Recommended Paper

Sports Field Recognition Using Deep Multi-task Learning

Shuhei Tarashima[1,a)]

Received: July 16, 2020, Accepted: January 12, 2021

Abstract: In this paper we propose a novel approach to build a single shot regressor, called *SFLNet*, that directly predicts a parameter set relating a sports field seen in an input frame to its metric model. This problem is challenging due to the huge intra-class variance of sports fields and the large number of free parameters to be predicted. To address these issues, we propose to train our regressor in combination with semantic segmentation in a multi-task learning framework. We also introduce an additional module to exploit the spacial consistency of sports fields, which boosts both regression and segmentation performances. SFLNet can be trained with a dataset that can be semi-automatically built from human annotated point-to-point correspondences. To our knowledge, this work is the first attempt to solve this sports field localization problem relying only on an end-to-end deep learning framework. Experiments on our new dataset based on basketball games validate our approach over baseline methods.

Keywords: homography, semantic segmentation, multi-task learning, sports analytics

1. Introduction

Sports analytics have been extensively used to build competitive teams, improve scouting, predict match outcomes, and enhance the fan experience [1], [2]. Among the techniques in sports analytics, computer vision plays a key role both in the automatic performance assessment of individual players and in the improvement of team formations and strategies. The majority of commercial systems such as STATS [*1] and TRACAB [*2] collect visual data using static cameras with fixed intrinsic parameters, making analysis simple but requiring costly installation. One way to reduce the cost is to leverage alternative resources such as broadcast videos or consumer-generated media. However, it is challenging to analyze such data because camera parameters may be varied over time. To extract valuable statistics from these resources, we need to estimate frame-by-frame correspondence between the sports field seen by the camera and the metric model of the field.

In this work we tackle the automatic sports field localization problem, on which algorithms estimate a set of parameters that corresponds the sports field in a given frame to its metric model without any manual intervention. Specifically, we here aim at developing a single shot regressor that can directly predict the parameter set from an input frame (cf., **Fig. 1**). Existing algorithms [3], [4], [5] tailored to the same problem consist of several steps and have a tradeoff between accuracy and efficiency. Single shot regression has already been employed to solve related tasks (e.g., camera pose estimation [6], [7], [8], [9], [10], [11]), but these approaches are difficult to apply directly due to the different problem settings. To this end, we propose a novel approach to build a regressor based on a convolutional neural network (CNN), called *SFLNet*, that can directly predict the correspondence parameter. To our knowledge, this is the first attempt in the literature to solve the sports field localization problem relying only on an end-to-end deep learning framework. The contributions of this work can be summarized as follows:

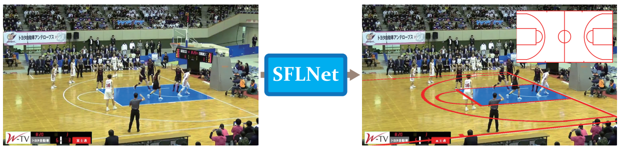


Fig. 1 Our single shot regressor, SFLNet, directly regresses a set of parameters that corresponds the metric model of a sports field (shown in top right on the right image) to the court seen in an input frame. Best viewed in color.

(1) We propose to build our parameter regressor in combination with a semantic segmentation module and train the whole model in an end-to-end multi-task learning framework. The semantic segmentation module is responsible for layout estimation of the input frame, and its intermediate feature map is used to regress correspondence parameters.

(2) We introduce an additional module to exploit contextual information focusing on the properties of sports fields. This module can exploit the spatial consistency of sports fields with a very low extra computational cost, and can efficiently boost both semantic segmentation and parameter regression performances. We will validate this module in our ablation studies.

