IPSJ Transactions on Computer Vision and Applications Vol.7 18–30 (Feb. 2015)

[DOI: 10.2197/ipsjtcva.7.18]

Research Paper

Lower Body Pose Estimation in Team Sports Videos Using Label-Grid Classifier Integrated with Tracking-by-Detection

Masaki Hayashi[1,a)] Kyoko Oshima[2,b)] Masamoto Tanabiki[2,c)] Yoshimitsu Aoki[1,d)]

Received: May 16, 2014, Accepted: November 6, 2014, Released: February 16, 2015

Abstract: We propose a human lower body pose estimation method for team sport videos, which is integrated with tracking-by-detection technique. The proposed Label-Grid classifier uses the grid histogram feature of the tracked window from the tracker and estimates the lower body joint position of a specific joint as the class label of the multi-class classifiers, whose classes correspond to the candidate joint positions on the grid. By learning various types of player poses and scales of Histogram-of-Oriented Gradients features within one team sport, our method can estimate poses even if the players are motion-blurred and low-resolution images without requiring a motion-model regression or part-based model, which are popular vision-based human pose estimation techniques. Moreover, our method can estimate poses with part-occlusions and non-upright side poses, which part-detector-based methods find it difficult to estimate with only one model. Experimental results show the advantage of our method for side running poses and non-walking poses. The results also show the robustness of our method for a large variety of poses and scales in team sports videos.

Keywords: human pose estimation, people tracking, tracking-by-detection, Random Forests, feature selection

1. Introduction

Video-based player tracking has drawn interest in computer vision. Since video-based object detection and tracking techniques have shown rapid improvement [1], 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 [2], [3] using a pose estimated by single image pose estimation techniques [4] 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 [5], [6] 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 Ref. [7] 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]. We 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 [9]. 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)[4]. 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 multi-view part-based pictorial structure techniques [10] with pan-tiled cameras cannot robustly estimate the side and part-occluded poses in team sports videos because they still depend on the discriminative part classification as Ref. [8] does. If the side poses of sports players could be estimated with monocular videos, a broad range of possibilities of

1 Keio University, Yokohama, Kanagawa 223–8522, Japan
2 R&D Division, Panasonic Corporation, Yokohama, Kanagawa 224–8539, Japan
a) mhayashi@aoki-medialab.org
b) ohshima.kyoko@jp.panasonic.com
c) tanabiki.masamoto@jp.panasonic.com
d) aoki@elec.keio.ac.jp

Fig. 1 Example result of our 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.

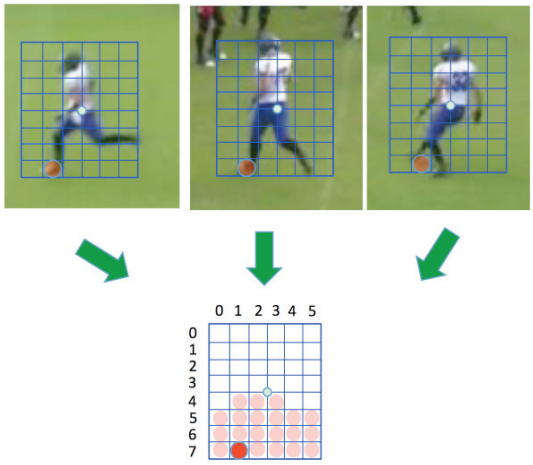


Fig. 2 Label-Grid Classifier. The red circle on the grid is the classified joint location $\boldsymbol{l}_t^j \in \mathbb{N}^2$ on each player window from the learned Label-Grid class candidates (pink circles). In this example, the Label-Grid classifier for the j-th joint is on the (6×8) grid structure, 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).

vision-based human motion and behavioral understanding would be opened up.

We propose a novel grid-wise pose estimation classifier for monocular team sports videos, which we call the *Label-Grid classifier*, integrated with a standard tracking-by-detection framework, such as Ref. [11]. Our 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 (**Fig. 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 **Fig. 2**).

To the best of our knowledge, our method is the first human pose estimation method that can estimate the pose of side-running players with scale changes, which part-based methods [4], [12] cannot estimate very well. As the example results in Fig. 1 show, our method can robustly estimate the poses of the side running sequence. Similar to (frontal) facial recognition techniques [13], our 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 our framework employs Histograms of Oriented Gradients (HOG) [14] 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 paper 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 paper we 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 Ref. [9], which uses a walking pose manifold or temporal pose prior.
- Per-frame pose estimation for videos *without* pictorial structure and part detectors. Our ignorance of pictorial structure framework achieved fast pose estimation (about 1 fps computational time).

The rest of the paper is organized as follows. In Section 2 we first review the related work for human pose estimation methods for images and videos. Our framework is presented in Section 3. Section 4 describes how to learn the Label-Grid classifier with a prepared dataset. Section 5 illustrates the experimental results and evaluates the performance of our framework. Finally, Section 6 is the conclusion.

2. Related Work

First, we review human pose estimation methods using the classical silhouette-based template matching technique for team sport videos (Section 2.1), which is not robust and is just for graphics visualization. Then we review two types of human pose estimation techniques: human pose estimation from a single image using pictorial structure (Section 2.2), and motion model regression (Section 2.3). Finally, we review instance template detection techniques such as poselets [15] and Exemplar-SVMs (Section 2.4).

2.1 Exemplar-based Pose Search Methods

Germann et al. [16] 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.

