

Figure 3.10 shows the temporal analysis of the side pose test (9). By observing Figure 3.10(a), the left foot position gradually becomes unstable as the left leg is out of the window. This sequence shows the nature of the window alignment scheme. While you can use wider windows for the Label-Grid classifiers to prevent this case by restricting the person within

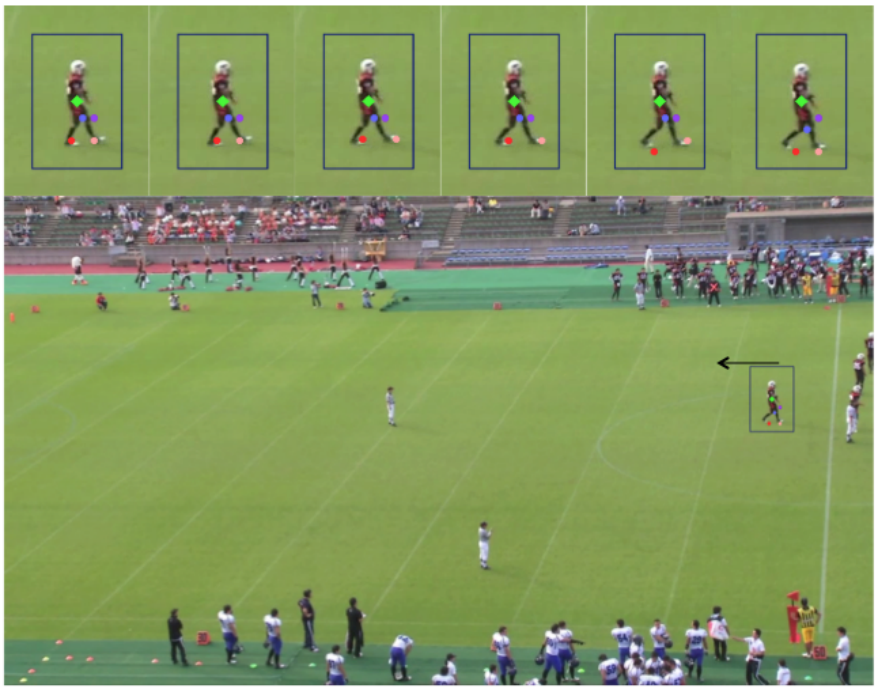


Figure 3.11: Walking back result of test (10).

the window, you need to make more patterns for the dataset because the number of classes increases with a wider Label-Grid widow.

Figure 3.10(b) shows the error values of four lower body joints and the pelvis position in each frame of the test (9). Owing to the dependence of the alignment of the tracker, Label-Grid classifiers tend to have more errors along with an increase in pelvis position error. In Figure 3.10(a), each joint errors tends to increase when pelvis error becomes large.

In addition, whether the player is within the scales of the dataset during the test time is also important. In the experiments, I used scales $s = \{0.70, 0.725, \ldots, 1.0\}$ for creating the augmented training images. If the player scale within the window is too small or too large, the three-level pyramid HOG features will become an unknown pattern for the Label-Grid classifier (Random Decision Forests).

Our framework has two advantages for overcoming this problem. First, Random Decision Forests can learn the inter-class distance, as the Label-Grid classifier tends to misclassify the sample with the neighboring Label-Grid. In addition, the three-level pyramid HOG features also help the appearance over three resolution levels and help to evade misclassification to the distant Label-Grid class. Even though these two advantages help to embed as many (contin-

uous) scales of features (in Random Decision Forests feature space) as possible, the failure happens if the player is an unknown scale (e.g., $s = 0.60$).

Per Frame Estimation for Moving Back Players.

In team sports videos, players during defensive action tend to have a pose or body direction that is not the same as the player's moving direction. Figure 3.11 shows the result of test (10).

This shows the ability of the proposed method to classify the pose correctly even when the player is moving backward. This feature shows the high applicability to team sports videos, whereas walking and running backwards only rarely appears in surveillance videos.

3.5 Conclusion

To estimate lower body poses from low-resolution images, this chapter proposed a new human pose estimation method using a Label-Grid classifier that is integrated with an object tracker. The Label-Grid classifier does *not* use the pictorial structure, *but* use the alignment of the player's pelvis position and use this aligned whole body HOG appearance features to classify various types and scales of poses into a grid structure with off-the-shelf multi-class classifiers (I use Random Decision Forests in this chapter). Alignment between the tracking-by-detection module and the Label-Grid classification module is the key to realize the estimation of lower body poses with all poses in team sports videos.

Our system can even estimate poses of the isolated player with part-occlusions and non-upright poses, which are difficult to estimate with the methods using pictorial structures and part detectors. The proposed pose classification strategy using a whole person HOG makes it possible to classify the lower body joint locations of a player even during the side running poses. The proposed framework can be viewed as a revisited version of [90] by using machine-learning and dense visual features, while [90] only used noisy silhouettes of the person for pose estimation. In other words, traditional silhouette matching strategy for pose estimation was innovated by my approach using HOG features and Random Decision Forests to embed all pose appearance patterns into the randomized feature space.

In this chapter, I only investigated the possibility of my framework for an isolated player without any occlusion between players. However, the lower body pose estimation of isolated players will be useful for many team sports videos because players are mostly isolated during play.

3. LOWER BODY POSE ESTIMATION WITH LABEL-GRID CLASSIFIER AND THE CENTER-ALIGNED PLAYER TRACKER

As experiments showed, the proposed system can estimate all types of poses with only RGB videos if a sufficient amount of poses are prepared in datasets. In addition, my method can estimate side-running poses while FMP [11] can only detect star-shaped part configurations and cannot estimate non-star side running poses. However, the estimation fails when the alignment is not very accurate because of the drift of the tracker.

I believe that this method's advantage over previous pose estimation methods will open up the wide range of potential of player activity recognition from the estimated joint positions using only monocular cameras and any low-resolution settings of people tracking. Joint positions of the lower body will provide a new source of richer information for sports data analysis with only passive sensing.

4

Upper Body Pose Estimation with Poselets-Regressor for Spine Pose and the Body Orientation Classifiers Conditioned by the Spine Angle Prior

This chapter will present the upper body orientation estimators conditioned by the spine angle range that can also deal with the non-standing poses of the team sports players for the first time. This work has previously published as [128], which is the extension of [119].

The key contribution of this chapter is the *poselets-regressor* which is the estimator of a relative joint position (pelvis center) from the origin joint (head center) with global aligned upper body appearance as poselets detector [30] uses mid-level human parts. While the previous work tend to use local part appearance for estimating a part or joint location [8], the poselets-regressor employs aligned person appearance as the label-grid classifier in Chapter 3.

4.1 Introduction

In team sport video analysis, tracking players with computer vision-based methods is widely studied because the trajectories of players are the most basic and important kinds of information. There have been many applications that track players using a monocular view [129] or multiple views [130]. However, these player-tracking methods can only achieve location-based activity recognition of players, for example [131, 132]. Conversely, there are a few studies on

4. UPPER BODY POSE ESTIMATION WITH POSELETS-REGRESSOR FOR SPINE POSE AND THE BODY ORIENTATION CLASSIFIERS CONDITIONED BY THE SPINE ANGLE PRIOR

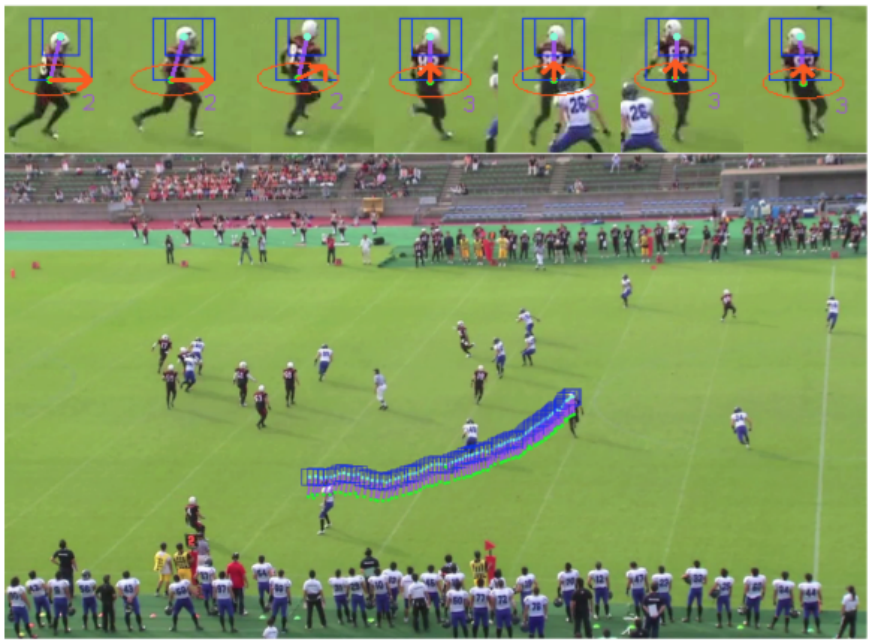


Figure 4.1: Example result from our framework. Images in the upper row show the tracked player images in each frame of the input video with the estimated horizontal body orientation (orange arrow), and 2D spine pose (2D purple line). The lower image is a summary image from a test video, which shows the location of the head region and the pelvis center of the subject player in each frame. The proposed method can track and estimate the upper body pose even when lower body occlusion occurs because it only utilizes the upper body region appearance of the tracked player.

group activity recognition for sports videos using per-player actions as features [6, 133]. While these methods can infer the semantic actions of each player (e.g., "running", "jumping"), they cannot recognize the fine-grained activity differences between activity classes because they just perform (discretized) classification between those semantic actions.

If sport-specific *upper body pose* patterns can be estimated from a vision-based recognition technique, there will be a new opportunity to realize the more detailed *pose-based* activity recognition of team sports players. The extraction of the upper body pose will achieve a deeper understanding of player actions by recognizing changes in the upper body pose, such as the gradual spine angle change during running from the starting position, defensive bending poses, and blocking poses. Moreover, gaze and direction-based attention prediction [63] and group activity recognition from orientations and relative locations of people [6, 133, 134] can also be achieved even for the team sports videos in the future. However, estimating the upper body

(a) Spine angle variation in bending poses.

(b) Running while looking back (the moving direction and the body orientation are different.)

(c) While the player is moving back.

Figure 4.2: Variation of human poses in team sports videos. These poses rarely appear in pedestrian surveillance videos and produce visual patterns for classifying body orientation.

orientation or body tilt from team sports videos has rarely been explored.

Realistic upper body pose appearances of team sports players have a wider variety of articulated pose patterns (Figure 4.2) than pedestrian cases [18, 26, 117] with the following variations:

- Many types of spine angle (body tilt) (Figure 4.2 (a)).
- Running while looking backward (Figure 4.2 (b)).
- While moving back (Figure 4.2 (c)).

These postural patterns, unseen in surveillance videos for pedestrian tracking, make it more difficult to realize the upper body pose estimation method for team sports videos. Specifically, sports players in team sports videos have more *body tilt variations* than pedestrians poses because they tend to bend their (upper) body while in defensive actions or some specific actions (e.g., passing action in soccer). Larger body tilt variations make the body orientation problem more difficult because the previous body orientation estimators during pedestrian tracking or detection [26, 135] depend on the alignment of the input window by the pedestrian detector for only standing walking poses.

To cope with those postural variations in team sports, I propose a framework for estimating the upper body orientation of a player with the *head-center-aligned* upper body region using the selected classifier conditioned by the spine pose angle (Figure 4.1. The proposed framework, which depends on the alignment of the upper body appearance, not only estimates the body orientation of the tracking player (orange arrow in Figure 4.3 (a)) as in previous work on

surveillance [26, 135], but also estimates the *2D spine pose* of the players (purple line in Figure 4.3 (a)), which consists of the head center location and the pelvis center location *even during the bending poses*.

Our framework is composed of three steps. In the first step, the head position of the moving player is tracked by the head tracker of [3]. In the second step, I estimate the relative pelvis center location from the head center position using our proposed *poselet-regressor* with *head-center-aligned* Histogram-of-Oriented Gradients (HOG) [24] features within the upper body window. This results in estimating the 2D spine pose in each frame. Our poselet-regressor has continuous output space, while the original poselets [30] are trained as a pose exemplar detector. In the third step, I use the estimated 2D spine angle from the second step as a conditional prior for selecting a corresponding upper body orientation classifier, and then the upper body orientation is estimated by the head-center aligned (or pelvis-center aligned) upper body region HOG features within a corresponding spine angle range.

This framework is the extension of our previous body orientation and spine pose estimation work [119] with two major contributions: (1) relative spine pose estimation with the head tracker and the poselet-regressor with head-centered-aligned upper body appearance; and (2) classifier selection scheme using spine-angle prior and aligned appearance window, which was inspired by [105, 110], but with the difference that our conditional prior is the 2D spine angle class. Note that this chapter only focuses on the body orientation estimation problem while the previous version [119] also proposed the head orientation estimator.

The first contribution of this chapter is the proposed poselet-regressor (in step 2) that estimates the 2D spine pose *without* using pictorial-structures-based pose detectors such as [11, 12, 61] and the poselet detectors [30, 31], which has been a popular strategy for human pose estimation in computer vision. Our poselet-regressor predicts the relative pelvis position from the head center using the head-center-aligned upper body appearance determined from regression forests [93]. Compared with the multiview pose estimation method using poselets [103], our framework tries to estimate the 2D spine pose (the line between the head center and the pelvis center) using the poselet-regressor and the head-center-aligned window from the head tracker. In other words, our poselet-regressor is a relative joint location predictor that depends on the local origin (head center) estimated by the head tracker in each frame of the video.

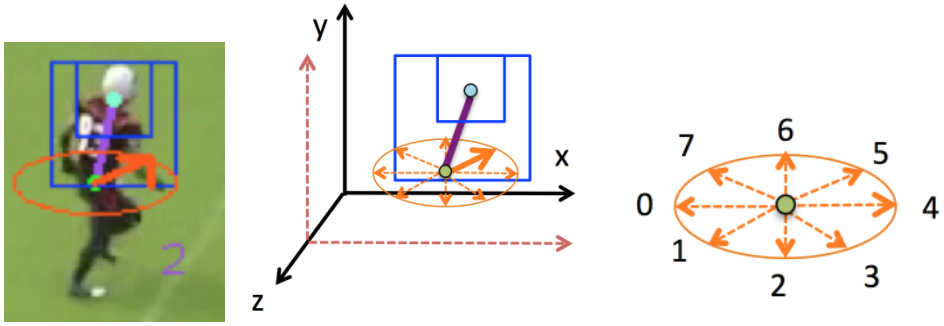
The second contribution of this chapter is the switching scheme between multiple upper body orientation classifiers (in step 3) using the spine angle value as a conditional prior. This

enables each body orientation classifier (random decision forests) to focus on selecting the important (HOG) features from the conditioned subset of the whole training dataset that includes similar and spine-pose-aligned upper body appearances with the same spine angle range. Previous body orientation estimation approaches for surveillance videos [18, 26] do not deal with bending poses but only pedestrians walking or standing upwright. Our conditioned prior scheme is inspired by the conditional regression forests [105, 110], but our conditional prior is the spine angle, which has never been explored before as a prior.

In addition to these major two contributions, another key contribution of this framework is that our method estimates the upper body pose of the player even when partially occluded, because it depends on the head tracker and the upper body orientation classification using *only* the selected features within the upper body region during training time with Regression Forests [93]. Since my previous work [119] depends on tracking the whole body, it can only track and estimate the pose of isolated players. Our new design, which tracks the head and uses the head-center-aligned upper body appearance, which is inspired by our another lower body pose estimation framework in the previous chapter [102], opens up more chances to track and estimate the pose of players even in congested situations in team sports by only tracking the head region and using only the tracked upper body region for spine pose and body orientation estimation.

Since the variation of the arm pose is very large in unconstrained team sport player appearances, it is also important to depend aligned global appearance of the person. Because parts exemplars such as poselets [30, 31] and pictorial structures [11, 12] have to deal with the all of the arm parts for pose prediction, they are not always scalable to the unconstrained huge patterns of articulated poses. Instead of training parts detectors, I introduce the alignment of the whole body appearances from the tracker, which is the standard approach of body orientation classification methods during tracking-by-detection [26, 119, 135], and utilizes only discriminative HOG features within the upper body region selected by random decision forests training. Moreover, our poselet-regressor also uses the same aligned-global appearance approach to predict the relative pelvis center location from the (tracked) head center.

In summary, our alignment via head tracking and the feature selection with conditional spine pose prior are aimed at dealing with pose appearance patterns of sports players rather than pedestrians (for estimating their body orientation). By acquiring head and pelvis-center aligned upper body images, each body orientation classifier needs to deal with only the smaller



(a) Example output. (b) Example output in 3D space. (c) Body orientation classes.