(3) We compile a novel dataset to evaluate sports field localization methods. This dataset is built on a number of basketball games held in different stadiums with various camera installations and moves. We use this dataset to demonstrate the

[1] NTT Communications Corporation, Minato, Tokyo 105–0023, Japan
[a)] tarashima@acm.org

[*1] https://www.stats.com/

[*2] https://chyronhego.com/products/sports-tracking/tracab-optical-tracking/

The preliminary version of this paper was published at Forum on Information Technology 2019 (FIT 2019), held in September 2019. The paper was recommended to be submitted to Journal of Information Processing (JIP) by Program Committee Chair of FIT 2019.

superiority of our approach over several baseline methods.

2. Related Works

Assuming the sports field is planar, the transformation between its metric model and the field seen in an input frame can be defined by a homography matrix $H \in \mathbb{R}^{3\times3}$, which has 8 degrees of freedom (DoF). One of the simplest ways to estimate this homography is to first detect field markings (e.g., points, lines, intersections) in the frame and then associate them with corresponding markings in the model. Given these correspondences, the homography can be easily estimated by the closed form Direct Linear Transform (DLT) algorithm [13]. Unfortunately, this approach is difficult to perform fully automatically: Field marking detection remains a non-trivial task because markings are usually small, textureless, and sometimes cannot be seen in the frame. Therefore, most existing sports field localization methods assume manual intervention [10], [14], [15], [16], [17], [18], [19], [20], [21], [22], which make them less applicable within a real-time setting.

To our knowledge, relatively fewer works [3], [4], [5] focus on fully automatic approaches. For instance, Homayounfar et al. [3] formulate automatic sports field localization problem as a branch and bound inference in a Markov random field where an energy function is defined in terms of semantic cues such as the field surface, lines, and circles obtained from a semantic segmentation result. On the other hand, Sharma et al. [4] formulate the problem as a nearest neighbor search in a precomputed dictionary with known homographies. Chen et al. [5] improve Sharma's approach by adopting case-specific assumptions (e.g., PTZ camera and its position) to extend the dictionary and employing a GAN framework for better feature extraction. All the above methods consist of several steps and the whole pipelines cannot be optimized end-to-end. More importantly, they all suffer from a tradeoff between accuracy and efficiency: To improve accuracy, finer label spaces or dictionaries must be provided, which makes online procedure less efficient. This can be problematic especially when both accuracy and efficiency are highly demanded.

One alternative way to bypass the above issue is to directly predict a set of parameters in a single step. This approach has been employed in the camera pose estimation problem, which has been an active research topic in the computer vision community. Specifically, recent camera pose estimation methods [6], [7], [8], [9], [10], [11] fine-tune pre-trained CNN (e.g., GoogLeNet [23], ResNet [24]) to directly regress pose parameters from the input frame. Adopting these methods seems to be a straightforward solution for sports field localization, but it has two major issues that have not been considered. First, the appearances of sports fields are different among courts/stadiums (cf., Fig. 6). This means one parameter set may correspond to multiple appearances of courts, which is very different from the typical camera pose estimation setting where one parameter set corresponds to almost only one appearance. Second, parameters to be predicted (i.e., homography) have higher DoF than pose parameters. This mainly comes from different camera settings: While intrinsic parameters are fixed (or known) in camera pose estimation, this does not hold in sports field localization due to different camera installations or some camera work like zooming. Regressors should deal with these issues, but they are not explicitly considered in existing camera pose estimation methods.

We develop our SFLNet based on these understandings. In the next section, we will detail SFLNet with respect to its architecture and training.

3. SFLNet

3.1 Architecture

Figure 2 shows the architecture of our proposed SFLNet. Once a frame is fed into the network, SFLNet generates a parameter set **p**, a segmentation mask **B** and a label adjacency prediction **a**, where **B** and **a** are the by-products used in our model training. SFLNet consists of (A) semantic segmentation module, (B) parameter regression module, and (C) label adjacency prediction module, which will be described in the following.