2.2 Single Image Pose Estimation Using Deformable Part Models

The FMP [4] is the extension of the deformable part models (DPM) [12] 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 [12] 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, FMP tends to estimate the pose of a person who looks 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 [15] part-based approaches have been studied [17], [18]. 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 [10] 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 [19]. 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 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 [9] is popular for learning a (latent) 3D pose manifold from cyclic pedestrian images from one camera view. For pedestrian pose estimation, Gammeter et al. [20] proposed a people tracking and pose estimation method for pedestrians using Gaussian Process Regression. Rogez et al. [21] proposed a per-frame pose estimation of human pose estimation based on Random Decision Forests [22] and pose manifolds of a gait sequence with HOG features. Reference [21] is close to our method in using Random Decision Forests for pose classification. However, their Random Decision Forests class is based on camera views and gait manifold cycle, while our 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.

2.4 Poselets and Exemplar-SVMs

Our pelvis-aligned detector and Label-Grid classifier are both inspired by the poselets framework [15]. 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-SVM [23] 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 learn each pose or view with one exemplar-SVM (e.g., left-view car SVM, frontal-view car SVM, jump-pose human SVM).

On the other hand, our 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.

3. Proposed Framework

The proposed framework (**Fig. 3**) is composed of two modules: a tracking player with tracking-by-detection technique with the pelvis-aligned detector (Section 3.1), and estimating the four joint positions on the grid structure independently using Label-Grid classifiers (Section 3.2).

These two modules share the tracked player window as HOG (Histogram-of-Oriented Gradients) feature [14] to estimate the pose (locations of four joints) in each frame of the video (Section 3.2). At test time, the only input of our 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.

We learn 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. Our 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 [4] [*1].

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

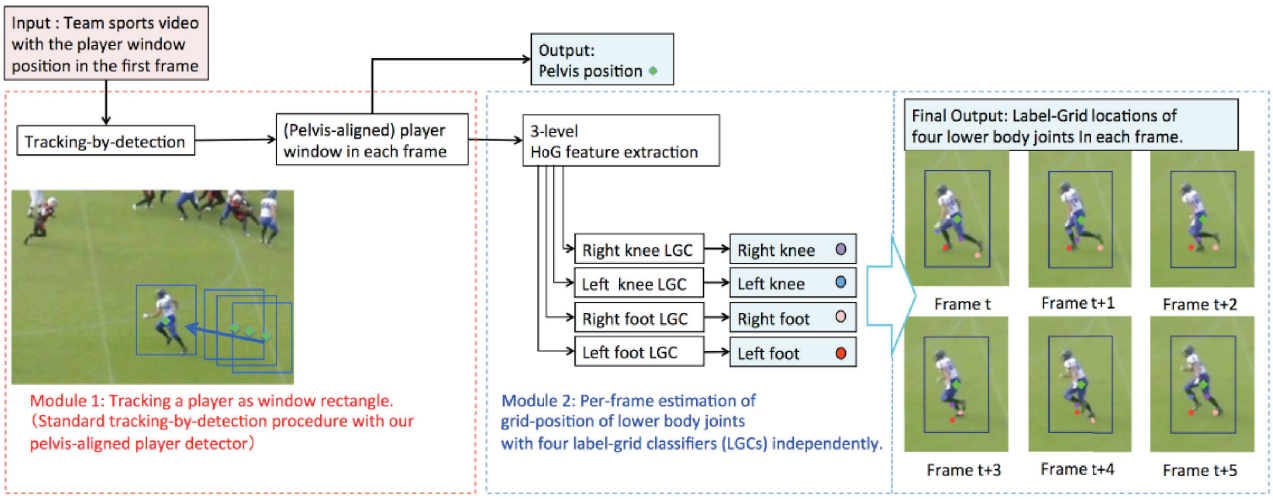


Fig. 3 Flowchart of the proposed framework.

3.1 Player Tracking with Pelvis-aligned Detector

Our first module is the player tracking method with standard tracking-by-detection, such as Ref. [11], to provide an aligned player window for the second module. We also used this pelvis-aligned detector in our upper body pose estimation framework [24], the explanation in Ref. [24] was very short because of the page limit. Hence, this paper provides a more detailed procedure to prepare the dataset of our pelvis-aligned player detector in Section 3.3.

For tracking-by-detection, we use the player detector learned from the dataset $\mathcal{D}_{all}$ (Section 3.3). We use a Kalman Filter (instead of Particle Filter in Ref. [11]) 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 our method mainly targets at estimating side poses during running or walking, which occurs very often in team sports videos (as mentioned in the introduction), we 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 we use HOG feature for classifying the pose, we 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.

3.2 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 [14]. 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 we denote the input visual feature vector (in our 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 Fig. 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 $\boldsymbol{x}_t$:

$$\hat{\boldsymbol{y}}_t^j = F^j(\boldsymbol{x}_t) \tag{1}$$

The reason why we use not only the lower body but also the full body window for the HOG feature is that we 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. We 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 [*2].

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

$$\boldsymbol{l}_t^j = M^j(\hat{\boldsymbol{y}}_t^j) \tag{2}$$

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 we call Label-Grid. This mapping dictionary M^j is built during training. We 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 Fig. 2 for the example grid index assignment). Note that we need the inverse mapping of $M^j(\hat{\boldsymbol{y}}_t^j)$ during training, because we 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, we need only $M^j(\hat{\boldsymbol{y}}_t^j)$ for converting $\hat{\boldsymbol{y}}_t^j$ to $\boldsymbol{l}_t^j$.