Figure 4.3: System output in image and 3D space. (a) Example results shown in image. The smaller blue rectangle is the head region tracked by the head tracker. The larger blue rectangle is the upper body region for the poselet-regressor and the body orientation classifiers. (b) Visualized result in 3D space. The orange arrows show the eight horizontal orientation classes. The purple line is the spine, which includes the head center position (cyan) and the pelvis center position (green). The number showing at the right-bottom corner of the upper body means the selected spine-class s. (c) Class numbers for each body orientation class.

appearance distribution within the corresponding spine angle class. To achieve this, even for an unconstrained setting, I propose the regression forests-based skeletal spine pose estimation.

The rest of this chapter is organized as follows: Section 4.2 introduces the overview of our framework. Section 4.3 introduces the spin pose estimation procedure using the head tracker and the poselet-regressor. Section 4.4 introduces the multiple body orientation classifiers conditioned by spine angle prior. Section 4.5 is the evaluation of our method with American football and soccer scenes. Section 4.6 is the conclusion of this chapter. Related work of this chapter was already discussed in Section 2.4.

4.2 Overview of Proposed Framework

In this section, I will show the overview of our framework for estimating the upper body pose of the player: the *spine pose* and the *body orientation*. Figure 4.4 shows the flowchart of our framework. Our framework is composed of the following three steps:

1. Tracks the head region in each frame with the head tracker [3] to estimate the head center location in each frame.

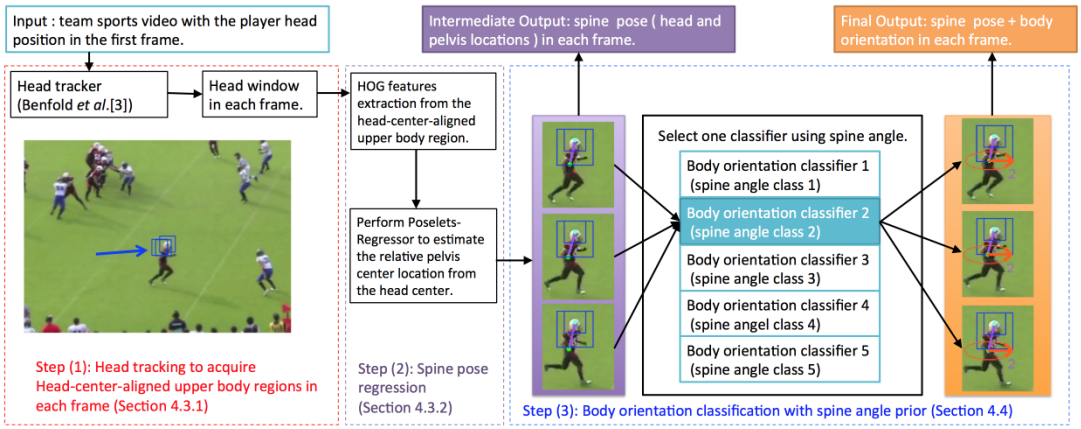


Figure 4.4: Proposed framework.

2. Estimates the relative pelvis position against the head center in each frame with the poselet-regressor, using the HOG features of the upper body region aligned to the head center as input. This step results in estimating the spine pose.

3. Estimates the body orientation with a classifier selected by the spine angle value of the player. (Optionally: estimates the head orientation with a head orientation classifier in the same way as in our previous work [119].)

Steps (1) and (2) estimate the spine pose $\mathbf{s}_t = (\mathbf{h}_t, \mathbf{p}_t)$ where $\mathbf{h}_t = (x_t^h, y_t^h)$ is the head location and $\mathbf{p}_t = (x_t^p, y_t^p)$ is the pelvis location in each frame t. Step (3) estimates the body orientation $\mathbf{o}_t^b \in \{0, 1, \ldots, 7\}$ of the player in each frame (Figure 4.3 (c)). The spine angle calculated from the spine pose is used to select one corresponding upper body orientation classifier f_s^b from multiple body orientation classifiers for each spine angle range. In other words, the spine pose acts as a mediator between the steps (1) and (2) and step (3). I will define the procedures of steps (1) and (2) in Section 4.3, then define the procedure of step (3) in Section 4.4.

4.3 Head Tracking and Spine Pose Estimation with Poselet-Regressor

To estimate the relative pelvis position $\mathbf{p}_t = (x_t^p, y_t^p)$ from the head center location $\mathbf{h}_t = (x_t^h, y_t^h)$ estimated by the head tracker at each frame t, I propose a body spine pose regressor, which I call the *poselet-regressor*. Using the head tracker and our poselet-regressor, our framework can estimate the 2D spine pose of the player $\mathbf{s}_t = (\mathbf{h}_t, \mathbf{p}_t)$ at each frame t of the video by regression (see Figure 4.5).

With the global coordinate head center location $\mathbf{h}_t = (x_t^h, y_t^h)$ in a video frame (Figure 4.6 (a)), our poselet-regressor first calculates the multi-level HOG feature $\mathbf{x}_t^b$ within the upper body region around the head center location $\mathbf{h}_t$ (Figure 4.6 (b)). Then it estimates the relative pelvis position $\mathbf{p}'_i = (x_i'^p, y_i'^p)$ from the local coordinate head center $\mathbf{h}'_t = (0, 0)$ (Figure 4.6 (c)) using $\mathbf{x}_t^b$ for the random regression forests input vector. In other words, I use the selected visual features from the upper body region for this regression.

4.3.1 Head Tracking

To estimate the head center locations in each frame, head tracking for one subject player in a team sports video is performed. The head tracking approach proposed by Benfold *et al.* [3] to track the head region is used here. This method tracks the head region of a person using tracking-by-detection with a Kalman filter. It uses a SVM (support vector machines) head rectangle detector with HOG features as the likelihood of Kalman filter and uses local feature tracking results to predict the next state. In our experiments, I trained scene-specific 24×24 head detectors for each scene. I regard the center of the tracked head regions in each frame t as the head location $\mathbf{h}_t$ used in the later steps.

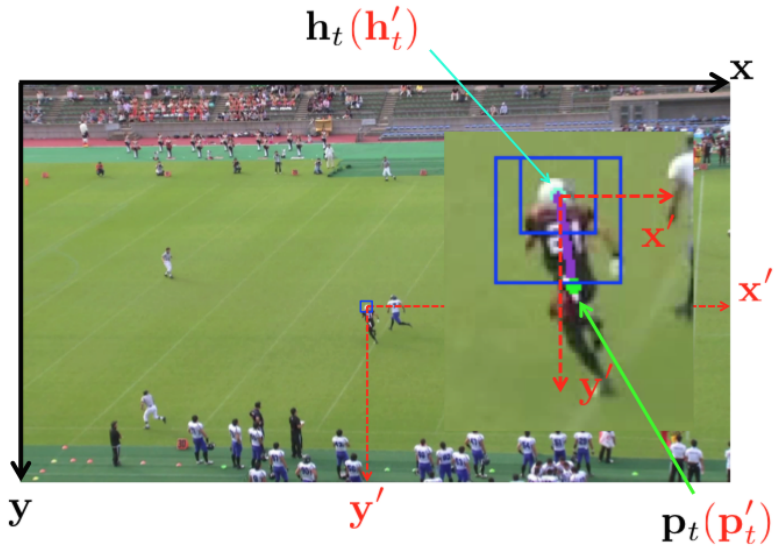


Figure 4.5: Local coordinate system for the poselet-regressor and the global coordinate system for the head tracker. The black axes **x** and **y** are the global coordinates and the red axes $\mathbf{x}'$ and $\mathbf{y}'$ in the magnified player image are the local coordinates. Pelvis position estimation is performed in these local relative coordinates.

4.3.2 Poselet-Regressor of Spine Pose

After the head region is estimated in each frame of the video in the first step, I employ the poselet-regressor to estimate the 2D spine pose, which consists of the 2D head position $\mathbf{h}'_t = (x'^h_t, y'^h_t)$ and the 2D pelvis center position $\mathbf{p}'_t = (x'^p_t, y'^p_t)$ in a local coordinate system whose origin $\mathbf{h}'_t = (0, 0)$ is the global head location $\mathbf{h}_t = (x^h_t, y^h_t)$ in each image (Figure 4.5). The upper body region for the poselet-regressor is located 20 pixels to the left and 12 pixels above the head center location $\mathbf{h}_t$ (Figure 4.3 (a)).

The proposed poselet-regressor does not try to detect segments of the subject person like the poselets detector [30, 31] or FMP detector [11]. Instead, our poselet-regressor estimates the continuous change of the relative joint position (pelvis center) from the center position (head center) using the selected discriminative features within the head-center-aligned upper body region HOG appearances. Adopting this design of *relative joint location estimation* by discarding the detection and segmentation ability of the original poselets [30, 31], our poselet-regressor can obtain the relative movement of one joint location from the center joint of the poselets window in the upper body visual feature space. Our poselet-regressor can also be regarded as the regressive version of the label-grid classifier [102] in Chapter 3, a visual grid classifier of the lower body joint location from the pelvis center with HOG-grid resolution. The center of alignment of the HOG features window in the label-grid classifier [102] is the pelvis center, while the alignment center of the poselet-regressor is the head center position in this paper.

Given this head-center-aligned upper body region at frame t, the poselet-regressor estimates the (regressed) 2D pelvis center location $\mathbf{p}'_t = (x'^p_t, y'^p_t)$ from the head center location $\mathbf{h}'_t = (0, 0)$. $\mathbf{p}'_t$ can be also regarded as an offset vector from the local origin $\mathbf{h}'_t$. I train the poselet-regressor $f^s(\mathbf{x}^b_t)$ as regression forests [93] to estimate $\mathbf{p}'_t$ with the selected features from the whole HOG feature vector $\mathbf{x}^b_t$ in the upper body region:

$$\hat{\mathbf{p}}'_t = f^s(\mathbf{x}^b_t) \tag{4.1}$$

Figure 4.7 shows some example results. Here, I assume that the head center locations in each frame were already tracked by the head tracker in Section 4.3.1. I then use the poselet-regressor to estimate $\mathbf{p}'_t$ in each frame.

I adopt the same feature vector as [26], a three-level pyramid of HOG features within the upper body region as a D dimensional feature vector $\mathbf{x}^b_t \in \mathbb{R}^D$ from the $W \times H$ window. The

block size of the HOG is 2×2 at every level. In the same way as in [26], a dimensionality reduced image (by PCA) is used as an input for HOG features calculation.

I name our relative pelvis location regressor as the poselet-regressor because it can be regarded as a regressive version of the poselets framework [30]. While the original poselets are *detectors*, our poselet-regressor trains a regressor of the relative pelvis offset using the upper body visual features, whose head positions are aligned. Note that one poselet detector is typically trained from people with many types of clothing and hair styles, while our poselet-regressor (in this paper [1]) is trained from the upper body regions of different players and poses wearing only one specific American football or soccer uniform type (see our experimental setting in Section 4.5).

Training the Poselet-Regressor.

The poselet-regressor is trained from the dataset $\mathcal{D}_{pose} = \{(\mathbf{x}_i^b, \mathbf{p}_i'), i = 1 \ldots, n_b\}$, where n_b denotes the number of samples in the dataset $\mathcal{D}_{pose}$, $\mathbf{x}_i^b$ denotes the feature vector from the upper body region aligned with the head center, and $\mathbf{p}_i'$ denotes the pelvis center offset from the head center location in local coordinates. As already mentioned, I use regression forests [93] as the poselet-regressor to train the local pelvis center location $\mathbf{p}_i'$ in a continuous 2D image space using the dataset $\mathcal{D}_{pose}$. Each sample $(\mathbf{x}_i^b, \mathbf{p}_i')$ in $\mathcal{D}_{pose}$ is collected and labeled from the videos from the same match and the players from the same team.

Optionally, the original training dataset $\mathcal{D}_{pose}$ is augmented to the $\mathcal{D}_{pose}^{aug}$ with some slides of the head center position to make the poselet-regressor (and the body orientation classifiers in Section5) recognize a slanted upper body appearance. The slide vector $\mathbf{v_s} = (x_s, y_s)$ where x_s denotes the x-axis slide value of the head center position and y_s denotes the y-axis slide value of the head center position. By using slide vectors, head center positions are augmented while the images in the original $\mathcal{D}_{pose}$ remain the same. Since our body pose estimators depend on the head-center-aligned upper body appearance, the drift of the head tracker often provides a little slanted upper body appearance. Although HOG [24] has local deformation invariance through block histogram quantization, our data augmentation procedure will provide additional robustness to the not-well-aligned head tracking results (I test this setting on women's soccer scenes in Section 4.5.

[1] While poselet-regressor can be learned from people with various types of clothing, I only use it for one specific clothing type for the team.

Potential advantage of the Poselet-Regressor.

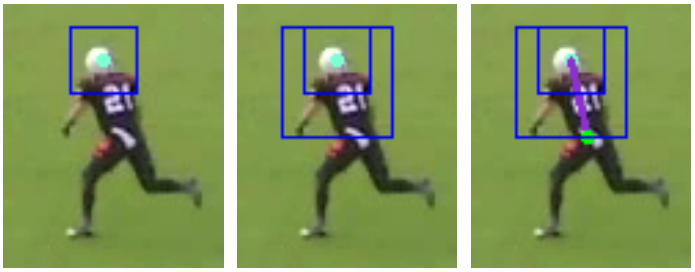
Figure 4.7 shows some example results from our poselet-regressor in our experiment (Section 4.5). These results show us two advantages of the design of the poselet-regressor. The first advantage is that the poselet-regressor realizes spine pose estimation for all types and views of sports player poses, even when some part-occlusions happen (Figure 4.7), because it achieves regression using only randomly-selected features within a holistic upper body region. As discussed in Chapter 2, part-detector based methods can estimate poses when the pictorial structure of *all* parts can be found. Conversely, our poselet-regressor uses only head-center-aligned upper body region appearance so that every upper-body visual features (including part-occluded poses and side poses) can be trained from the training images. This is a necessary trait for the human pose estimation methods for team sports videoss, because players run right or left with side-view poses and their arms often disappear from images because of part-occlusions.

The second advantage of the poselet-regressor is a separate and sequential estimation of *joint* locations via relative joint location estimation (in local coordinates). While FMP [11] and other *part*-based detectors need to decide the locations of all local parts simultaneously, our poselet-regressor enables part-location estimation one by one with global appearance. This flexibility evades the necessity of part localization for bone estimation (in our case, spinal bone) and also achieves the spine pose estimation that most previous work does not primarily focus on (or which is often ignored). Moreover, by integrating it with head tracking (in this paper), a better alignment of the body appearance is produced, which makes the pose estimation problem simpler [1], and can be done even while tracking the person. The head region has good *rigidity* for tracking and achieving good alignment for the pose estimators, while the previous body orientation estimation work depends on the pedestrian detection window as alignment.

4.4 Multiple Body Orientation Classifiers with Spine Angle Prior

The third and final step in our framework is to estimate the body orientation using the body orientation classifier with a corresponding spine angle range. I train each n_s upper body orientation estimator as eight-class random decision forests $\mathcal{F}^b = \{f_s^b, s = 1, \ldots, n_s\}$ with a training dataset from one team in a specific scene. Each body orientation classifier f_s^b is responsible

[1] Note that I insist on simplicity for (only) the pose estimation problem during tracking from low-resolution surveillance videos.



(a) Head region tracked by the head tracker.

(b) Upper body region (larger rectangle) used by poselet-regressor to for poselet-regressor.

(c) 2D spine estimated by poselet-regressor to estimate pelvis center position.

Figure 4.6: Estimating procedure of spine pose with head tracker and poselet-regressor.

for estimating the body orientation $\mathbf{o}_t^b \in \{0, 1, \ldots, 7\}$ (Figure 4.3 (c)) within the corresponding spine angle range class using the input feature vector $\mathbf{x}_t^b$ at frame t:

$$\hat{\mathbf{o}_t^b} = f_s^b(\mathbf{x}_t^b) \tag{4.2}$$

After tracking the head region and estimating the spine pose in the two previous steps (Section 4.3), I select a s-th class f_s^b from $\mathcal{F}^b$ to estimate the body orientation according to the spine angle θ_t of the player at frame t (Figure 4.8).

I use random decision forests [9] to train each upper body direction classifier using the subdatasets divided by the spine angle value (see Figure 4.9). I use the same feature window size of the poselet-regressor calculated from $W \times D$ upper body region (Section 4.4.1) for the two-level HOG features of the multiple body orientation classifiers.

4.4.1 Learning Multiple Upper Body Orientation Classifiers by Dividing the Dataset According to the Spine Angle

To estimate the body orientation of the player, I use one classifier f_s^b selected from n_s classifiers $\mathcal{F}^b = \{f_s^b, s = 1, \ldots, n_s\}$, where f_s^b is a body orientation classifier for each spine angle class s (Figure 4.8).