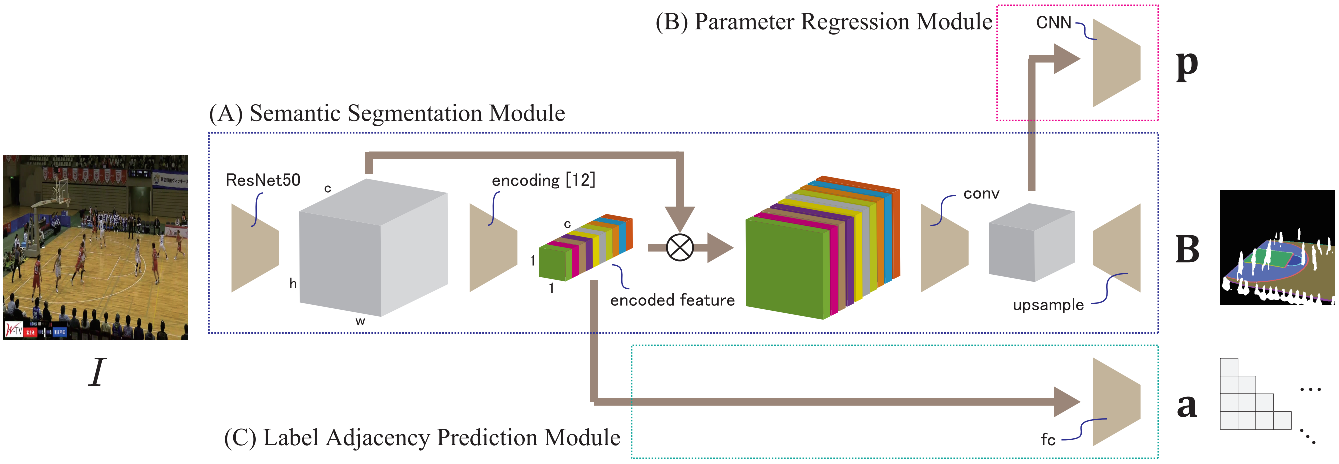

(B) Parameter Regression Module
CNN
p
(A) Semantic Segmentation Module
ResNet50
c
h
w
encoding [12]
c
1
1
encoded feature
conv
upsample
B
I
(C) Label Adjacency Prediction Module
fc
a

Fig. 2 The architecture of SFLNet, which consists of (A) Semantic Segmentation Module, (B) Parameter Regression Module and (C) Label Adjacency Prediction Module. SFLNet takes a single frame as input, then generates a set of parameters **p**, a label mask **B** and a label adjacency prediction **a**. The input of (B) is the output of the last convolutional layer in (A), and the input of (C) is the encoded feature produced by an encoding module [12] in (A). Notice that ⊗ represents channel-wise multiplication. More details are described in Section 3. Best viewed in color.

3.1.1 (A) Semantic Segmentation Module

The semantic segmentation module assigns one of the predefined labels to every pixel in an input frame. This is helpful for a regressor to understand the spatial layout of a sports court under large intra-class variance. In our problem, we have several choices for defining labels. One of the simplest cases is to divide a frame into court, person, and background regions as shown in **Fig. 3** (a), which is a relatively easier setting for semantic segmentation but only coarser information remains for the following regression. On the contrary, we can also define more labels like Fig. 3 (b)–(e) for finer layout representations, which provide richer information for the regressor but pose more difficult problems for semantic segmentation. Note that we can use these label definitions with almost the same annotation cost by following an approach shown in Section 3.2. In this work we select the best label definition experimentally, as will be shown in our parameter studies (cf., Section 4.4).

We build this segmentation module based on the state-of-the-art semantic segmentation approaches [12], [25]. Specifically, we use 50-layer ResNet [24] pretrained on ImageNet as a backbone and build the Context Encoding module [12] on top of the last convolutional layer right before the upsampling module to yield a per-pixel prediction. The output feature of the Context Encoding module is used as the input of the Label Adjacency Prediction module detailed later. To obtain higher resolution feature maps which preserve finer spatial information, we adopt Joint Pyramid Upsampling module [25] to our backbone network, which can approximate standard dilated convolution [26], [27], [28] while saving computation and memory overhead.