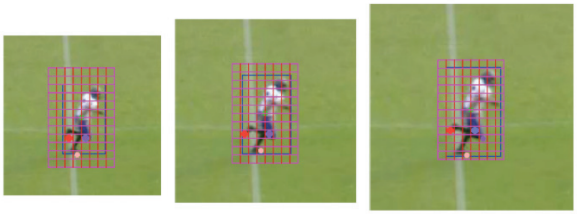
The training dataset for the Label-Grid classifier must include most of the types and scales of players' appearances that could occur in the target videos. Hence, our system can estimate the lower body pose in any location of the image (in our case, the

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

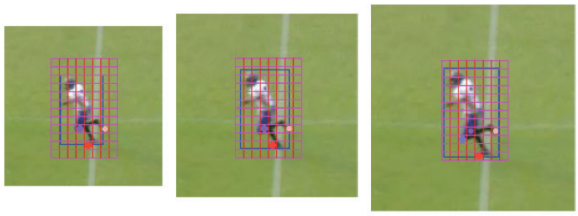
sports field) where player scale varies according to the position and the pose of the camera.

3.3 Dataset Preparation

We 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 $\mathcal{D}_{all}$, we perform data augmentation with different scales



(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 $\mathcal{D}_{sca}$.

Fig. 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.

and mirrored images from the original dataset $\mathcal{D}_{ori}$ (**Fig. 4**).

We 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. $\mathcal{D}_{sca}$ (Fig. 4 (a)) is the resampled player window images and their labels from the original dataset $\mathcal{D}_{ori}$ for which we should only need to prepare the labels. $\mathcal{D}_{mir}$ (Fig. 4 (b)) is the mirrored dataset of $\mathcal{D}_{sca}$ whose images and labels are flipped horizontally. See the left half of the **Fig. 5** for this dataset preparation procedure.

First, we 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, we 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, ..., 1.00) by resizing $\mathcal{D}_{ori}$ (Fig. 4 (a)).

Finally, we acquire $\mathcal{D}_{mir}$ by flipping the images and labels of $\mathcal{D}_{sca}$ horizontally to learn the mirrored features and labels (Fig. 4 (b)).

The player detector uses only the images of $\mathcal{D}_{all}$ because the

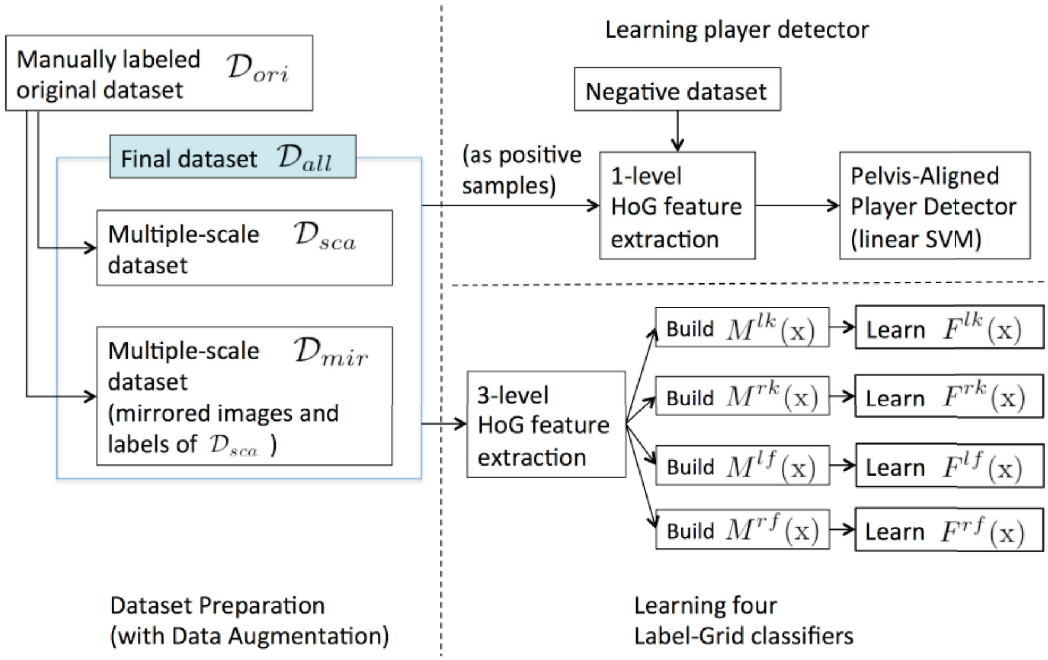


Fig. 5 Learning procedure.

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, we use Random Decision Forests [22] as a Label-Grid classifier in our experiments.

4. 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 our 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 (Fig. 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 Fig. 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 4.1). 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.

4.1 Learning Procedure

We will explain how to learn a Label-Grid Classifier for classifying the j-th joint (e.g., the left knee joint). Figure 5 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 $\mathcal{D}_{all}$, we 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 $\mathcal{D}_{all}$, we 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, we can finally learn $F^j(\boldsymbol{x}_i)$ (using Random Decision Forests, in this paper) with Label-Grid s and the calculated feature $\boldsymbol{x}_{all}$ that we will explain in the next subsection.

4.2 Multi-level HOG Feature and Feature Selection

We 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, we normalize the feature vector $\boldsymbol{x}_t$ to L_2 unit vector both at training time and test time.

We use L-class Random Decision Forests as the L-class Classification Forests [22] 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 $\boldsymbol{y}_t^j$.