To train each f_s^b, I first prepare the dataset $\mathcal{D} = \{(\mathbf{x}_i^b, \mathbf{o}_i^b, \mathbf{s}_i), i = 1 \ldots, n_b\}$ as in Section 4.3.2, where $\mathbf{x}_i^b$ is the upper body region feature vector, $\mathbf{o}_i^b \in \{0, 1, \ldots, 7\}$ is the body orientation label, and $\mathbf{s}_i = (\mathbf{h}_i, \mathbf{p}_i)$ is the spine pose label of the i-th sample in the dataset, respectively. After the learning procedure, each f_s^b uses different features selected from the same upper region feature

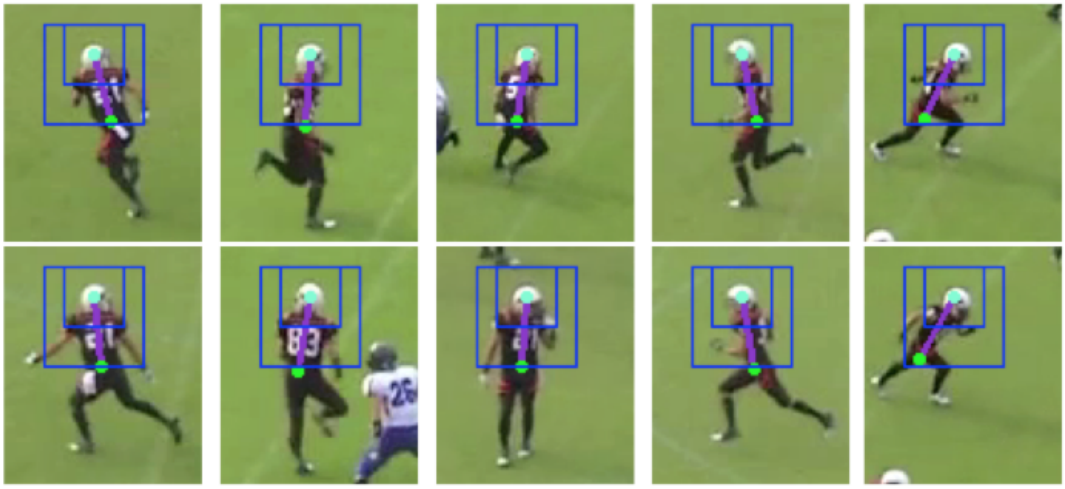


Figure 4.7: Example results of the relative pelvis position estimation using poselet-regressor.

$\mathbf{x}_i^b \in \mathbb{R}^D$. I also define the spine angle θ_i on the image plane as the angle against the x-axis direction. This spine angle θ_i can be calculated from the spine pose estimated with the 2D spine pose $\mathbf{s}_i$ (see Figure 4.8).

Next, I divide $\mathcal{D}$ into n_s-subdatasets $\{\mathcal{D}_s, s = 1, \ldots, n_s\}$ according to the angle value θ_i of each i-th instance in $\mathcal{D}$ (Figure 4.8). Spine angle θ_i space is separated into n_s regions, which I call *spine angle classes*, according to the spine angle value θ_i calculated from $\mathbf{h}_i, \mathbf{p}_i$. With each $\mathcal{D}_s$, I learn f_s^b as random decision forests. This dataset preparation procedure is shown in Figure 4.9.

I use $n_s = 5$ by default as showed in Figure 4.8. After the preliminary tests with our experimental datasets, I decided to divide the spine angle θ_i space into the following five spine angle classes:

$$s = \begin{cases} 1 & \text{if } 60 > \theta_i \\ 2 & \text{if } 80 \geq \theta_i > 60 \\ 3 & \text{if } 100 \geq \theta_i > 80 \\ 4 & \text{if } 120 \geq \theta_i > 100 \\ 5 & \text{if } \theta_i > 120 \end{cases} \tag{4.3}$$

At test time, after the value of s was selected using this conditions, the corresponding classifier f_s^b was used to estimate the body orientation.

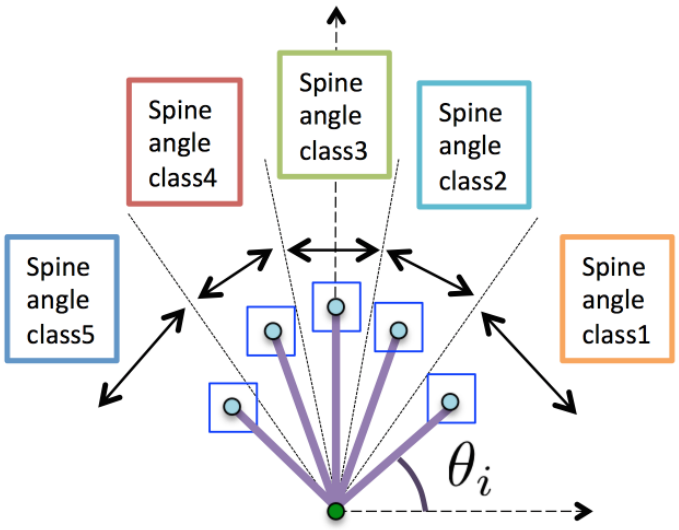


Figure 4.8: Spine angle classes. The blue circle is the head center **h** and the green circle is the pelvis center **p** of the subject player. The spine angle range of the training dataset is divided into five spine angle classes

This spine-driven dataset grouping both at training time and test time makes it easier for each random decision forest to select more discriminative split functions at each node rather than using the whole dataset. The reason is that images in one grouped subdataset $\mathcal{D}_s$ are more similar and thus it is easier to automatically select the informative (different) feature dimensions for random forest learning (see the average images in each subdataset $\mathcal{D}_s$ for each spine angle class in Figure 4.9).

Our idea of using spine angle value as a conditional prior for selecting body orientation classifier was inspired by the conditional regression forests in [110], but there is a difference. While [110] also tries the conditional prior of the head orientation by selecting the T trees from one holistic random decision forest for all the head orientations (conditions) in the experiments, our method trains multiple independent random decision forests for each spine angle range separately and do not use any trees from the different conditions. The reason for adopting only this approach is that the upper body appearance does not have continuous (manifold-like) feature space along the spine angle because arm and head appearances are sometimes inconsistent with the spine angle changes.

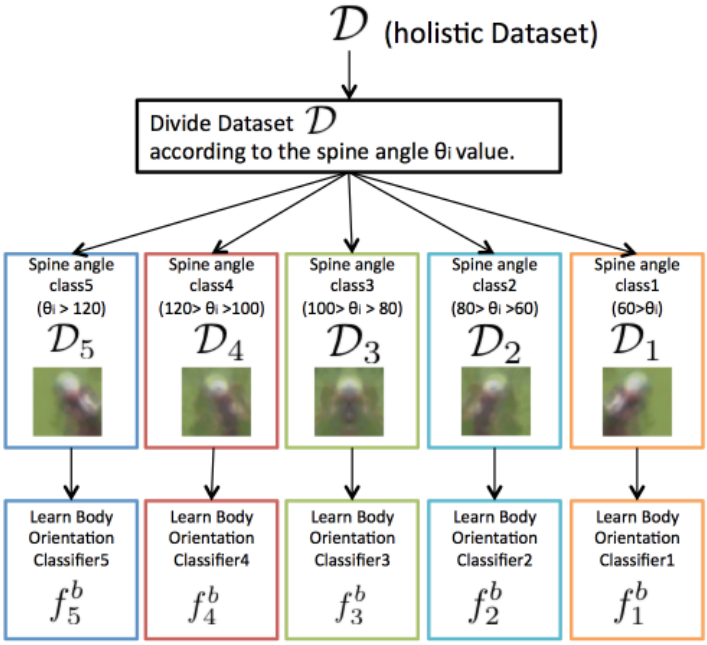


Figure 4.9: Learning multiple body orientation classifiers (random decision forests) by grouping datasets into the subsets having the same spine angle range. This conditional classifiers learning makes the random decision forests easier to select discriminative features for body orientation classification from only the spine pose aligned HOG features in each D_s.

4.5 Experiments

I evaluated our framework with videos from an American football game and a women's soccer game. I performed experiments for each pose estimation step, namely (1) head and pelvis center estimation with head tracker and the pelvis center (Section 4.5.2), and (2) the body orientation estimation (Section 4.5.3). All videos were captured at a high place in the stadium with fixed cameras at 29 fps. In the American football videos, horizontally moving players are mainly shown. However, in the women's soccer videos, players often move diagonally (to the goal or the opponents). Hence, I can expect different body orientation statistics for each scene.

In each scene, I prepared our experimental data by dividing videos into a test dataset and a training dataset. The datasets were prepared with images and manually annotated labels of the spine pose and the body orientation. The American football videos were captured from one match of Panasonic IMPULSE [1], which is a Japanese professional American football team.

[1] http://panasonic.co.jp/es/go-go-impulse/

The women's soccer videos were captured from the match Waseda-Keio Game of the Keio University womens soccer team[1].

I evaluate each of the three steps of our method in the following three subsections (Section 4.5.1, 4.5.2, 4.5.3). I will focus only on the poses of the black-uniformed players in the American football videos and focus only on the poses of the brown-uniformed players in the women's soccer videos (see the figures for those uniform colors).

I prepared a test video dataset $\mathcal{D}_{test}$ with 12 test videos for the American football game and prepared $\mathcal{D}_{test}$ with 10 videos for the women's soccer game, with 22 test video sequences in total. Each test video sequence in both games is composed of 80 frames, and I track one player in each video to evaluate our framework. To evaluate the advantage of using only the upper body region appearances for human pose estimation, some of the tests include frames with some lower body occlusions between players. In addition, to demonstrate the advantages of our body orientation classifiers conditioned by spine angle class, some of the tests include bending or twisting poses (where the upper body orientation and the lower body orientation (movement direction) are different). Later in Section 4.5.4, I discuss the effectiveness of our method for occlusion cases and bending poses by showing the visualized result images.

For each scene, a head detector (which I will not evaluate), the poselet-regressor of the pelvis center and the body orientation classifiers are independently trained from the training dataset $\mathcal{D}_{train}$ with manually labeled poses, which only includes team players from one specific team. In addition, to overcome the head-center drift of the head tracker in the soccer scene, I augmented the $\mathcal{D}_{train}$ of the soccer scene to $\mathcal{D}^{aug}_{train}$ to train pose estimators that also understand shifted examples. I made $\mathcal{D}^{aug}_{train}$ with slide vectors $(-8, 0), (8, 0), (0, 8), (0, -8)$.

To train the body orientation classifiers with the symmetric images and labels of the originally labeled samples, I resampled the symmetric samples using the following procedure. First, I made the dataset $\mathcal{D}_{train}$ in specific scenes (e.g., American football or women's soccer, in our experiments) by manually labeling the images from the training videos. Second, I made a flipped copy of $\mathcal{D}_{train}$ as $\mathcal{D}'_{train}$, which is composed of the flipped images and flipped labels from $\mathcal{D}_{train}$. Finally, I acquired the whole training dataset $\mathcal{D} = \{\mathcal{D}_{train}, \mathcal{D}'_{train}\}$ with symmetrically resampled images and labels. In the American football scenes, about 40 percent of the images were used in the test videos (tests 2,3,4,10 and 12)[2]. In the soccer scene, I com-

[1] http://keio-soccer.net/

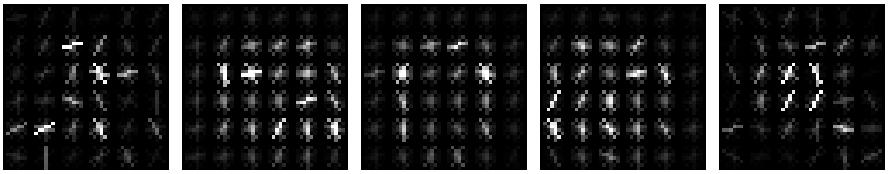
[2] Even though some test sequences were overlapped with the training dataset, most of the test images in each frame were different from the ones in the training dataset because tracking provides a shifted upper body window appearance based on the drift of the head tracker

pletely separated the test dataset and the training dataset. Note that again our goal was training scene-specific (or sport-specific) body orientation estimators that can deal with bending poses. Hence, I tried a supervised-learning approach as a first step toward unconstrained sport pose estimators.

I trained and tested our method for one specific team (with the same clothing but with different body shapes). $\mathcal{D}_{train}$ in the American football scenes includes 16334 (after reflection, original dataset includes 8167 samples) feature vectors $\mathbf{x}_i^b$ calculated from images and their labels $(\mathbf{o}_i^b, \mathbf{s}_i)$. $\mathcal{D}_{train}$ in the women's soccer scene includes 1053 examples, which is small for random decision forests training. For this reason, I augmented the original with reflections and four slide vectors. After data augmentation, I obtained $2822 \times 5 slides = 5265$ examples in total). I used a 48×48 upper body region for the American football scenes and a 64×64 upper body region for the women's soccer scenes, from which I calculated the feature vectors for both the poselet-regressor and the body orientation classifiers. For the body orientation classifiers of the women's soccer scene, I change the center of the upper body region to the pelvis center estimated by the poselet-regressor while the center of the upper body region for the classifiers of American Football scene is the head center estimated by the head tracker (larger window shows the regions for body orientation classifiers in each figure in this paper). And I also make the upper body region size for the soccer scene body orientation classifiers to 64×64 in order to include arm regions of all training samples.

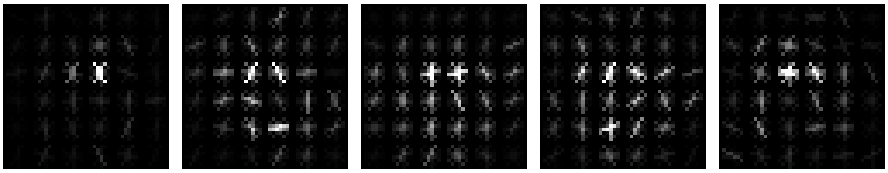
To train five body orientation classifiers in each spine angle range, I divided the training dataset $\mathcal{D}_{train}$ (or $\mathcal{D}_{train}^{aug}$ in the soccer scene) into five subdatasets $\{\mathcal{D}_s, s = 1, \ldots, 5\}$ and trained each spine angle class classifier f_s^b with D_s independently as in Section 4.4.1. In the American football scenes, each $\mathcal{D}_s$ had 934, 4005, 6456, 4005, and 934 examples (16334 in total) respectively. In the women's soccer scenes, each $\mathcal{D}_s$ had 336, 437, 1216, 437, and 336 images (5265 in total), respectively. For training the American football scene poselet-regressor, $\mathcal{D}_{train}$ with 16334 examples was used. For training the women soccer scene poselet-regressor, $\mathcal{D}_{train}^{aug}$ with 5265 was used. The feature importances of HOG features selected by random decision forests with American football dataset are visualized in Figure 4.10 and the feature importances of random decision forests trained from Women soccer dataset are also visualized in Figure 4.11.

I performed the evaluations for players on the lower side of the field in each scene so that the image scale of the players became almost the same. For the same reason, I also collected training examples from the players who played on the lower side of the field. As a result, I



(a) spine angle class 5. (b) spine angle class 4. (c) spine angle class 3. (d) spine angle class 2. (e) spine angle class 1.

Figure 4.10: Visualization of the importances of HOG features for each body orientation classifier of American football scene.



(a) spine angle class 5. (b) spine angle class 4. (c) spine angle class 3. (d) spine angle class 2. (e) spine angle class 1.

Figure 4.11: Visualization of the importances of HOG features for each body orientation classifier of women soccer scene.

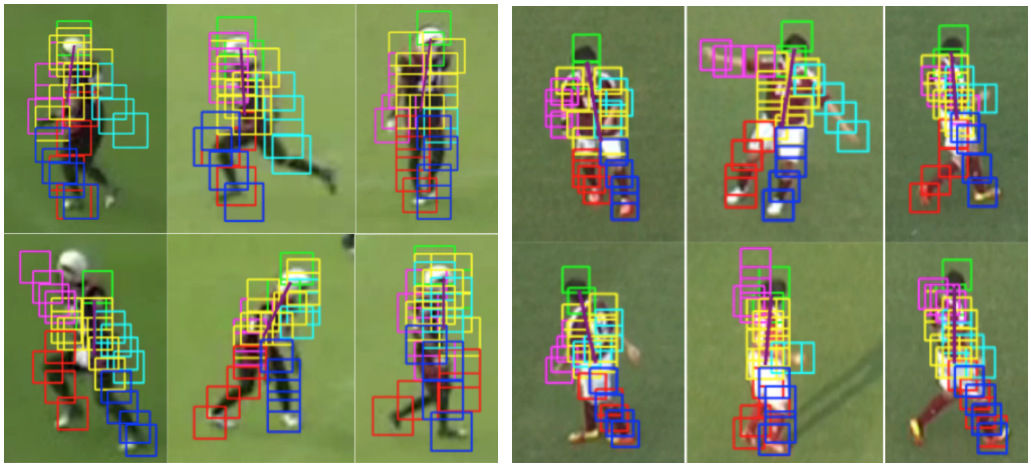
could only consider a small range of scales of the players in the experiments (and also the camera views and distance).