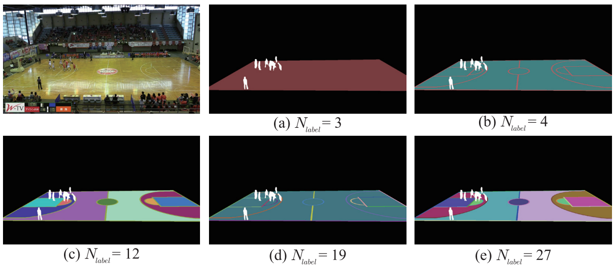
3.1.2 (B) Parameter Regression Module

This module regresses parameters that correspond a court metric model to the court seen in an input frame. In this work we define the parameter set as a 8-dimentional vector, where each value corresponds to a free parameter of a homography $H \in \mathbb{R}^{3\times3}$. Specifically, we relate a prediction $\mathbf{p} = [p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8]^{\mathrm{T}}$ to a homography H as follows:

$$H = \begin{bmatrix} p_1 + 1 & p_2 & p_3 \\ p_4 & p_5 + 1 & p_6 \\ p_7 & p_8, & 1 \end{bmatrix}. \tag{1}$$

Following Ref. [29], before computing the homography we normalize coordinate systems of both the model and the frame.

We build this module as a tiny CNN on top of the last convolutional layer of the semantic segmentation module, i.e., the input of this module is the output of the last convolutional layer [*3] in the semantic segmentation module as in Fig. 2. We set this CNN architecture as `C36-C48-C60-F8`, where `Ck` denotes a Convolution-BatchNorm-Relu-Maxpool block with `k` filters, `Fk` denotes a fc layer with `k` neurons. Every convolution layer has 3 × 3 filters and the stride of each maxpooling layer is set as 2.


(a) $N_{label} = 3$ (b) $N_{label} = 4$

(c) $N_{label} = 12$ (d) $N_{label} = 19$ (e) $N_{label} = 27$

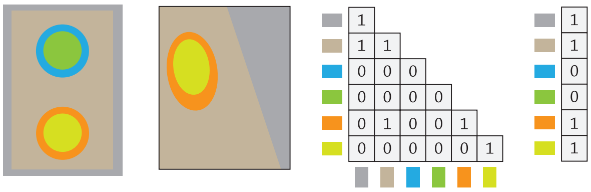
Fig. 3 We have several design choices about the label definition of a sports field. Examples (a)–(e) are ground truth segmentation masks of the top left image on different label definitions, where N_{label} is the number of labels used in each setting. Different colors represent different labels in each case. Best viewed in color.

Notice that we may have a choice to define $\mathbf{p}$ as a parameter set yielded via homography decomposition [10]: By introducing the natural camera assumption [13], we can break up a homography into a focal length, a rotation matrix and a translation vector, which in total have smaller degrees of freedom (7-DoF) than the homograpy itself. However, our preliminary experiments indicate that simply predicting the decomposed parameters does not work well for sports field localization. One reason is that errors are amplified when homographies are recovered, resulting in totally different results from ground truth. Therefore, in this work we regress homographies almost directly, and leave the above issue as a future work.

3.1.3 (C) Label Adjacency Prediction Module

In the standard training process of semantic segmentation, the network is learned from isolated pixels and context (e.g., sizes, spatial relations) is not explicitly considered. Here we introduce Label Adjacency Prediction (LAP) module to regularize the model training via exploiting contextual information lying in the sport field localization problem. Specifically, LAP module predicts the adjacencies of label pairs in addition to their presence in an input frame. **Figure 4** shows a toy example, in which the labels of this court are defined as in (a). When the court is shown like (b) in a frame, the corresponding ground truth for the output of LAP module is (c), in which each orthogonal element represents the presence of the label (1 if the label exists and 0 otherwise) and the others represent adjacencies of label pairs (1 if the row-column pair is adjacent and 0 otherwise).