5. Experiments

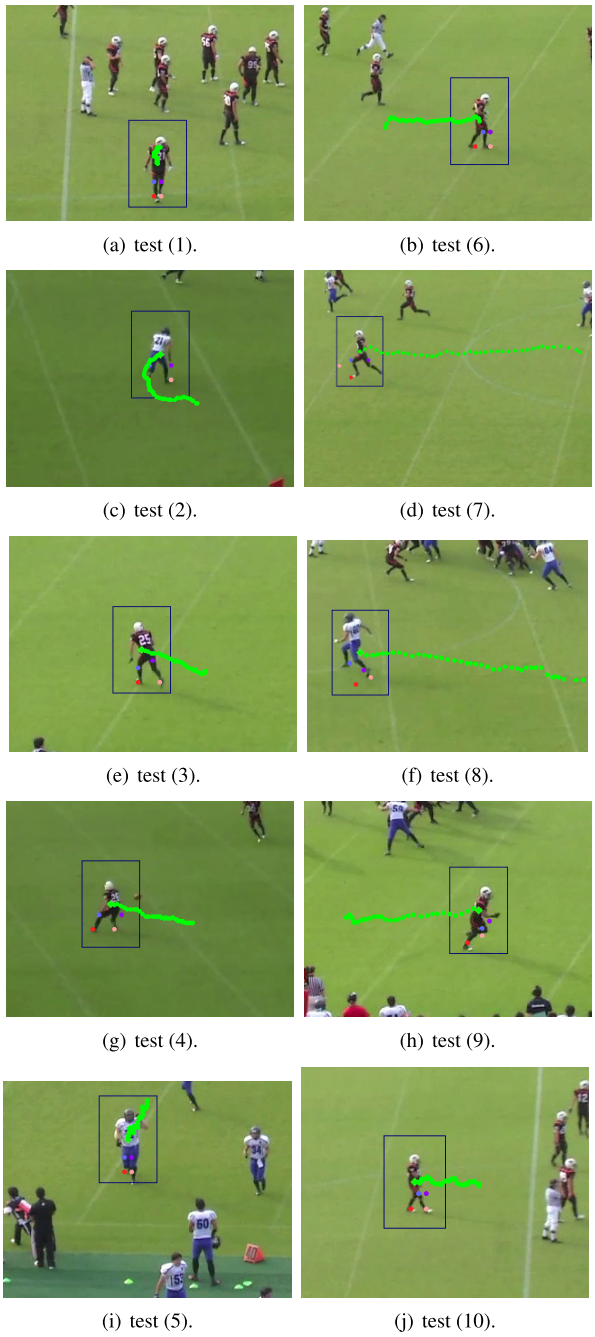
We tested our framework in two scenarios: frontal pose sequences (Section 5.3.1), which part-based pose estimator [4] can also estimate robustly, and side pose sequences (Section 5.3.2), which part-based pose estimator *cannnot* predict properly as argued in Section 2.2.

5.1 Experimental setup

We performed experimental evaluations on our system with American Football videos in professional league matches. The size of each video is $1{,}280 \times 780$. The videos are taken from the matches of Panasonic IMPULSE [*3]. 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 we captured high-resolution videos, motion blur of moving legs and arms sometimes occurs and the players are captured with a relatively low resolution. We created 10 test sequences (test (1)–(10)) for the five frontal pose tests and five side pose tests from these videos (see **Fig. 6** to see the player trajectories on each sequence). Each test video is composed of 40 frames. We will show the detail behavior (pose) on each sequence in Section 5.3.1 for the frontal pose sequences and Section 5.3.2 for side pose sequences.

We manually clipped player windows from video frames and assigned labels to create our original dataset $\mathcal{D}_{ori}$ for training both Label-Grid classifiers and the player detector. We 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 the American Football videos. We 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 we 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 25,402 images ($25{,}402 = 977 \times 13 \times 2$) for training four Label-Grid classifiers independently.

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

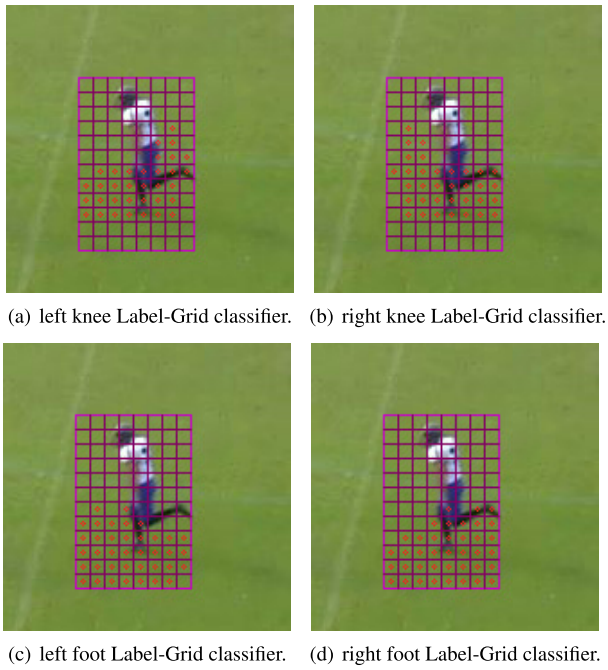


(a) test (1). (b) test (6).
(c) test (2). (d) test (7).
(e) test (3). (f) test (8).
(g) test (4). (h) test (9).
(i) test (5). (j) test (10).

Fig. 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.