4.5.1 Head Tracking

To apply the head tracking algorithm from [3], I trained head detectors for each scene as linear SVMs using HOG features [24] with 4×4 cells within a 24×24 pixels window so that the head region was included in the $50--70$ percent of the window size. The first column in Table 4.1 is the result of the error in the head tracker for our test dataset. The error in the spine angle class will be evaluated in the next subsection.

4.5.2 Spine Pose and Spine Angle Class Precision

I evaluated the precision of the poselet-regressor using the test dataset from two perspectives: (1) precision of the poselet-regressor itself; and (2) precision of the assignment of the spine angle class. As a baseline of (1), I also evaluated the performance of the Flexible-Mixtures-of-Parts (FMP) [11] as the head center and the pelvis center estimator. Since FMP is a person



(a) American football scene. (b) Women's soccer scene.

Figure 4.12: Example results of skeletal pose estimation with FMP [11]. The purple line is the spine pose between the head center and the pelvis center.

Table 4.1: Average estimation error (Euclidean distance in pixels) of the head center and the pelvis center in each test dataset.

	American football	women's soccer
head center (ours)	3.99 (12 tests)	7.68 (11 tests)
head center ([11])	10.44 (10 tests)	7.40 (9 tests)
pelvis center (ours)	4.03 (12 tests)	6.25 (11 tests)
pelvis center ([11])	9.75 (10 tests)	7.69 (9 tests)

detector, I used the 200×200 image centered at the head position from the head tracker to (re-)detect the FMP for this evaluation. I used software and the default model of FMP provided by the authors of [11]. I resized the 200×200 image to 400×400 size so that the FMP model could detect the person with the trained person size.

The first row in Table 4.1 shows the average error of the head center locations estimated by the head tracker with our estimators and FMP [11]. The second row in Table 4.1 is the result of the pelvis center location estimated by the poselet-regressor from the tracked head locations in each frame and the results of FMP. Note that in both scenes, some tests are omitted for calculating the results of FMP (test 2,3,4 in American football and test 5 in women's soccer). The reason is that the subject player was not detected by the FMP because of inter-player occlusion. Figure 4.12 shows some example results of FMP detection on our test sequences.

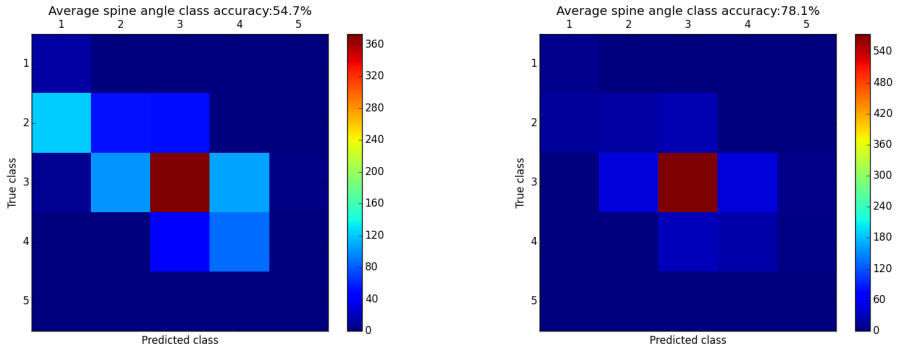
While our approach for estimating the pelvis center location is two-step estimation (steps (1) and (2)), the average error of the pelvis locations remains almost half the size of the cell size 8×8 of HOG features for the body orientation classifiers input vector. This makes HOG features pooling effective for estimating the same output body orientation class as long as the translation of the upper body window is small. In the American football scenes, our method predicted accurate spine pose even when the spine angle is acute or even when the pose is side-view. At the same time, FMP also shows accurate results for spine pose estimation (Figure 4.12). While the results for the two joints are good (for our body orientation classification in step 3), all parts of the FMP do not fit well for the subject person. While torso parts (yellow rectangles) and head parts (green rectangles) are well fitted to the players, arm parts and leg parts are not well fitted because of the hard occlusions or disappearance of those parts. Hence, FMP is not valid for our purposes, even though the head center and the pelvis center seem well fitted.

Next, I evaluated the precisions of the spine-angle classification of the whole test dataset $\mathcal{D}_{test}$. Figure 4.13 shows the confusion matrix of the results of the spine angle classification performed with the whole test dataset $\mathcal{D}_{test}$ in each scene type. American football videos include mainly standing poses (spine class 2,3, and 4) and few bending poses (spine class 1 and 5). There are some samples whose true class 3 is wrongly classified as one of the neighboring classes 2 and 4. The results for the women's soccer videos shows the accuracy of the spine angle classification: 78.1 percent, which is higher than the results for the American football scene (54.7 percent).

While my spine angle range strategy reduces the effects of the spine pose estimation error by discretizing the spine angle, higher accuracy of the assignment of the spine angle class s also strengthens the accuracy of the spine angle class prior for selecting the appropriate body orientation classifier f_s^b. In this sense, I would prefer the more precise poselet-regressor, while this might be difficult for our problem setting with very low-resolution videos.

4.5.3 Body Orientation Precision

I evaluated the precision of the body orientation classifiers (Section 4.4) using the test dataset. I compared the result of proposed body orientation classifiers to the result of our body orientation classifier in our previous work [119], which uses only one body orientation classifier for the whole training dataset. To test a fair comparison in terms of alignment, I trained body



(a) American football scenes.

(b) Soccer scenes.

Figure 4.13: Confusion matrices of the spine angle class estimation.

orientation classifiers for the women's soccer scenes with a pelvis-aligned window, where the pelvis is located at $(32, 48)$ from the top left corner of the 64×64 window (see Figure 4.17).

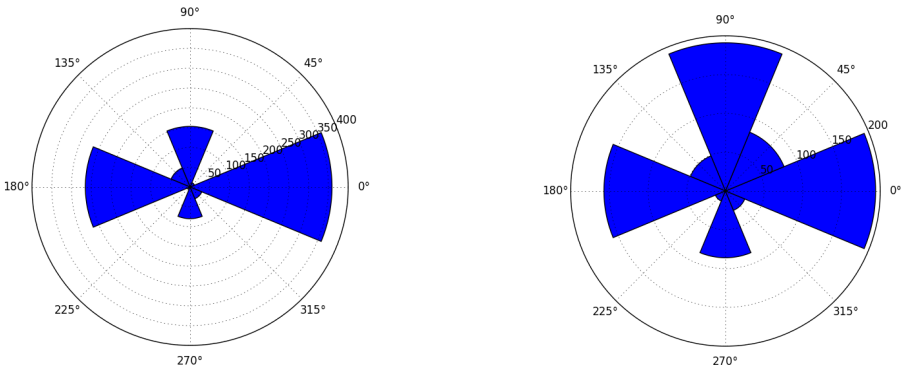
To compare with the body orientation classifier of our previous method for sports videos [119], I also trained one body orientation classifier with random decision forests using the overall training dataset $\mathcal{D}_{train}$ for American football scenes, $\mathcal{D}^{aug}_{train}$ for the women's soccer scenes. As described in Section 4.3.2, I used the input feature vector for random forests with the 3-level pyramid HOG with a PCA-reduced one-channel image for our proposed method. For the input features of our previous method, I used a one-level pyramid HOG as in [119].

For the American football scenes, I used the spine angle class boundaries $[60, 80, 100, 120]$ between two neighboring classes of equation 4.4.1. For the soccer scenes, I used different boundaries $[70, 80, 100, 110]$ because the spine angles of the soccer players are not so acute as those of the American football players.

I compared the results of the proposed body orientation classifiers with the spine angle prior and the body orientation classifier of [119] from two perspectives with the same tracked results of spine poses: a multi-class classification perspective and a body orientation angle estimation perspective.

Multi-Class Classification.

I first evaluated the proposed body orientation classifiers as a multi-class classifier. Figure 4.14 shows the body orientation class distribution, where 0 degrees is equivalent to class 4 and 180 degrees is equivalent to class 0 (see Figure 4.3 (c) for the class index assignment). Since most



(a) Body orientation distribution in 12 American football tests.

(b) Body orientation distribution in 10 women's soccer tests.

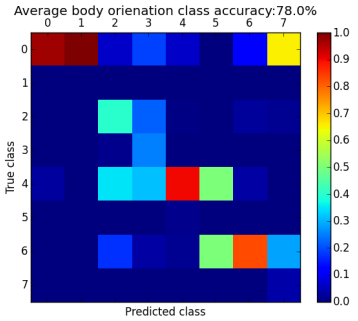
Figure 4.14: Body orientation class distribution (histogram) in each type of scene.

of the subject players (who are mainly running back and wide receivers) run horizontally in the American football scenes (Figure 4.14 (a)), most of the body orientations are in horizontal directions (class 0 (left) or class 4 (right)) and there are only few diagonal orientations. In the women's soccer scenes (Figure 4.14 (b)), the target brown clothing team is attacking to the right direction in the field. Most of the frames are in the right direction, with some diagonal orientations.

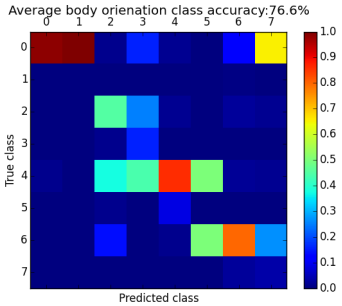
Figure 4.15 shows the confusion matrices of the classification results using the body pose classifier in [119] (Figure 4.15 (a) and (c)), and the body orientation classifiers proposed in this chapter (Figure 4.15 (b) and (d)) in each scene. The confusion matrices for the proposed method show slightly more precise results than those of [119], while the class prediction accuracy is almost the same. However, Figure 4.15 (c) shows a little more misclassification (i.e., predicted class 1 vs. true class 5 is salient in Figure 4.15 (c)) while Figure 4.15 (d) shows the more misclassification to the neighborhood classes.

Table 4.2: Average estimation error (in degree) of the body orientation in each scene dataset. The baseline is our previous work[119].

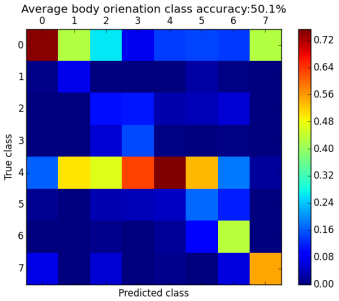
	Proposed	[119]
American football scenes	20.90	23.57
Women's soccer scenes	39.99	47.02



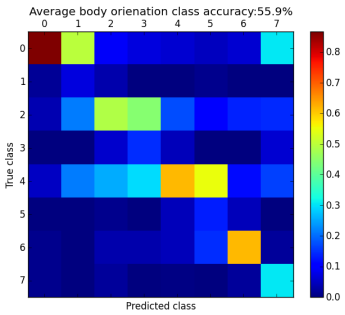
(a) Body orientation classifier of [119] in American football scenes.



(b) Proposed method in American football scenes.



(c) Body orientation classifier of [119] in women's soccer scenes.



(d) Proposed method in women's soccer scenes.

Figure 4.15: Confusion matrices of body orientation estimation results.

Orientation Angle Error.

As also evaluated in the other papers for head or body orientation estimation reviewed in Section 2.4, I evaluated the average angle error of the estimated body orientation angle from the same test results.

Table 4.2 shows the average estimation error of the body orientation angle in degrees for the test dataset $\mathcal{D}_{test}$ by the proposed method and our previous method [119] in each type of scene. Although the average errors of the proposed method shows that it performs the better than [119] in Table 4.2 in both scene types, this does not give us a good understanding of the overall results because the precision and accuracy are almost the same between the two methods. Hence, I will visualize many example results in specific tests and situations in the next Section 4.5.4 with further detailed discussion to provide evidence for each specific challenge.

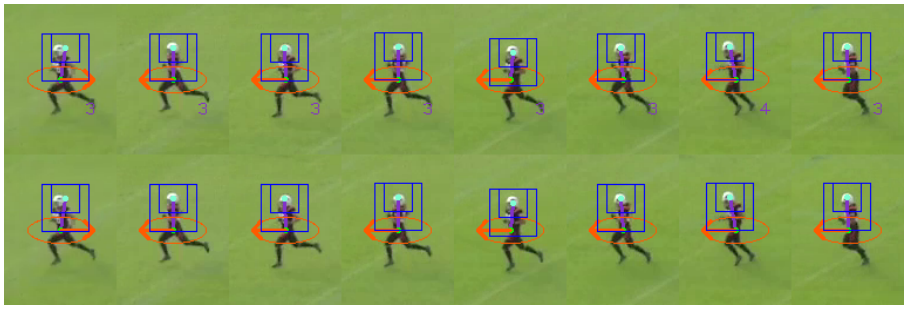
4.5.4 Discussions

Running Poses.

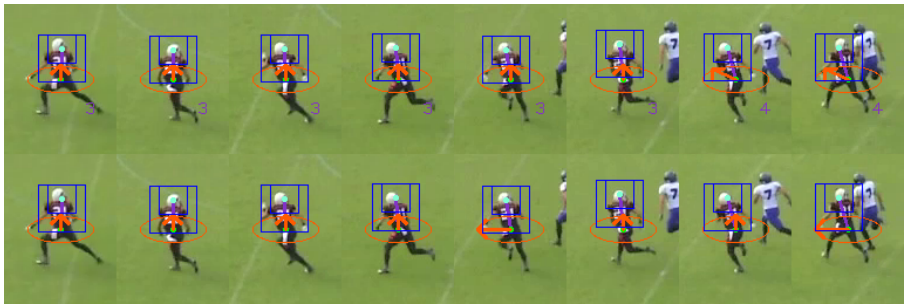
Since *running* is the most frequent action in team sports videos, the robustness of estimating running poses is the most important for evaluating the human pose estimation method for team sports. Figure 4.16 shows some results from the proposed method and [119] in seven consecutive frames in the test dataset while the player is running (tests 1,2,3,4 in the American football scenes). Figure 4.16 (a) (test 1) is a typical example of a standard straight running case, which occurs very often in team sports videos. Figure 4.16 (b) (test 3) shows the results when the movement direction of the player is different from the body orientation (the player is moving to the left while the body orientation is in the upward direction). This capability of the proposed framework is very important for team sports videos, where players often look at different directions from their movement direction. Figure 4.16 (c) (test 4) shows the results of twisting behavior (body rotation against the camera pose) during running. Even though [119] shows good results, the proposed method gives perfect results during the transition of body orientation (see 4th frame from the left on both rows in Figure 4.16 (d)). Figure 4.17 shows some key frames in the women's soccer tests. Figure 4.17 (a) and (c) (test1 and test7) shows running sequence examples. Compared with the American football scenes, the soccer scenes have more *diagonally running* players and diagonal body orientations: classes 1, 3, 5, and 7 in Figure 4.3 (c).

Occlusions and Using Only the Upper Body Region.

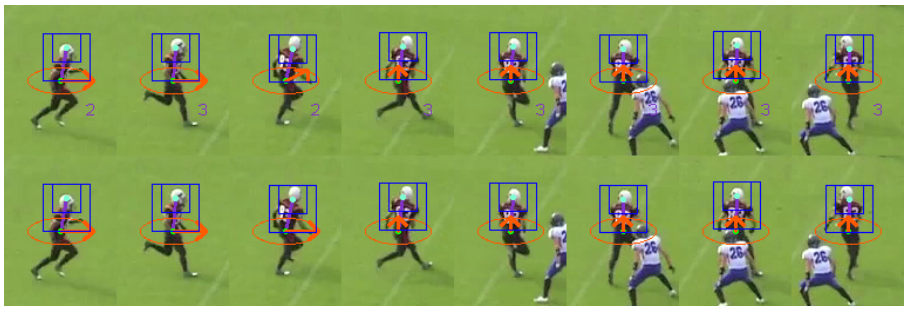
Figure 4.18 shows some cases with hard occlusions within the upper body region. Since the proposed method only uses the upper body region (larger blue rectangle) for body pose estimation, the upper body poses are estimated correctly (Figure 4.18 (b)). However, estimation of the upper body orientation tends to fail if most of the background becomes an unknown image pattern for the poselet-regressor and body orientation classifiers (Figure 4.18 (a) and (c)). Even though the background in my experimental videos consists of simple green flat texture and the random decision forests can select the features mostly of the foreground (player) HOG cells after the feature selection, the upper body pose estimation results become unstable when hard occlusions occur because I have not yet built the foreground-only selection. This is a very important problem in the application of my framework to other sports videos where the background consists of more complex textures.