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

We learned four Label-Grid classifiers as Random Decision Forests [22] with the feature vector $\boldsymbol{x}_t$ for each lower body joint



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

Fig. 7 Four Label-Grid classifiers with 8×12 Label-Grids, which we use in our experiments. Each red circle shows the candidate Label-Grid class of the classifier.

independently (right/left knee and right/left foot) from the training dataset $\mathcal{D}_{all}$. Consequently, we 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 **Fig. 7** for the class assignment).

To apply FMP [4] as a baseline, we used PartsBasedDetector software [25]. We 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 our 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). We assume that the center of the rectangle of each detected part is the corresponding joint position in the image.

5.2 Evaluation Manner

5.2.1 Pixel Error of the Joint Position

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

$$E_t = d(\hat{\mathbf{p}}_t, \mathbf{p}_t^{\mathbf{GT}}) \tag{3}$$

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 our 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 our experimental videos because we apply our 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 we 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.

(a) test (1). (b) test (2). (c) test (3).

(d) test (4). (e) test (5).

Fig. 8 Example results from frontal pose experiments. The left tow panels in each subfigure (a)–(e) show the results of our 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).

Although we would like to compare our methods with FMP using the PCP [26] score, which is broadly applied to the evaluation of part-based methods, we cannot calculate the PCP score because our method does not infer the stick area of each part which is needed for calculating PCP scores. This is one of the reasons why we used the Euclidean distance for the evaluation.

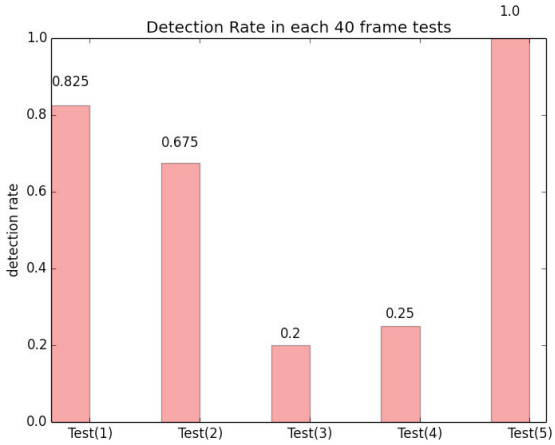
5.2.2 Detection Rate of FMP and How to Apply FMP to Our Videos

To test the limitations of FMP for side poses in our videos, we 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 our case, $N = 40$ frames).

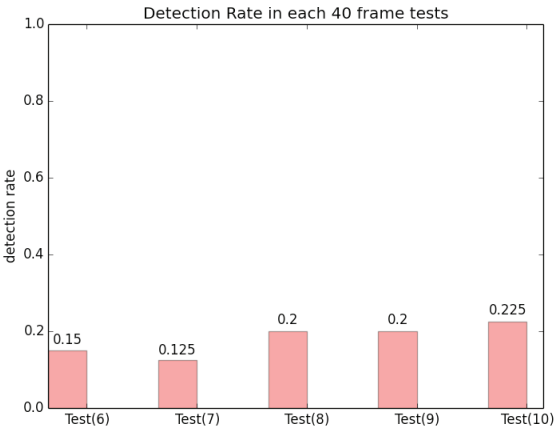
Since FMP [4] was the object detector (but jointly estimate the pose while detecting the object), we automatically clipped the magnified and margin-added image to apply the FMP detector. We first clipped the player window from the tracking-by-detection tracking module (Section 3.2) by adding 40×40 margin, and magnified it 200% to enable FMP to detect the players in our video. Each of images in **Fig. 8** shows the clipped images with this procedure.

5.3 Experimental Results

Figure 9 (a) shows the detection rate of FMP [4] in tests (1)–(5). Since the movement of the players in tests (2)–(4) are diagonal and curved (Fig. 6), most of their poses were difficult for FMP with hard occlusion and low-resolution, even though we defined those tests as frontal tests. However, since our method does not use any part-models and just use tracked (whole) player window appearance in each frame, it can classify the joint position in all frames in test (1)–(5). For example, in the Fig. 8 (d), while FMP could not detect the player in each frame, our Label-Grid classifier estimated the joint positions correctly from the same images. While we wanted to compare the position error between our methods and FMP, we abandoned the calculation of the pixel



(a) Detection rate of FMP [4] in frontal pose tests (1)–(5).



(b) Detection rate of FMP [4] in side pose tests (6)–(10).

Fig. 9 Detection Rate of FMP [4] in each test.

Table 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

error for FMP since we were not able to get enough detections from even frontal poses (Fig. 9 (a)).

5.3.1 Frontal Pose Experiment

We tested our system on the following frontal pose scenarios to compare the performance or detection rate with FMP [4].

We prepared the following five sequences for a frontal pose dataset (Fig. 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.

Figure 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 1** shows the average error of our method and FMP [4] for each joint.

Table 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

Note that our Label-Grid is 8×8 pixels for all tests. While FMP sometimes failed to detect a player who had occluded parts, our method could detect non tree-structured poses. Figure 8 shows the example results of our method and FMP to compare with each other.

5.3.2 Side Pose Experiment

Just as for the frontal pose experiment in the previous section, we also performed evaluation of our method and FMP with the following five side pose scenarios (Fig. 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 runs straight from the starting position.
- Test (10): The player walks backward.

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

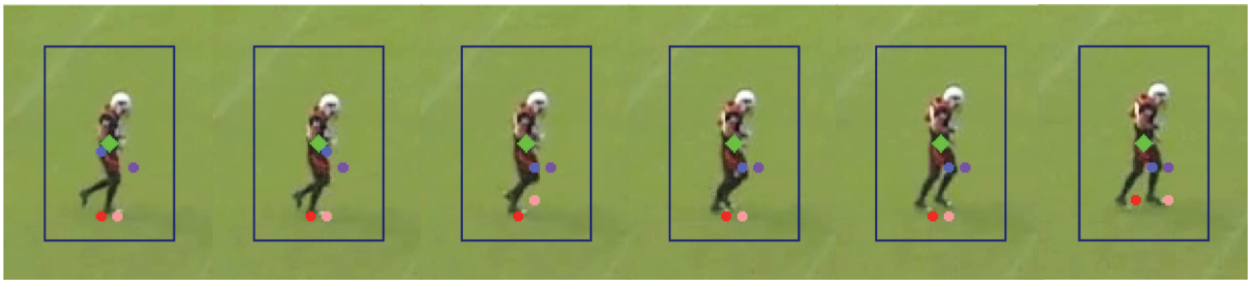
Table 2 shows the average error of our method in these side pose tests and **Fig. 10** shows the example results of tests (6)–(8). Figure 9 (b) shows the detection rate of FMP [4] in side pose tests (6)–(8). As Fig. 10 shows, FMP can rarely detect the player in side pose tests. FMP detects player with a detection rate 0.15 to 0.25. Compared with these results of FMP, our method could estimate the pose in all frames via its tracking and classification procedure within about two Label-Grid errors (Table 2).

5.4 Discussions by Topic

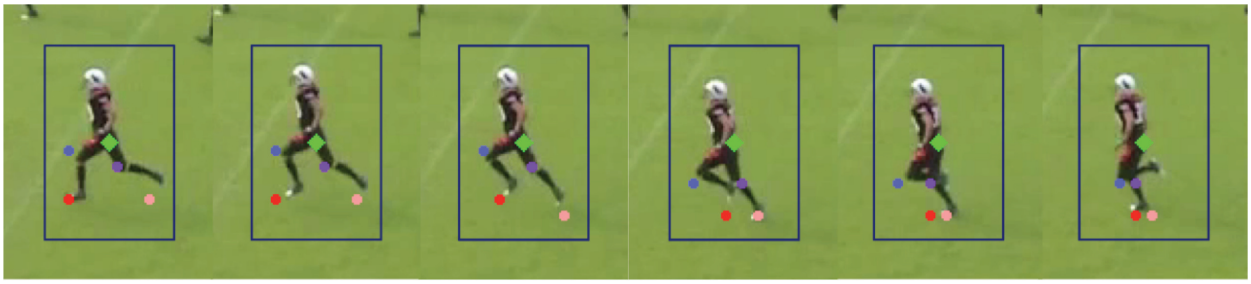
5.4.1 Whole Body Appearance Feature as Multi-level HOG

As already argued in Section 3.2, we use the whole body appearance to classify the lower body pose. Our HOG-based classification approach can be viewed as the modern replacement of the classical silhouette-matching schemes using background subtraction, such as Ref. [16]. We instead use randomized HOG features (learned by Random Decision Forests) to robustly classify the pose with machine learning. Our strategy has richer information with which to discriminate Label-Grid classes than using only the lower body appearance. We instead use the whole body HOG appearance to estimate the joint position.

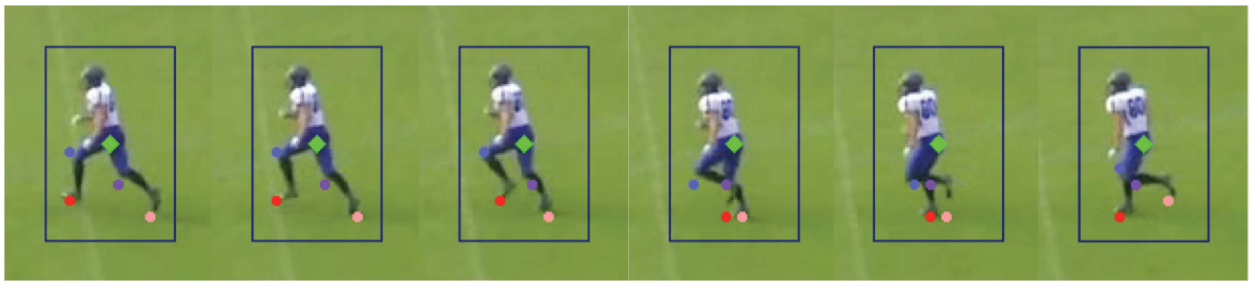
Moreover, owning to the deformation invariance of the HOG features, our Label-Grid classifier can estimate the pose of a larger or slimmer person until the gradient distribution changes from the feature distribution of the dataset. For instance, we performed an experiment using a large player in test (5) (see Fig. 8 (e)). Even though we only included middle-sized and thin players in the original dataset $\mathcal{D}_{ori}$, our classifiers could still esti-



(a) test (6).



(b) test (7).



(c) test (8).

Fig. 10 Example results from side pose experiments. Each row shows the results of side pose test (6), (7), and (8).

mate the joint location of the large-sized player.

5.4.2 Disregard of Pictorial Structure and Low-resolution Invariance of Our Method

Another important part of the nature of our method is that our framework disregards the pictorial structure strategy [27] while it depends on the aligned window appearance. As we showed in our experimental results for side pose tests (Fig. 10), our 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 we already argued in Section 4.2, our 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 Fig. 8 and Fig. 10).

While FMP and the other part-detector techniques assume clear and non-blurred images, our 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 [14].