(a) Results from test 1.



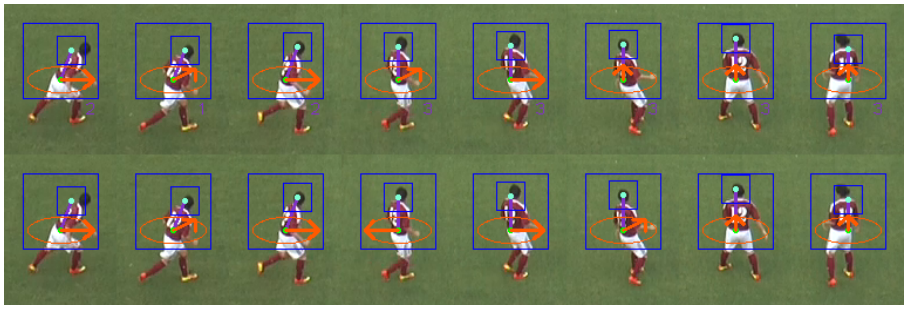
(b) Results from test 3.



(c) Results from test 4.

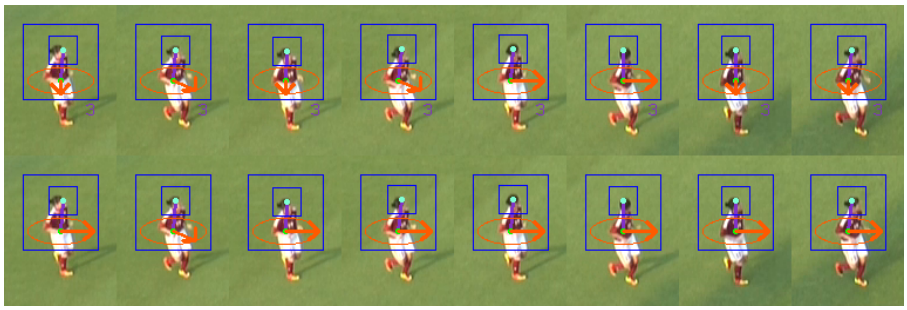
(d) Results from test 6.

Figure 4.16: Results from the American football scenes. The first row shows the results of the proposed method and the second row shows the method of [119].



(a) Result frames from test 1.

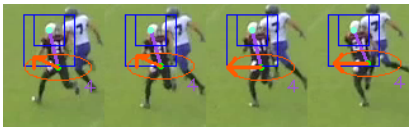
(b) Result frames from test 2.



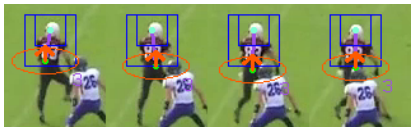
(c) Result frames from test 7.

(d) Result frames from test 10.

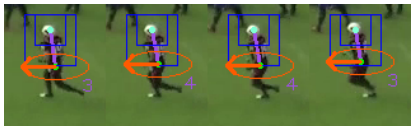
Figure 4.17: Results from women's soccer scenes. The first row shows the results of the proposed method and the second row shows the method of [119].



(a) Occlusion samples in test 3.



(b) Occlusion samples in test 4.



(c) Occlusion samples in test 10.

Figure 4.18: Results during occlusion between players.

Figure 4.17 (d) (test10 for women's soccer scenes) is also an example of the advantage of the usage of only the upper body region's appearance. My method can estimate the upper body pose of soccer players even as they interact with the ball with their legs, because the ball does not affect the estimation at all as long as it does not enter the upper body region.

Bending Poses.

Figure 4.19 shows the results of bending (side-view) poses. If the spine pose is estimated correctly, the upper body orientation is also classified correctly (Figure 4.19 (a) and (c)). If the drifts of the head tracker become large, the poselet-regressor tends to provide an incorrect pelvis center location and the upper body orientation classifiers tend to fail because of the wrongly estimated spine angle class (Figure 4.19 (b) and (d)). These spine pose estimations while bending from side-view monocular videos (not only for pedestrians) are novel outputs in the computer vision field, while having some errors in the experiments.

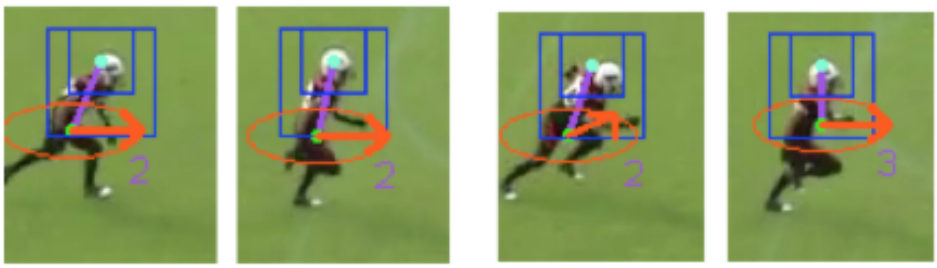
The Effect of the Alignment of the Upper Body Region and the Selected Features.

My body orientation estimation method depends on the alignment of the tracker and the selected features in each spine angle. Figure 4.20 shows some examples of the effect of the alignment of the head tracker and pelvis center estimation. In Figure 4.20 (a) and (b), the head center location is not good. This misalignment causes the misclassification of the body orientation in Figure 4.20 (b), while the body orientation is correct in Figure 4.20 (a) because the misalignment of the head position is small for 8×8 pooling of HOG features for both the pelvis poselet-regressor and the body orientation classifiers.

In Figure 4.20 (c) and (d), the head tracker is good enough. However, in Figure 4.20 (d), the pelvis estimation is not accurate and my algorithm selects the wrong spine angle class from the misaligned upper body region. There is a tradeoff between the selected features of each spine angles class vs. alignment of the upper body region and the precision of the pelvis center location.

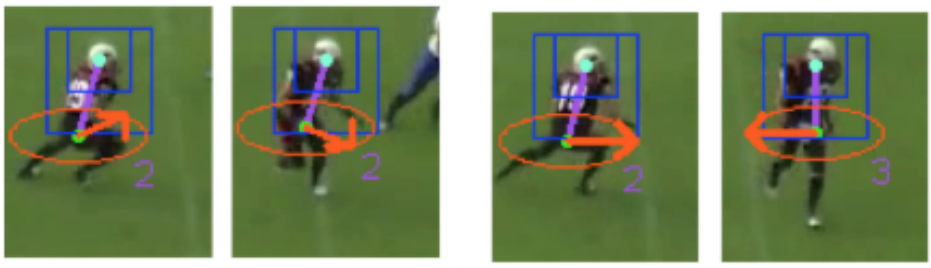
Models for One Specific Team.

Our evaluation for American football videos shows the robustness of our method mainly for running poses, which account for the vast majority of player poses not only in American football but also in all other team sports. Compared with FMP [11] or other frontal upper body estimators that must know the various types of clothing and hair styles, human pose estimators for team sports (such as our method) only need to know the appearance of the two team's uniforms for a specific match. In this sense, although our method uses fully supervised models for one specific team (or uniform), our experiments shows that our classifiers are robust enough to estimate the upper body poses of the target team players.



(a) Correct samples from test 6.

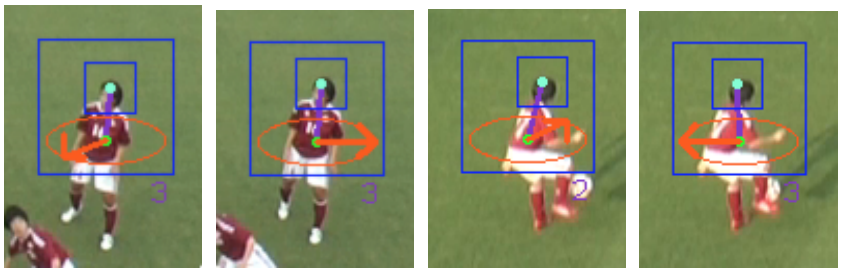
(b) Incorrect samples from test 6.



(c) Correct samples from test 12.

(d) Incorrect samples from test 12.

Figure 4.19: Sample results of bending poses from American football tests.



(a) Correct sample from test 5. (b) Incorrect sample from test 5. (c) Correct sample from test 8. (d) Incorrect sample from test 8.

Figure 4.20: Effect of the alignment of the upper body region for body orientation estimation.

4.6 Conclusion

This chapter proposed an upper body pose estimation framework for team sports videos, which estimates the upper body orientation and the spine pose of one player from the tracked and aligned upper body appearances and feature selection with random decision forests. Our method employs a scene-specific head tracker, a spine pose regressor (poselet-regressor of relative pelvis center location from the head center location), and body orientation classifiers conditioned by the spine angle value ranges. Both our poselet-regressor and the conditioned body orientation classifiers are trained from the player images of the same team, and can be used for the players wearing the same uniform (or performing the same sports actions in the other scenes). Our alignment-based body orientation classification, guided by the 2D spine pose, can predict not only the body orientation but also the 2D spine pose even when hard-occlusions or part disappearance occurs, because it uses a few selected features within the aligned upper body window. This alignment-based pose estimation framework, is suitable for side view running poses as the method in Chapter 3 [102] and suitable for upper body bending poses which both frequently appear in team sports videos.

Moreover, our previous method [119] of this chapter proposed a rough conversion of a 2D spine pose to a 3D pose by combining the 2D spine pose with horizontal body orientation recognition (Figure 4.3 (b)). This means that the upper body poses estimated by the proposed method can be also used for generating (approximate) 3D upper body pose information as [119] does.

Future work includes upper body orientation estimation using team contexts such as movement directions of all players on the same team or their common attending direction. In ad-

dition, 3D geometry of the scene and players should be considered to restrict the variation of the human appearance on images. Also, I would like to utilize the spine pose information estimated from our poselet-regressor as a mid-level feature for action recognition, such as the understanding of defensive behavior from bending poses or the team activity analysis as proposed in [5, 56, 97] for surveillance.

5

Integrated Estimation of the Upper-Body Pose and the Lower-Body Pose

In the previous two chapters, Chapters 3 and 4, I proposed two independent pose-estimation methods: one for the upper half of the body, and one for the lower half. In this chapter, I will propose a simple, integrated version of the two frameworks. The lower-joint estimation method used a label-grid classifier to classify the grid locations, and this has been replaced with a continuous regression that uses the poselets-regressor, as was done for the pelvis-center regression in Chapter 4. The resulting method gave a better estimate of the joint, due to the subpixel level capability of regression forests. I performed some additional tests of the revised lower-body joint estimation method and compared the results to those presented in Chapter 3; these tests included the intentional misalignment of the pelvis center, feature description from only the half-body region, and a real-world-based left or right labeling strategy.

5.1 Introduction

In this chapter, I will propose a final, integrated version of independent pose estimators, which were introduced in the previous two chapters. The integrated method first estimates the upper-body pose (the body direction and the spine pose) in the same way that this was done in the upper-body module (Chapter 4), and it then estimates the locations of the four lower-body

joints relative to the pelvis center by using the poselets-regressors, instead of using the label-grid classifier (Chapter 3).

The joint-location prediction approaches presented in the previous two chapters have the following properties in common:

- They use a few selected HOG features that are calculated for the whole- or half-body window; these features are selected while training the random forests.
- The player windows that are used to calculate the HOG features are aligned to the pelvis center (label-grid classifier) or the head center (poselets-regressor).
- They predict the locations of each joint relative to the aligned center (the position of each lower-body joint relative to the pelvis center in the label-grid classifier; the position of the pelvis center relative to the head center in the poselets-regressor).
- They disregard the part-detection and the part-segmentation strategies that are currently common approaches to human pose estimation techniques [8, 11, 75]; see Chapter 2 for details.

In a nutshell, in the previous chapters, I proposed joint-location estimation methods that use random decision forests to evaluate the globally aligned appearance of both the lower-body and upper-body joints. These methods employ either the whole- or half-body HOG to estimate the relative locations of the joints.

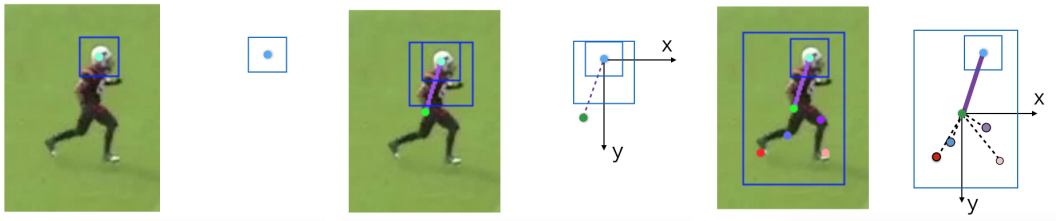
I will describe the procedure for the integrated approach in Section 5.2. In Section 5.3, I will present the results of experiments and compare the results obtained from the integrated method with those obtained from the separate ones.

In order to explore more-detailed experimental evaluations and settings, I also used additional experimental settings that were not yet performed in the previous two chapters; these include using only the lower-body HOG to predict the lower-body pose, and learning the joint-position regressors from the original left- and right-leg labels from the 3D world (note that the label-grid classifiers trained them with the labels from a 2D image).

Note that I will ignore the label-grid classifiers in these experiments so that I can compare the results of the regression forest learning paradigm with different settings. Please also note that the poselets-regressor provides continuous pixel-wise resolution of the estimated joint location, while the label-grid classifier only provides grid-wise resolution (an 8×8 window).

5.2 Integrated Approach

Figure 5.1 shows the three sequential steps used by the integrated approach to estimate the joint locations. First, I use the head-tracker method [3] to estimate the global head location $\mathbf{h}_t = (x_t^h, y_t^h)$ in each frame t in the same way as in Chapter 4. Next, I use the learned poselets-regressors to estimate the relative position of the pelvis center $\mathbf{p}'_t = (x_t'^p, y_t'^p)$ in frame t. Note that these two steps are the same as the method introduced in Section 4.3.2. Finally, I use the four poselets-regressors to estimate the position $\mathbf{l}_t'^j$ of the j-th joint relative to the location $\mathbf{p}'_t$ of the pelvis center; here, $j = rf, lf, rk, lk$, which mean the right-foot, left-foot, right-knee, and left-knee, respectively, for the *image coordinates*, such as were used in Chapter 3. In the first two steps, I used the same poselets-regressors that were used in Chapter 4. In the third step, I used two-level pyramid HOG features for the whole-body regions centered on the pelvis position estimated by the poselets-regressor in the previous step. Thus, the only difference between the integrated method and the separated methods is the third step, in which the label-grid classifiers are replaced by poselets-regressors when estimating the four lower-body joints.



(a) Step 1: The head region (blue rectangle) in each frame is tracked by the head tracker.

(b) Step 2: The 2D spine pose is estimated by the poselets-regressor. The larger rectangle indicates the region used for calculating the HOG features.

(c) Step 3: The lower-body joints are estimated by poselets-regressors for each joint. The larger rectangle indicates the region used for calculating the HOG features.

Figure 5.1: The integrated pose-estimation procedure. Steps 1 and 2 correspond to the method presented in Chapter 4, and Step 3 corresponds to the method presented in Chapter 3. After tracking the head center in each frame, I perform a two-step regression for the head, pelvis, and lower body. The poselets-regressor is used for estimating the position of the pelvis center relative to the head center. Finally, the other four poselets-regressors are used for estimating the positions of each of the joints relative to the location of the pelvis center, which was estimated in the previous step.

While both the label-grid classifier and the poselets-regressor employ the random forests model, the entropy functions used to split the data at each node are different in the classification

forest (label-grid classifier) and the regression forest (poselets-regressor) [25]. While classification forests use discrete class labels to optimize the information gain of the split functions at each node, regression forests use the information gain of continuous output labels to split samples at each node. This results in the approaches fitting different models, even though both use the original (continuous) joint location labels for optimizing the split functions.

As already shown by experimental results in Chapter 4, using the head tracker and the HOG features selected by random forests enables the pose to be estimated even when there are occlusions between players because random forests disregards non-selected HOG features within the upper-body regions. In addition, the integrated version can estimate the locations of the lower-body joints at pixel resolution, while the label-grid classifier can only estimate their locations at grid resolution (by default, the grid is 8×8 pixels).

5.3 Experiments

Using American football tests (1)–(10) also used in Section in Chapter 3, I performed experiments under each of the following three settings in order to compare the precision of the integrated approach with that of the poselets-regressors when determining the locations of the four lower-body joints. The settings were as follows:

- Setting 1: Integrated approach (Section 5.3.1).
- Setting 2: Integrated approach using only the lower-half HOG (Section 5.3.2).
- Setting 3: Integrated approach using original left and right labels (Section 5.3.3).