5.4.3 Sport-specific Classifier

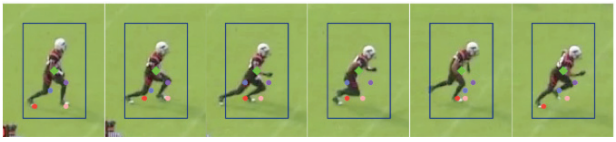
While our method is able to estimate the lower body pose with various types of poses in American Football robustly, our Label-Grid classifier is learned from the same clothing type while the pictorial structure includes various types of clothing. In our experiments, we learned Label-Grid classifiers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

5.4.4 Alignment and Scale of the Window is Important

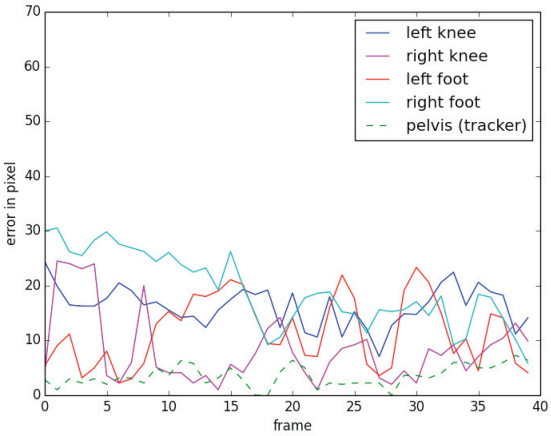
As already mentioned, our framework depends on the alignment of the player window between the tracking module and the classification module. In our experiments, our player detector is learned with a HOG of 8×8 cell size, which tracks the player within a cell error. This means that Label-Grid classifiers 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 11 shows the temporal analysis of the side pose test (9). By observing Fig. 11 (a), the left foot position gradually becomes unstable as the left leg is out of the window. This sequence shows the nature of our window alignment scheme. While you can use wider windows for the Label-Grid classifiers to prevent this case by restricting the person within the window, you need to make more patterns for the dataset because the number of classes increases with a wider Label-Grid widow.

Figure 11 (b) shows the error values of four lower body joints and the pelvis position in each frame of the test (9). Owing to the



(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).

Fig. 11 Temporal analysis of test (9).

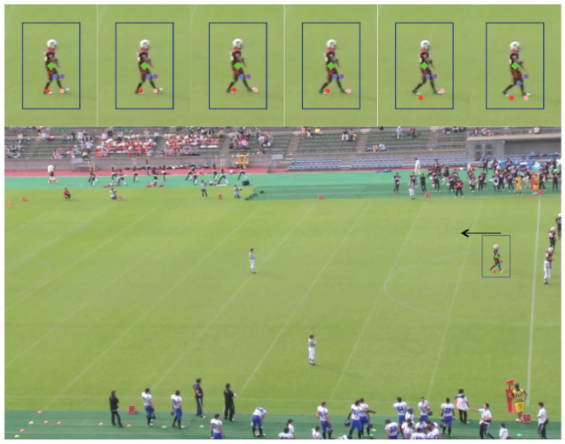


Fig. 12 Walking back result of test (10).

dependence of the alignment of the tracker, Label-Grid classifiers tend to have more errors along with an increase in pelvis position error. In Fig. 11 (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, we 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 our 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 (continuous) scales of features (in Random Decision Forests feature space) as possible, the failure happens if the player is an unknown scale (e.g., $s = 0.60$).

5.4.5 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 12** shows the result of test (10).

This shows the ability of our 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.

6. Conclusion

To estimate lower body poses from low-resolution images, we proposed a new human pose estimation method using a Label-Grid classifier that is integrated with an object tracker. Our Label-Grid classifier does *not* use the pictorial structures, *but* use the alignment of the player's pelvis position to classify various types and scales of poses into a grid structure with off-the-shelf multi-class classifiers (we use Random Decision Forests in this paper). 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. Our 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. Our framework can be viewed as a revisited version of Ref. [16] by using machine-learning and dense visual features. In other words, traditional silhouette matching strategy for pose estimation was innovated by our approach using HOG features and Random Decision Forests to embed all pose appearance patterns into the randomized feature space.

In this work, we only investigated the possibility of our 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.

As our experiments showed, our system can estimate all types of poses with only monocular RGB videos if a sufficient amount of poses are prepared in datasets. In addition, our method can estimate side-running poses while FMP [4] 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.

We 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.

Our future work includes joint estimation of multiple joints and probabilistic formulation using a kinematic body model (while this paper only investigates one-shot classification as a first simple proposal of grid-wise classification for pose estimation). Moreover, we will use 3D pose information such as the upper body pose, which we are exploring in other research [24], or movement direction to restrict the pose feature space using these types of information as priors. For behavior understanding using the lower body pose, we will investigate the leg-based activity recognition such as recognizing foot steps and key-pose extraction for sports video summarization and retrieval.

Acknowledgments The data were provided by the Panasonic Corporation and the Japan American Football Association. The authors also would like to thank Kiyoshi Matsuo for his help on the discussion of the experimental settings.