I then performed a test to determine how the alignment of the input window affects the performance; to do this, I translated the window using multiple values (Section 5.3.4).

For each of these three settings, I only changed the poselets-regressors for the four lower-body joints, and I used the same head-tracking results and the same estimation of the pelvis center for the pelvis poselets-regressor. Setting 1 is the standard setting for the integrated approach; for the HOG, it uses the whole-body window for which the center is aligned with the pelvis center. Settings 2 and 3 are provided to check obvious changes that were not explored in Chapter 3.

For these experiments, I used the augmented dataset used in Section 5.3 to train the poselets-regressors for each lower-body joint; this dataset was prepared in the way shown in Section

3.2.3 for training the four label-grid classifiers. I trained poselets-regressors for each of the four lower-body joints; I used the augmented dataset used in Section 5.3, which contains 13 player scales $s = \{0.70, 0.725, 0.75, \ldots, 0.975, 1.0\}$.

The aim of Setting 2 is to compare the results obtained when the whole-body appearance is used as input for the regression forests to that obtained when only the lower-half body appearance is used (Section 5.3.2).

The aim of Setting 3 is to ensure that my methods for labeling the left and right images and for training the label-grid classifier and poselets-regressor are valid; I wanted to determine if there would be a problem if they were trained by left- and right-leg labeling (Section 5.3.3).

For the resolution for the poselets-regressors, I used the two-level HOG features pyramid for the lower-body joints. Although the three-level HOG features were used for the label-grid classifiers in Chapter 3, here, a two-level pyramid was used for the lower-body joints in order to ensure that the resolution of the features was at the same level as that of the poselets-regressor that was used for predicting the pelvis center. The whole-body region, whose size is 64×96, was used for training the four poselets-regressors for Settings 1 and 3. The lower-half body region, whose size is 32×96, was used for training the four poselets-regressors for Setting 2.

Tests were performed with Settings 1–3. Table 5.1 shows the averaged errors at each joint location; the values are given as the Euclidean distance, which was also used as the error metric in the experiments in Section 5.3. Since the same tracker results and the same pelvis poselets-regressor results were used for all three settings, the head center and pelvis center errors are the same. Note that the integrated method (the poselets-regressors) uses regression forests to estimate the joint location $\mathbf{j}_t = (x_t, y_t)$ at frame t as a float image at subpixel precision.

In order to measure the effect of the cell-resolution size of the HOG, I added Setting 1a, which has a 4×4 cell size for the HOG, instead of the default, which is 8×8.

The following sections will show and discuss the results of each of the three settings.

5.3.1 Precision on Integrated Approach

The first and second columns in Table 5.1 shows the average precision of the results of tests (1)–(10) when using the whole-body HOG. All of the results for the lower-body joints had errors within 5.5-9.0 pixels; compare this to the test results of the label-grid classifiers shown in Tables and , which had errors around 8.0–10.00 pixels.

Moreover, while the label-grid classifiers can predict the grid location of the joint *within* the player region, the poselets-regressor can predict the joint location even outside of the player

Table 5.1: Average estimation error for each joint in the American football tests (1)–(10), with four settings. All errors are in pixels. The columns list the results for each setting (1–3 and 1a); the results are from using the poselets-regressor to estimate the location of the specified joint.

setting	1	1a	2	3
input region	whole body	whole body	lower body	whole body
joint label policy	image left/right	image left/right	image left/right	left/right leg
HOG cell size	8 ×8	4 ×4	8 ×8	8 ×8
Left knee	6.73	8.35	7.85	11.57
Right knee	9.07	10.18	9.98	18.04
Left foot	6.92	7.50	7.96	8.88
Right foot	5.56	6.72	6.69	10.44
Head	0.60			
Pelvis	7.06			

region, since it is trained as a regression model by using random regression forests. This characteristic helps the poselets-regressor predict the joint locations more precisely than does the label-grid classifier. For example, if the foot joint is located outside the player window when a player's legs are open, the label-grid classifier predicts the position of the foot joint only within the region (trained label-grid locations). However, the poselets-regressor can predict the foot joint location even when it is outside of the window.

5.3.1.1 HOG-Cell Size

The 1a column in Table 5.1 shows the results when using a 4×4 HOG cell size. Compared with the results when using 8×8 pixels, as shown in the first column, the errors are slightly greater, but the effect of the change in cell size does not seem to be large. Compared to the vector for the 8×8 cell, the feature vector for the 4×4 cell has more dimensions for the whole-person window, but the values of each dimension have smaller variance, because the small cell size decreases the contribution of each histogram. In other words, 4×4 cell HOG features within the same person regions becomes higher dimensional concatenated vector than that of 8×8 cell while the variety of the values in each orientation (dimension) becomes smaller.

Generally speaking, higher dimensional feature tend to become more discriminative. However, in this case, smaller pooling size of 4×4 cell becomes more sensitive to the alignment error of the person window because of the smaller invariance of the pooling in each cell. In

Section 5.3.4, I will see how shifting of the person window will affect the performance of the lower body poselets-regressors.

5.3.2 Results Using Only the Lower-Half HOG

The last column in Table 5.1 shows the average precision results for tests (1)–(10) when using only the HOG for the lower-half appearance.

Figure 5.3 shows an example of a result of the key frames with Tests (2), (6), (8), and (10) with Setting 2. As can be seen in Table 5.1, the total precision of the whole-body HOG is less than that of the lower-body HOG. These results show that the two approaches have different advantages and disadvantages.

For instance, the results of Test (2) (Figure 5.3 (a)) and Test (8) (Figure 5.3 (c)) show that the two methods predict similar locations for standing poses, even they use the different coverages of the input HOG region, and the features from the corresponding regions are selected in different ways. However, for Test (6) (5.3 (b)), in which the subject is bent over sharply, the lower-body HOG test (in the bottom row) shows a result that is slightly more precise than that of the whole-body HOG (in the top row). Since the lower-body HOG setting only takes into account the lower-body appearance, it is not affected during training or testing by changes in the upper-body appearance (in this case, the change due to bending over). I note that the whole-body HOG training seems to provide better results, even when the pelvis center error is not small; this is because it considers the whole-body appearance when predicting the location of each of the lower-body joints.

5.3.3 Results Using Left/Right Leg Labels

I performed a precision test for the integrated method using Setting 3. The poselets-regressors for the four lower-body joints were independently trained with the left-leg joint or right-leg joint labels. For example, the left knee joint was labeled in both the left and right halves of the player in Setting 3, while the proposed labeling scheme that was used in Settings 1 and 2 (proposed in Section 3.3) separately processes the left (in the image) and right sides for the whole training dataset.

Figure 5.2 shows the result for the key frames used for Setting 3. In the images in Figure 5.2, it can be seen that the locations predicted for the left and right joints are often almost the same. Although the average errors of Setting 3 are not much larger than those of Settings 1

and 2 (see Table 5.1), this figure shows that training with left and right leg labels is not a good strategy, and instead, these labels should be used in the proposed poselets-regressors (or the label-grid classifiers).

However, the proposed left/right labeling approach can only predict the locations of the left and right joints in the image; that is, it cannot say if the left joint is on the left or right leg. In Section 6.2, I will discuss this limitation in connection with the candidates for future areas of investigation.

5.3.4 Shifting the Input Window for the Lower-Body Joint Poselets-Regressors

To test the robustness when the alignment was changed for the input window of the lower-body joints poselets-regressors, another test was performed. In this test, the input windows for the lower-body joints poselets-regressors were intentionally shifted around the ground-truth position of the pelvis center. In each trial, the pelvis center and the input windows were moved from -8 to 8 at intervals of 4 pixels; this was done along both the x-axis and the y-axis of the image coordinate system. The ground-truth location of the pelvis center was used as a zero-move window location. That is, if the input window was shifted -4 pixels along the x-axis, the poselets-regressor took input for the multilevel HOG features with an error of -4 pixels along the x-axis .

Tables 5.2, 5.3, 5.4, and 5.5 show the error values resulting from shifts in the window (or shifts in the pelvis center). In these tables, the columns list the moves along the x-axis $(-8, -4, 0, 4, 8)$ and the rows list the moves along the y-axis $(-8, -4, 0, 4, 8)$. Figure 5.4 shows the results of this experiment for Tests (1), (3), (4), and (6). From these figures and tables, we can see that the misalignment of the window results in an error that is greater than that found with an accurately aligned window.

5.4 Conclusion

This chapter explored the integrated version of the proposed methods using the head tracker and the two-step poselets-regressors, which consist of the pelvis regressor and the regressors for the four lower-body joints. The experiments in this chapter, which can be viewed as variations on those in Chapter 3, in which the label-grid classifier is replaced with the poselets-regressor, compared the features obtained from the whole-body appearance with those obtained from the

Table 5.2: Left-foot errors in window-sliding test.

	-8	-4	0	4	8
-8	13.07	11.39	9.72	9.21	9.73
-4	10.87	8.93	7.62	7.37	7.88
0	9.60	7.27	6.40	6.40	7.08
4	9.96	6.59	5.91	6.53	7.92
8	11.10	7.88	7.14	8.45	11.07

Table 5.3: Right-foot errors in window-sliding test.

	-8	-4	0	4	8
-8	12.59	10.80	11.04	11.54	13.17
-4	11.00	9.23	8.79	9.43	10.93
0	10.16	8.14	7.57	8.14	9.62
4	11.28	8.76	7.66	8.12	9.31
8	14.06	10.88	9.88	9.97	11.24

Table 5.4: Left-knee errors in window-sliding test.

	-8	-4	0	4	8
-8	14.31	12.14	11.69	12.00	13.64
-4	11.25	8.53	8.08	8.79	10.85
0	9.49	6.18	5.91	6.57	8.92
4	9.92	6.08	5.60	6.38	9.36
8	12.44	9.28	8.33	9.43	12.41

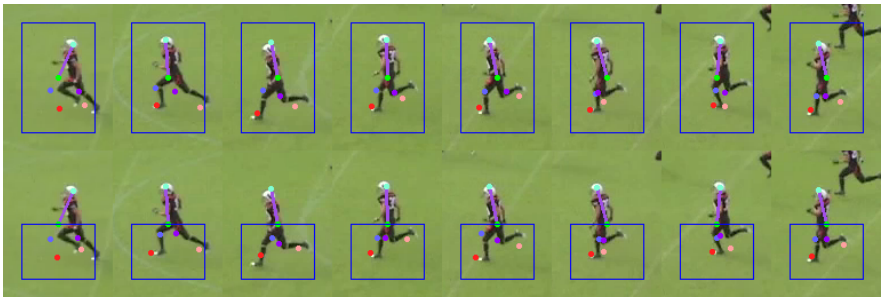
Table 5.5: Right-knee errors in window-sliding test.

	-8	-4	0	4	8
-8	12.19	11.36	11.21	11.15	12.30
-4	8.59	7.78	7.81	7.51	8.88
0	6.57	5.43	5.61	5.59	6.66
4	6.29	4.91	5.23	6.02	7.43
8	8.11	7.49	7.87	8.65	9.86

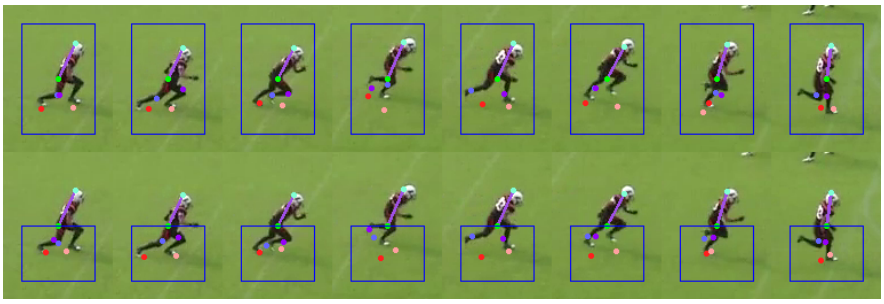
lower-body appearance. An additional experiment showed the validity of my left/right image-labeling approach. These experiments showed that the integrated approach has the benefits of both the lower-body and upper-body estimators and that it also provides a pixel-level lower-

body joint prediction, while the label-grid classifiers can only predict at a grid-level (8×8) precision.

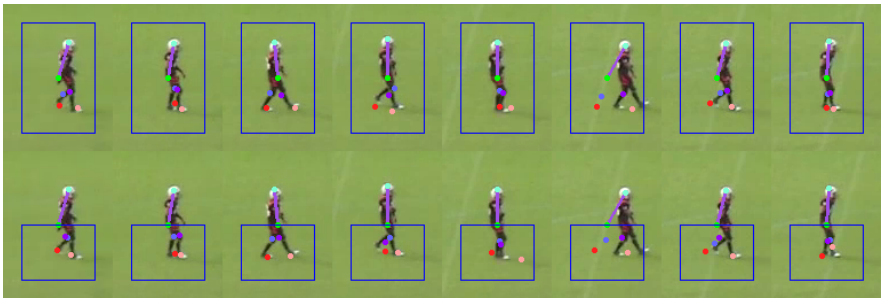
The predictions of the integrated method are more precise, but I have not yet explored the use of other priors, such as temporal tracking or motion features. In Section 6.2.1, I will discuss candidates for further study and ideas for improving the integrated method.



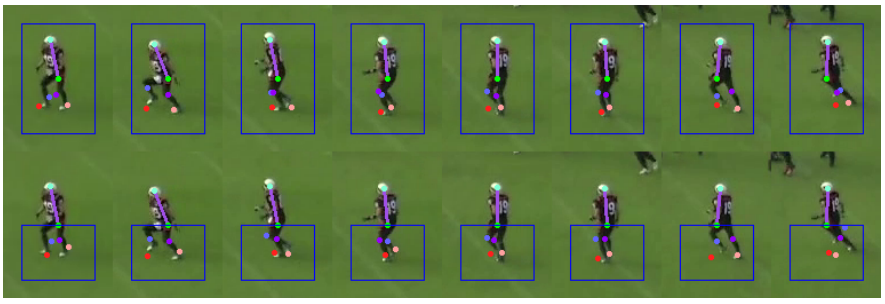
(a) Test (2).



(b) Test (6).

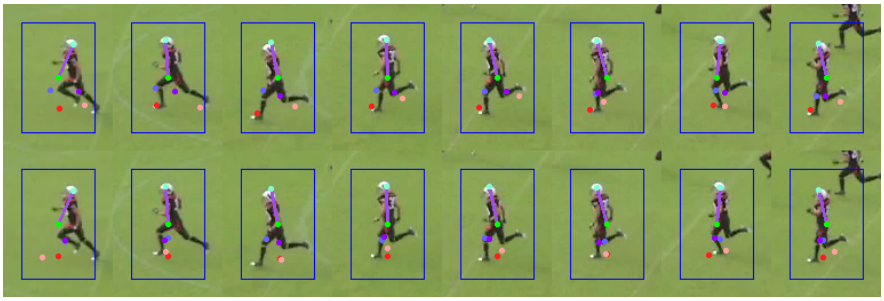


(c) Test (8).

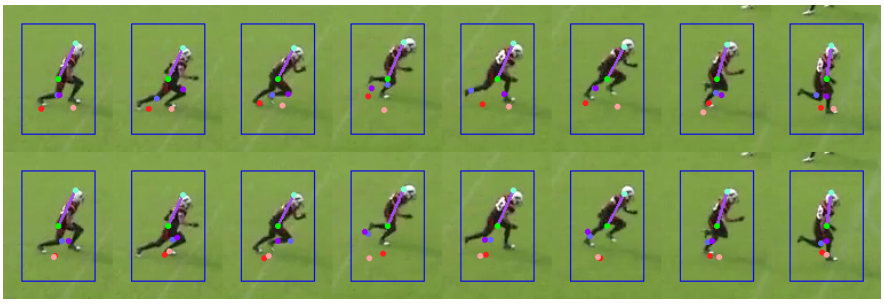


(d) Test (10).

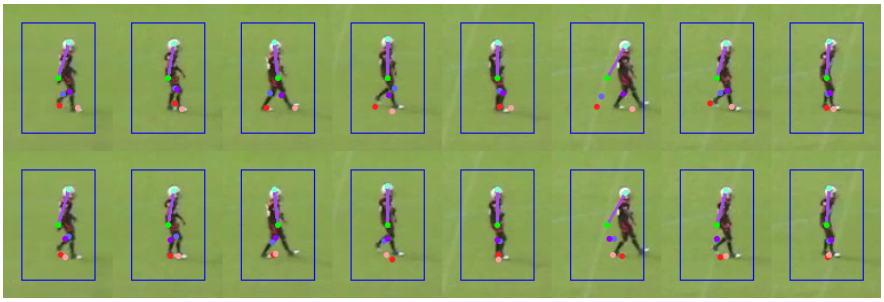
Figure 5.2: Whole body HOG vs. lower-half HOG in Tests (2), (6), (8), and (10). The images in the top rows of each test show the results obtained when using the features of the whole-body region, and the images in the bottom rows show the results obtained when using the features of the lower-body region. The blue rectangles indicate the feature regions used for the poselets-regressors.



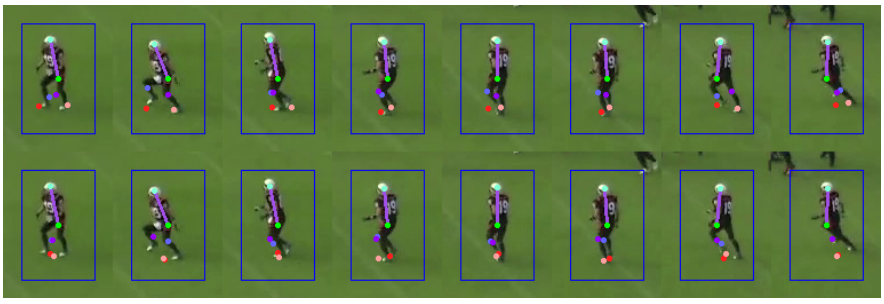
(a) Test (2).



(b) Test (6).

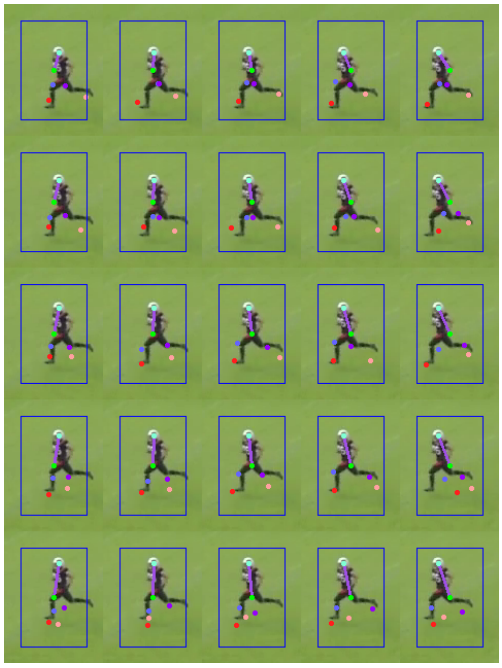


(c) Test (8).



(d) Test (10).

Figure 5.3: Results of labeling joints with left/right image vs. left/right leg in Tests (2), (6), (8), and (10). The images in the top rows for each test show the results obtained when using the features of the whole-body region, and the images in the lower row show the results obtained when using the features of the lower-body region.



(a) Example of result with Test (1).

(b) Example of result with Test (3).

(c) Example of result with Test (4).

(d) Example of result with Test (5).

Figure 5.4: Results of window-shifted tests. The columns show the moves along the x-axis $(-8, -4, 0, 4, 8)$, and the row show moves along the y-axis $(-8, -4, 0, 4, 8)$. The central figure shows the (0,0) move with the ground-truth location of the pelvis center.

6

Conclusion

6.1 Thesis Summary

In this thesis, I have developed a novel framework for human pose estimation that is innately integrated with tracking-by-detection techniques via per-frame predictions using an aligned input window. The lower-body pose estimator presented in Chapter 3 and the upper-body pose estimator presented in Chapter 4 estimate the pose from the tracking results. Both estimators utilize a few HOG features that are selected by random forests within the center-aligned whole-body or half-body region of the player. This globally-aligned appearance approach enables us to estimate the locations of joints provided that I have a corresponding pose in the training dataset. In addition, the proposed estimators can determine the pose even when parts are occluded; this is due to the deformation invariance of the HOG features and the sparse representation of the features trained by random forests. Moreover, the alignment strategy allows the training dataset to be resampled in order to evaluate the appearance at various scales. This enables the method to use only one model to deal with various or even unknown scales of the global appearance aligned to the head center or the pelvis center.

Since the rough tracker-based center alignment is due to the deformation and short-translation invariance of the HOG features, the trackers (head and whole body) can be replaced with any other trackers. In addition, a pose estimation can be made from even a single image when using the person detector and the head detector (or face detector). The selected HOG features results in pose estimators that are robust to translations of the player window and deformations within the cell. The selected HOG features for each pose estimator can be regarded as sport-specific estimators, since players in a specific sport have similar clothing (in this thesis, I primarily

trained with estimators derived from American football players). We can assume that players in any given sport will have similar clothing, and the proposed framework can be applied to any sport for which this is true.

Compared to the inspired poselets framework [30], the proposed label-grid classifier and poselets-regressors can be regarded as machine learning models that learn the continuous poselets models from aligned feature-calculation windows. The label-grid classifier and the poselets-regressor both use training images to learn the locations of the joints relative to the location of an aligned center joint; note that the training images are all aligned to the head center, but the target pelvis joint is unique in that its location can vary continuously).

I proposed two approaches for estimating the relative joint position using HOG features and random forests: the first uses label-grid classifiers as classification forests, and the second uses poselets-regressor as regression forests. As already discussed in Section 3.5, they can be viewed as a revisited version of a previous method [90], if I use machine learning and dense *selected* visual features. In other words, the traditional silhouette-matching strategy for pose estimation was renewed with the proposed approach that uses HOG features and random decision forests to embed all the pose patterns into the randomized feature space. This thesis also reveals that a very basic and important approach, that of using (globally) aligned images or features, is also important for estimating the location of human joints; this approach has been used in many face-recognition or body-orientation classification paradigms.

In the upper-body module, I also explored if the use of spine angle with similar values would help the learning of more-robust features. In each set of five spine angles, the upper-body HOG features had a similar appearance (similar contours) around the head and body region, but they had different and varied arm poses. With this alignment scheme, the random forests used for predicting body orientation were able to determine the edges around the head and body contours, and to use these to determine the body orientations that occurred in almost the same location within the aligned upper-body region. This novel feature-selection scheme for body orientation, which works even when the subject is leaning over, differs from previous methods, which are only able to deal with upright pedestrians.

The proposed method allows, for the first time, various types of pose information to be automatically obtained from team sports videos; this will open up a broad range of new research, primarily in the field of activity recognition. In the next section, I will discuss some possible future pose-based activity recognition studies for team sports, and I will also suggest ways to further develop this pose-estimation framework.

6.2 Future Work

In this last section, I will discuss two areas of future work:

- Estimating human poses (Section 6.2.1).
- Recognizing activities of sports players by using the estimated poses as input features (Section 6.2.2).

As I emphasized in Chapters 1 and 2, the proposed framework is targeted at developing pose-based activity recognition techniques, since this cannot be accomplished using the previous activity-recognition approaches based on visual features.

6.2.1 Future Work on Human Pose Estimation

Estimation of Left and Right Labels on Joints.

The proposed framework estimates the 2D joint locations of the players, and the label of left or right is based only on the location in the image. I would like to develop an algorithm that can use the label from the 2D joints to directly predict the 3D labels. This idea is related to the idea of foot-print estimation, which I will discuss in the next section.

Also the classification of the steps of leg movements will be explored. Especially in tennis or soccer, whether the legs are crossed or not and the degree of opening of the foot (i.e, open stance or closed stance) are the important cues because the play styles differ with the steps.

Integrate Joint Estimation Based on Global Appearance with the Classification of Local Parts.

As I have tried to point out several times in this thesis, one of the main contributions of the proposed framework is that it adopts the *global* (HOG) appearance aligned to a specific joint, and it then proceeds with continuous and relative training of each set of two adjacent joints by using a label-grid classifier or a poselets-regressor.

I intend to introduce into this framework some local part classification or regression techniques, such as that found in [75, 136], and to combine this with the joint location prediction likelihood obtained from the global appearance.

3D Pose Estimation Using Multiple Views.

Although this thesis is restricted to monocular videos, the framework can be easily extended to a multiview scenario. The reason for this is that the multiview approaches that use pictorial structures [75, 137] first estimate the 2D pose of the subject in each view, and they then integrate those results into one 3D pose by using pictorial structures.

3D Projection of 2D Pose.

Ramakrishna et al. [138] proposed using a 3D projection of the 2D joint landmarks to predict the 3D skeletal pose from images. Carr et al. [129] proposed projecting the 3D geometric primitives onto the image plane, and then using camera calibration to detect objects that have a shape similar to that of the primitives (e.g., cylinders for people, cubes for cars). Like the ideas for 3D projections from monocular images found in these papers [129, 138], the proposed framework also has the capability of projecting pose information from 2D images to 3D by using camera calibration or the known scale of the human body.

I already proposed an approximate 3D projection of the estimated spine pose of a player by using the calibrated ground floor information in [119]. For that method, I assumed that a local orthonormal projection around the tracked player would enable us to project the 2D spine pose into the 3D world; this would use the known length of the upper body in the image, and it would not require camera calibration. If a way can be found to accurately estimate the left/right labels, then I will be able to propose a 3D projection approach for the locations of the other lower-body joints.

6.2.2 Future Work on Recognizing the Activities of Sports Players

Footstep and Footprint Recognition by Discriminating between the Right and Left Legs.

The lower-body estimator predicts the left/right joint locations in an image, but with further research, I hope to turn those joint locations into information about the left/right legs in the real world. Once we determined the left/right leg labels for the joints, we can learn the classifiers for recognizing different types of footsteps and footprints. We have already published a proposal for estimating footprints [139]. However, it only uses the lower-body joints estimated with the label-grid classifiers and can be improved with more temporal models such as conditional random fields.

Player Attention Search.

Using the upper-body poses of multiple players, as estimated by the proposed framework, we can calculate attention maps, such as that proposed by Benfold and Reid [3], or we can calculate the visual focus of attention (VFOA) [140]. Motion fields [63], which are a combined map of the trajectories of the players, are similar to attention maps. Unlike motion fields, attention maps, which are based on the head/body direction, show the adversarial behavior of the players. For example, a player may glance at the opponents to determine their behavior, or defensive players make turn their bodies to face the opponents.

In the context of team sports, I intend to develop an attention search system that will be able to determine which players are looking at the same region, players, or ball. The intent is that this system will also be able to rank the degree of attention being directed at a specific region or during a specified time, by using degree-of-attention maps calculated from multiple players. As already explored, vision-based social-interaction or group-recognition information can be based on head or body orientations [6, 56]. I intend to use these ideas to categorize the attention maps from each pair of players in order to further our understanding of man-marking and combination plays.

References

[1] Chunyu Wang, Yizhou Wang, and Alan L Yuille. An approach to pose-based action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 915–922. IEEE, 2013. 1, 9, 12, 31

[2] Michalis Raptis and Leonid Sigal. Poselet key-framing: A model for human activity recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2650–2657. IEEE, 2013. 1, 9, 12, 27, 30

[3] Ben Benfold and Ian Reid. Guiding visual surveillance by tracking human attention. In *BMVC*, pages 1–11, 2009. 1, 2, 3, 18, 28, 29, 56, 58, 60, 70, 85, 101

[4] Isarun Chamveha, Yusuke Sugano, Daisuke Sugimura, Teera Siriteerakul, Takahiro Okabe, Yoichi Sato, and Akihiro Sugimoto. Appearance-based head pose estimation with scene-specific adaptation. In *in Proc. IEEE International Workshop on Visual Surveillance (VS2011), pp. 1713-1720, November 2011.*, 2011. 1, 11, 21, 28

[5] Isarun Chamveha, Yoichi Sato Yusuke Sugano, and Akihiro Sugimoto. Social group discovery from surveillance videos: A data-driven approach with attention-based cues. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2013. 1, 2, 9, 25, 82

[6] Tian Lan, Leonid Sigal, and Greg Mori. Social roles in hierarchical models for human activity recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1354–1361. IEEE, 2012. 1, 54, 101

[7] Vignesh Ramanathan, Bangpeng Yao, and Li Fei-Fei. Social role discovery in human events. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2475–2482. IEEE, 2013. 1, 25

[8] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013. 1, 5, 6, 7, 12, 17, 26, 27, 32, 53, 84

[9] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 1, 64

REFERENCES

[10] Pedro F Felzenszwalb and Daniel P Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005. 1, 5, 6, 21, 26

[11] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1385–1392. IEEE, 2011. 1, 2, 3, 5, 6, 9, 14, 15, 17, 21, 22, 26, 30, 31, 32, 33, 35, 41, 42, 44, 45, 46, 47, 52, 56, 57, 61, 63, 70, 71, 80, 84

[12] Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Poselet conditioned pictorial structures. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 588–595. IEEE, 2013. 1, 9, 23, 26, 56, 57

[13] Guodong Guo and Alice Lai. A survey on still image based human action recognition. *Pattern Recognition*, 47(10):3343–3361, 2014. 2

[14] Vittorio Ferrari, Manuel Marin-Jimenez, and Andrew Zisserman. Progressive search space reduction for human pose estimation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 2, 5, 21, 43

[15] Raquel Urtasun, David J Fleet, and Pascal Fua. 3d people tracking with gaussian process dynamical models. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 238–245. IEEE, 2006. 2, 5, 21, 23, 27, 32, 33

[16] Grégory Rogez, Jonathan Rihan, Srikumar Ramalingam, Carlos Orrite, and Philip HS Torr. Randomized trees for human pose detection. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 2, 5, 21

[17] Ben Benfold and Ian Reid. Unsupervised learning of a scene-specific coarse gaze estimator. In *International Conference on Computer Vision*, pages 2344–2351, 2011. 2, 11, 21, 25, 28, 29

[18] Cheng Chen and J Odobez. We are not contortionists: coupled adaptive learning for head and body orientation estimation in surveillance video. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1544–1551. IEEE, 2012. 2, 11, 21, 25, 28, 29, 30, 55, 57

[19] Alonso Patron-Perez, Marcin Marszalek, Ian Reid, and Andrew Zisserman. Structured learning of human interactions in tv shows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(12):2441–2453, 2012. 2, 25

[20] Jingchen Liu, Peter Carr, Robert T Collins, and Yanxi Liu. Tracking sports players with context-conditioned motion models. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1830–1837. IEEE, 2013. 2

[21] Wei-Lwun Lu, Kenji Okuma, and James J Little. Tracking and recognizing actions of multiple hockey players using the boosted particle filter. *Image and Vision Computing*, 27(1):189–205, 2009. 2

[22] Caroline Rougier, Jean Meunier, Alain St-Arnaud, and Jacqueline Rousseau. Monocular 3d head tracking to detect falls of elderly people. In *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, pages 6384–6387. IEEE, 2006. 2

[23] Ben Benfold and Ian Reid. Stable multi-target tracking in real-time surveillance video. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3457–3464. IEEE, 2011. 2

[24] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005. 2, 4, 7, 28, 33, 34, 36, 48, 56, 62, 70

[25] Antonio Criminisi and Jamie Shotton. *Decision forests for computer vision and medical image analysis*. Springer, 2013. 2, 3, 40, 42, 86

[26] Davide Baltieri, Roberto Vezzani, and Rita Cucchiara. People orientation recognition by mixtures of wrapped distributions on random trees. In *Computer Vision–ECCV 2012*, pages 270–283. Springer, 2012. 3, 11, 18, 21, 25, 28, 29, 55, 56, 57, 61, 62

[27] Junli Tao and Reinhard Klette. Integrated pedestrian and direction classification using a random decision forest. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 230–237. IEEE, 2013. 3, 18

[28] Peter N Belhumeur, João P Hespanha, and David J Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):711–720, 1997. 3

[29] Wenyi Zhao, Rama Chellappa, P Jonathon Phillips, and Azriel Rosenfeld. Face recognition: A literature survey. *ACM computing surveys (CSUR)*, 35(4):399–458, 2003. 3

[30] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1365–1372. IEEE, 2009. 3, 18, 25, 26, 30, 53, 56, 57, 61, 62, 98

[31] Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik. Detecting people using mutually consistent poselet activations. In *European Conference on Computer Vision (ECCV)*, 2010. 3, 26, 56, 57, 61

[32] Piotr Doll'ar, Peter Welinder, and Pietro Perona. Cascaded pose regression. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1078–1085. IEEE, 2010. 4, 27

[33] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. *International Journal of Computer Vision*, 107(2):177–190, 2014. 4

[34] Leonid Pishchulin, Arjun Jain, Mykhaylo Andriluka, T Thormahlen, and Bernt Schiele. Articulated people detection and pose estimation: Reshaping the future. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3178–3185. IEEE, 2012. 5

[35] Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic. 3d pictorial structures for multiple human pose estimation. *CVPR, IEEE*, 2014. 5