References

[1] Wu, Y., Lim, J. and Yang, M.-H.: Online object tracking: A benchmark, *2013 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp.2411–2418, IEEE (2013).
[2] Wang, C., Wang, Y. and Yuille, A.L.: An approach to pose-based action recognition, *2013 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp.915–922, IEEE (2013).
[3] Yang, Y., Baker, S., Kannan, A. and Ramanan, D.: Recognizing proxemics in personal photos, *2012 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp.3522–3529, IEEE (2012).
[4] Yang, Y. and Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts, *2011 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp.1385–1392, IEEE (2011).
[5] Klaser, A. and Marszalek, M.: A spatio-temporal descriptor based on 3d-gradients (2008).
[6] Sadanand, S. and Corso, J.J.: Action bank: A high-level representation of activity in video, *2012 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp.1234–1241, IEEE (2012).
[7] Jhuang, H., Gall, J., Zuffi, S., Schmid, C. and Black, M.J.: Towards understanding action recognition, *IEEE International Conference on Computer Vision* (*ICCV*), Sydney, Australia, pp.3192–3199, IEEE (2013).
[8] Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M. and Moore, R.: Real-time human pose recognition in parts from single depth images, *Comm. ACM*, Vol.56, No.1, pp.116–124 (2013).
[9] Urtasun, R., Fleet, D.J. and Fua, P.: 3D people tracking with Gaussian process dynamical models, *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol.1, pp.238–245, IEEE (2006).
[10] Kazemi, V.: Multi-view Body Part Recognition with Random Forests, *2013 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp.588–595, IEEE (2013).
[11] Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E. and Van Gool, L.: Robust tracking-by-detection using a detector confidence particle filter, *2009 IEEE 12th International Conference on Computer Vision*, pp.1515–1522, IEEE (2009).
[12] Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A. and Ramanan, D.: Object Detection with Discriminatively Trained Part-Based Models, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.32, pp.1627–1645 (online), DOI: 10.1109/TPAMI.2009.167 (2010).
[13] Jain, A.K. and Li, S.Z.: *Handbook of face recognition*, Springer (2005).
[14] Dalal, N. and Triggs, B.: Histograms of Oriented Gradients for Human Detection, *Computer Vision and Pattern Recognition*, Vol.1, pp.886–893 (2005).
[15] Bourdev, L. and Malik, J.: Poselets: Body Part Detectors Trained Using 3D Human Pose Annotations, *International Conference on Computer Vision* (*ICCV*) (2009).
[16] Germann, M., Popa, T., Ziegler, R., Keiser, R. and Gross, M.: Space-Time Body Pose Estimation in Uncontrolled Environments, *Proc. 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission* (*3DIMPVT '11*), pp.244–251, Washington, DC, USA, IEEE Computer Society (2011).
[17] Wang, Y., Tran, D. and Liao, Z.: Learning hierarchical poselets for human parsing, *2011 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp.1705–1712, IEEE (2011).
[18] Pishchulin, L., Andriluka, M., Gehler, P. and Schiele, B.: Poselet conditioned pictorial structures, *2013 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp.588–595, IEEE (2013).
[19] Radwan, I., Dhall, A., Joshi, J. and Goecke, R.: Regression based pose estimation with automatic occlusion detection and rectification, *2012 IEEE International Conference on Multimedia and Expo* (*ICME*), pp.121–127, IEEE (2012).
[20] Gammeter, S., Ess, A., Jäggli, T., Schindler, K., Leibe, B. and Van Gool, L.: Articulated multi-body tracking under egomotion, *Computer Vision–ECCV 2008*, pp.816–830, Springer (2008).
[21] Rogez, G., Rihan, J., Orrite-Uruñuela, C. and Torr, P.H.: Fast human pose detection using randomized hierarchical cascades of rejectors, *International Journal of Computer Vision*, Vol.99, No.1, pp.25–52 (2012).
[22] Criminisi, A. and Shotton, J.: *Decision forests for computer vision and medical image analysis*, Springer (2013).
[23] Malisiewicz, T., Gupta, A. and Efros, A.A.: Ensemble of Exemplar-SVMs for Object Detection and Beyond, *International Conference on Computer Vision* (*ICCV*) (2011).
[24] Hayashi, M., Yamamoto, T., Ohshima, K., Tanabiki, M. and Aoki, Y.: Head and Upper Body Pose Estimation in Team Sport Videos, *2013 2nd IAPR Asian Conference on Pattern Recognition* (*ACPR*), pp.754–759, IEEE (2013).
[25] Bristow, H.: C++ implementation of Deva Ramanan's "Articulated Pose Estimation with Flexible Mixtures of Parts".
[26] Ferrari, V., Marin-Jimenez, M. and Zisserman, A.: Progressive search space reduction for human pose estimation, *2008 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp.1–8, IEEE (2008).
[27] Andriluka, M., Roth, S. and Schiele, B.: Pictorial structures revisited: People detection and articulated pose estimation, *2009 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp.1014–1021, IEEE (2009).

Masaki Hayashi received an M.Sc. degree in Computer Vision and Image Processing from Keio University in 2006. Since 2012, he has been a Ph.D. candidate at Keio University. His research interests are video-based human pose estimation and activity recognition, especially for team sports videos.

Kyoko Oshima received a B.A. of Liberal Arts from International Christian University in 1992. She is a staff engineer in the Panasonic Corporation and works on research and development of computer vision system.

Masamoto Tanabiki received an M.Sc. degree in Electrical Engineering from Waseda University in 1998. His research interests are video-based human tracking and pose estimation, especially for security, business intelligence, and sports.

Yoshimitsu Aoki is an Associate Professor, Department of Electronics and Electrical Engineering, Keio University. He received his Ph.D. in Engineering from Waseda University in 2001. From 2002 to 2008, he was an Associate Professor in Shibaura Institute of Technology. Since 2008, he has been an Associate Professor at Department of Electronics and Electrical Engineering in Keio University. He performs research in the areas of Computer Vision, Pattern Recognition, and Media Sensing/Understanding.

(Communicated by *Greg Mori*)