[36] Magnus Burenius, Josephine Sullivan, and Stefan Carlsson. 3d pictorial structures for multiple view articulated pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3618–3625. IEEE, 2013. 5

[37] Dariu M Gavrila and Larry S Davis. 3-d model-based tracking of humans in action: a multi-view approach. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*, pages 73–80. IEEE, 1996. 6

[38] Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: reconstruction and parameterization from range scans. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 587–594. ACM, 2003. 7

[39] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 408–416. ACM, 2005. 7

[40] Nils Hasler, Carsten Stoll, Martin Sunkel, Bodo Rosenhahn, and H-P Seidel. A statistical model of human pose and body shape. In *Computer Graphics Forum*, volume 28, pages 337–346. Wiley Online Library, 2009. 7

[41] Alexander Weiss, David Hirshberg, and Michael J Black. Home 3d body scans from noisy image and range data. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1951–1958. IEEE, 2011. 7

[42] Juergen Gall, Carsten Stoll, Edilson De Aguiar, Christian Theobalt, Bodo Rosenhahn, and H-P Seidel. Motion capture using joint skeleton tracking and surface estimation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1746–1753. IEEE, 2009. 7, 20

[43] Nils Hasler, Bodo Rosenhahn, T Thormahlen, Michael Wand, Jürgen Gall, and H-P Seidel. Markerless motion capture with unsynchronized moving cameras. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 224–231. IEEE, 2009. 7, 20

[44] Juergen Gall, Bodo Rosenhahn, Thomas Brox, and Hans-Peter Seidel. Optimization and filtering for human motion capture. *International journal of computer vision*, 87(1-2):75–92, 2010. 7, 20

[45] Jan Bandouch, Florian Engstler, and Michael Beetz. Evaluation of hierarchical sampling strategies in 3d human pose estimation. In *BMVC*, pages 1–10, 2008. 7, 20

[46] Jan Bandouch, Odest Chadwicke Jenkins, and Michael Beetz. A self-training approach for visual tracking and recognition of complex human activity patterns. *International journal of computer vision*, 99(2):166–189, 2012. 7, 20

[47] Yebin Liu, Carsten Stoll, Juergen Gall, H-P Seidel, and Christian Theobalt. Markerless motion capture of interacting characters using multi-view image segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1249–1256. IEEE, 2011. 7, 20

[48] Thomas B Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, 104(2):90–126, 2006. 7, 17, 20

[49] Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1627–1645, 2010. 7, 22, 23, 26, 33

[50] Tracab player tracking system. http://chyronhego.com/sports-data/tracab. 8

[51] Sportvu tracking system. http://www.stats.com/. 8

[52] L. Bazzani. *Beyond Multi-target tracking: statistical pattern analysis of people and groups*. PhD thesis, University of Verona, 2012. 8

[53] Eran Swears and Anthony Hoogs. Learning and recognizing complex multi-agent activities with applications to american football plays. In *Applications of Computer Vision (WACV), 2012 IEEE Workshop on*, pages 409–416. IEEE, 2012. 8

[54] Jagannadan Varadarajan, Indriyati Atmosukarto, Shaunak Ahuja, Bernard Ghanem12, Narendra Ahuja13, and IL Urbana-Champaign. A topic model approach to represent and classify american football plays. 2013. 8

[55] Patrick Lucey, Alina Bialkowski, Mathew Monfort, Peter Carr, and Iain Matthews. quality vs quantity: Improved shot prediction in soccer using strategic features from spatiotemporal data. MIT Sloan Sports Analytics Conference, 2014. 8

[56] Tian Lan, Yang Wang, Weilong Yang, Stephen N Robinovitch, and Greg Mori. Discriminative latent models for recognizing contextual group activities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(8):1549–1562, 2012. 9, 25, 82, 101

[57] Limin Wang, Yu Qiao, and Xiaoou Tang. Video action detection with relational dynamic-poselets. In *Computer Vision–ECCV 2014*, pages 565–580. Springer, 2014. 9

[58] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena. Unstructured human activity detection from rgbd images. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 842–849. IEEE, 2012. 9

REFERENCES

[59] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013. 9

[60] Lulu Chen, Hong Wei, and James Ferryman. A survey of human motion analysis using depth imagery. *Pattern Recognition Letters*, 34(15):1995–2006, 2013. 9

[61] Yang Wang, Duan Tran, and Zicheng Liao. Learning hierarchical poselets for human parsing. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1705–1712. IEEE, 2011. 9, 14, 23, 25, 26, 30, 56

[62] Marco Cristani, R Raghavendra, Alessio Del Bue, and Vittorio Murino. Human behavior analysis in video surveillance: A social signal processing perspective. *Neurocomputing*, 100:86–97, 2013. 9

[63] Kihwan Kim, Matthias Grundmann, Ariel Shamir, Iain Matthews, Jessica Hodgins, and Irfan Essa. Motion fields to predict play evolution in dynamic sport scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 840–847. IEEE, 2010. 11, 54, 101

[64] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *Computer Vision–ECCV 2014*, pages 540–555. Springer, 2014. 12

[65] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *Computer Vision–ECCV 2014*, pages 505–520. Springer, 2014. 12

[66] Zheng Lu and Kristen Grauman. Story-driven summarization for egocentric video. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2714–2721. IEEE, 2013. 12

[67] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Temporal localization of actions with actoms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2782–2795, 2013. 12

[68] J. C. Niebles, C. W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. European Conference on Computer Vision (ECCV), 2010. 12

[69] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1995–2002. IEEE, 2011. 12

[70] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, volume 1, pages 3–2, 2012. 12

[71] Limin Wang, Yu Qiao, and Xiaoou Tang. Video action detection with relational dynamic-poselets. In *ECCV*, pages 565–280, 2014. 12

[72] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. pages 65–72. Visual Surveillance and Performance Evaluation of Tracking and Surveillance (PETS), 2005. 12

[73] Alexei A Efros, Alexander C Berg, Greg Mori, and Jitendra Malik. Recognizing action at a distance. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 726–733. IEEE, 2003. 12

[74] H. Wang, A. Klaser, and C. Schmid. Dense trajectories and motion boundary descriptors for action recognition, 2013. 12

[75] Vahid Kazemi, Magnus Burenius, Hossein Azizpour, and Josephine Sullivan. Multi-view body part recognition with random forests. In *BMVC*, 2013. 15, 16, 23, 26, 32, 84, 99, 100

[76] Vasileios Belagiannis, Xinchao Wang, Bernt Schiele, Pascal Fua, Slobodan Ilic, and Nassir Navab. Multiple human pose estimation with temporally consistent 3d pictorial structures. In *European Conference on Computer Vision, ChaLearn Looking at People Workshop*, number EPFL-CONF-200374, 2014. 16

[77] Thomas B Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer vision and image understanding*, 81(3):231–268, 2001. 17, 20

[78] Vicon. http://www.vicon.com/. 19

[79] Optitrack. http://www.optitrack.com/. 19

[80] Xsense. https://www.xsens.com/. 19

[81] Synertial. http://www.synertial.com/. 19

[82] Organic motion. http://www.organicmotion.com/. 19

[83] Jonathan Deutscher, Andrew Blake, and Ian Reid. Articulated body motion capture by annealed particle filtering. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 126–133. IEEE, 2000. 20

[84] Ling Shao, Jungong Han, Pushmeet Kohli, and Zhengyou Zhang. *Computer Vision and Machine Learning with RGB-D Sensors*. Springer, 2014. 20

[85] Genzhi Ye, Yebin Liu, Nils Hasler, Xiangyang Ji, Qionghai Dai, and Christian Theobalt. Performance Capture of Interacting Characters with Handheld Kinects. In *Proc. ECCV*, pages 828–841. Springer, 2012. 20

[86] Openni. http://structure.io/openni. 20

[87] Kinect for windows sdk. https://dev.windows.com/en-us/kinect/tools. 20

[88] Stephan Gammeter, Andreas Ess, Tobias Jäggli, Konrad Schindler, Bastian Leibe, and Luc Van Gool. Articulated multi-body tracking under egomotion. In *Computer Vision–ECCV 2008*, pages 816–830. Springer, 2008. 21, 23

[89] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *International Conference on Computer Vision (ICCV)*, 2009. 22, 23, 24

[90] Marcel Germann, Tiberiu Popa, Remo Ziegler, Richard Keiser, and Markus Gross. Space-time body pose estimation in uncontrolled environments. In *Proceedings of the 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, 3DIMPVT '11, pages 244–251, Washington, DC, USA, 2011. IEEE Computer Society. 22, 48, 51, 98

[91] Ibrahim Radwan, Abhinav Dhall, Jyoti Joshi, and Roland Goecke. Regression based pose estimation with automatic occlusion detection and rectification. In *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, pages 121–127. IEEE, 2012. 23

[92] Grégory Rogez, Jonathan Rihan, Carlos Orrite-Uruñuela, and Philip HS Torr. Fast human pose detection using randomized hierarchical cascades of rejectors. *International journal of computer vision*, 99(1):25–52, 2012. 23

[93] Antonio Criminisi, Jamie Shotton, and Ender Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends® in Computer Graphics and Vision*, 7(2–3):81–227, 2012. 23, 39, 56, 57, 61, 62

[94] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011. 24

[95] Andreas Schulz, Naser Damer, Mika Fischer, and Rainer Stiefelhagen. Combined head localization and head pose estimation for video–based advanced driver assistance systems. In *Pattern Recognition*, pages 51–60. Springer, 2011. 25, 28, 29

[96] Andreas Schulz and Rainer Stiefelhagen. Video-based pedestrian head pose estimation for risk assessment. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 1771–1776. IEEE, 2012. 25, 28, 29

[97] Wongun Choi and Silvio Savarese. A unified framework for multi-target tracking and collective activity recognition. In *Computer Vision–ECCV 2012*, pages 215–230. Springer, 2012. 25, 82

[98] Marcin Eichner, Manuel Marin-Jimenez, Andrew Zisserman, and Vittorio Ferrari. 2d articulated human pose estimation and retrieval in (almost) unconstrained still images. *International journal of computer vision*, 99(2):190–214, 2012. 25

[99] Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Describing people: Poselet-based attribute classification. In *International Conference on Computer Vision (ICCV)*, 2011. 25, 27

[100] Bangpeng Yao and Li Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 17–24. IEEE, 2010. 25

[101] Bangpeng Yao, Jiayuan Ma, and Li Fei-Fei. Discovering object functionality. In *Submitted to the IEEE International Conference on Computer Vision (ICCV)*, 2013. 25

[102] Masaki Hayashi, Kyoko Oshima, Masamoto Tanabiki, and Yoshimitsu Aoki. Lower body pose estimation in team sports videos using label-grid classifier integrated with tracking-by-detection. *IPSJ Transactions on Computer Vision and Applications*, 7(1):18–30, 2015. 25, 31, 57, 61, 81

[103] Subhransu Maji, Lubomir Bourdev, and Jitendra Malik. Action recognition from a distributed representation of pose and appearance. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3177–3184. IEEE, 2011. 25, 27, 56

[104] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1014–1021. IEEE, 2009. 26, 48

[105] Min Sun, Pushmeet Kohli, and Jamie Shotton. Conditional regression forests for human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3394–3401. IEEE, 2012. 26, 27, 56, 57

[106] Carl Henrik Ek, Philip HS Torr, and Neil D Lawrence. Gaussian process latent variable models for human pose estimation. In *Machine learning for multimodal interaction*, pages 132–143. Springer, 2008. 27

[107] Tobias Jaeggli, Esther Koller-Meier, and Luc Van Gool. Learning generative models for multi-activity body pose estimation. *International Journal of Computer Vision*, 83(2):121–134, 2009. 27

[108] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 27

[109] Ju Han and Bir Bhanu. Individual recognition using gait energy image. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(2):316–322, 2006. 27

[110] Matthias Dantone, Juergen Gall, Gabriele Fanelli, and Luc Van Gool. Real-time facial feature detection using conditional regression forests. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2578–2585. IEEE, 2012. 27, 56, 57, 66

[111] Yusuke Sugano, Yuki Matsushita, and Yuuki Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1821–1828. IEEE, 2014. 27

[112] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems*, pages 1799–1807, 2014. 27

[113] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1653–1660. IEEE, 2014. 27

[114] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, volume 2, page 5, 2010. 28

[115] Brian Sapp and Ben Taskar. Modec: Multimodal decomposable models for human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3674–3681. IEEE, 2013. 28

[116] Ben Benfold and Ian Reid. Colour invariant head pose classification in low resolution video. In *BMVC*, pages 1–10, 2008. 28

[117] Cheng Chen, Alexandre Heili, and Jean-Marc Odobez. Combined estimation of location and body pose in surveillance video. In *Advanced Video and Signal Based Surveillance*, 2011. 28, 29, 55

[118] Cheng Chen, Alexandre Heili, and Jean-Marc Odobez. A joint estimation of head and body orientation cues in surveillance video. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 860–867. IEEE, 2011. 28, 29, 30

[119] Masaki Hayashi, Taiki Yamamoto, Kyoko Ohshima, Masamoto Tanabiki, and Yoshimitsu Aoki. Head and upper body pose estimation in team sport videos. In *International Joint Workshop on Advanced Sensing/Visual Attention and Interaction (ASVAI2013), on 2nd IAPR Asian Conference on Pattern Recognition (ACPR) 2013*, pages 754–759. IEEE, 2013. 28, 29, 35, 53, 56, 57, 59, 72, 73, 74, 75, 76, 77, 78, 81, 100

[120] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2411–2418. IEEE, 2013. 31

[121] Yi Yang, Simon Baker, Anitha Kannan, and Deva Ramanan. Recognizing proxemics in personal photos. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3522–3529. IEEE, 2012. 31

[122] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, pages 275–1. British Machine Vision Association, 2008. 32

[123] Sreemananth Sadanand and Jason J Corso. Action bank: A high-level representation of activity in video. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1234–1241. IEEE, 2012. 32

[124] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J. Black. Towards understanding action recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3192–3199, Sydney, Australia, December 2013. IEEE. 32

[125] Michael D Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1515–1522. IEEE, 2009. 32, 35

[126] Anil K Jain and Stan Z Li. *Handbook of face recognition*. Springer, 2005. 33

[127] C++ implementation of articulated pose estimation with flexible mixtures of parts. `https://github.com/wg-perception/PartsBasedDetector`. 42

[128] Masaki Hayashi, Kyoko Oshima, Masamoto Tanabiki, and Yoshimitsu Aoki. Upper body pose estimation for team sports videos using a poselet-regressor of spine pose and body orientation classifiers conditioned by the spine angle prior. *IPSJ Transactions on Computer Vision and Applications*, 7(1):121–137, 2015. 53

[129] Peter Carr, Yaser Sheikh, and Iain Matthews. Monocular object detection using 3d geometric primitives. In *Computer Vision–ECCV 2012*, pages 864–878. Springer, 2012. 53, 100

[130] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. Multicamera people tracking with a probabilistic occupancy map. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):267–282, 2008. 53

[131] Indriyati Atmosukarto, Bernard Ghanem, Shaunak Ahuja, Karthik Muthuswamy, and Narendra Ahuja. Automatic recognition of offensive team formation in american football plays. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 991–998. IEEE, 2013. 53

[132] Alina Bialkowski, Patrick Lucey, Peter Carr, Simon Denman, Iain Matthews, and Sridha Sridharan. Recognising team activities from noisy data. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 984–990. IEEE, 2013. 53

[133] Zhenhua Wang, Qinfeng Shi, Chunhua Shen, and Anton van den Hengel. Bilinear programming for human activity recognition with unknown mrf graphs. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1690–1697. IEEE, 2013. 54

[134] Wongun Choi, Khuram Shahid, and Silvio Savarese. What are they doing?: Collective activity classification using spatio-temporal relationship among people. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1282–1289. IEEE, 2009. 54

[135] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Monocular 3d pose estimation and tracking by detection. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 623–630. IEEE, 2010. 55, 56, 57

[136] Varun Ramakrishna, Daniel Munoz, Martial Hebert, Andrew J. Bagnell, and Yaser Sheikh. Pose machines: Articulated pose estimation via inference machines. In *ECCV*, 2014. 99

[137] Sikandar Amin, Mykhaylo Andriluka, Marcus Rohrbach, and Bernt Schiele. Multi-view pictorial structures for 3d human pose estimation. In *British Machine Vision Conference*, volume 2, 2013. 100

[138] Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Reconstructing 3d human pose from 2d image landmarks. In *Computer Vision–ECCV 2012*, pages 573–586. Springer, 2012. 100

[139] Tracking players and lower body pose estimation of the player in team sports videos with label-grid classifier. Springer, 2013. 100

[140] Xiaoming Liu, Nils Krahnstoever, Ting Yu, and Peter Tu. What are customers looking at? In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, pages 405–410. IEEE, 2007. 101