LAP module can be seen as an extension of Semantic Encoding (SE) module [12] in the context of sports field localization. We can say almost all sports fields are not deformed and spa-



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

Fig. 4 In this toy example, the label mask of a field metric model (a) is transformed into a frame like (b). SE module of Zhang et al. [12] ideally predicts only the presence of each label like (d). In addition to the label presence, our proposed LAP module also predicts whether each label pair is adjacent or not like (c). Best viewed in color.

[*3] Hoping to alleviate the effect of non-court regions (e.g., background and player), in our preliminary study we have tried to exclude the channels that do not belong to the sports field from the output before feeding it to the regression CNN. However this makes the results slightly worse, which indicates our regressor learns to predict parameters from these non-court areas. Therefore in this study we feed the output of the last convolutional layer in the semantic segmentation module directly to the parameter regression module.

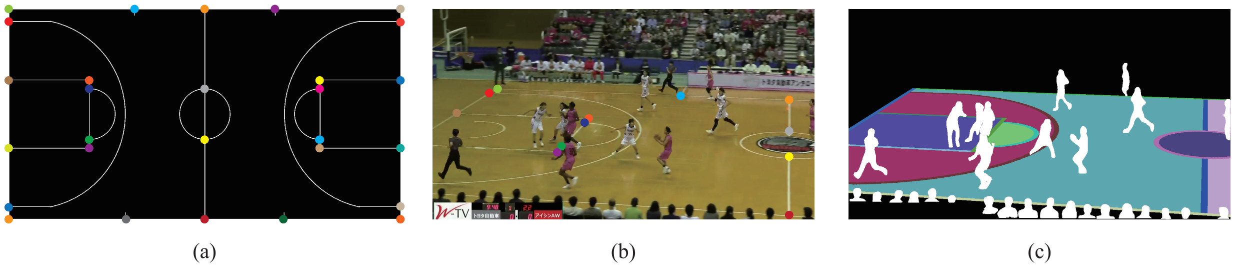


Fig. 5 Given a frame (b) and human annotated point-to-point correspondences to a field model (a), ground truth label mask $\mathbf{B}^*$ shown in (c) can be automatically generated by the approach described in Section 3.2.1. In (a) and (b), the markings of the same color are a correspondence. Best viewed in color.

tial relations between their parts (i.e., labels that belong to the court) are always consistent. While SE module makes predictions only for the presence of labels in the frame (cf., Fig. 4 (d)), LAP module can also exploit this problem-specific spatial consistency between labels, which makes LAP module a more efficient regularizer during the model training. Notice that we make our LAP module predict all the adjacencies of label pairs including non-court labels (i.e., person and background). While players and referees move on the court and adjacencies related to the person label would be different between images, LAP module still helps to recover spatial relations between labels and improve semantic segmentation performance, as will be shown in Section 4.4.

Following Ref. [12], we implement LAP module as an additional fully connected layer with a sigmoid activation function, which feeds an encoded feature produced by the context encoding layer [12] as input (cf., Fig. 2). The output dimension depends on the label definition, which can be computed by $N_{label}(N_{label}+1)/2$. LAP module usually has higher computation cost than SE module due to larger output dimensions, but the overall cost is still very small.

3.2 Training

3.2.1 Training Data

To learn model weights of SFLNet $\mathbf{\Theta}$, we need to provide a training data $\mathcal{D} = \{(I, \mathbf{p}^*, \mathbf{B}^*, \mathbf{a}^*)\}$, consisting of the quadruplets of a frame I, a ground truth parameter set $\mathbf{p}^*$, a label mask $\mathbf{B}^*$ and a label adjacency indicator $\mathbf{a}^*$. Unfortunately, fully manual labeling of such a dataset is costly and cumbersome. So here we propose a semi-automatic approach to obtain the training data $\mathcal{D}$ from human annotated point-to-point correspondences. For each frame $I \in \mathcal{D}$ and its point-to-point correspondences, we first apply DLT algorithm to estimate a homography H^* that transforms a court model into the court seen in the frame. This homography can be used to project the label mask of the court model into the frame. Since players and referees are usually on the sports fields, we adopt a state-of-the-art person segmentation algorithm [30] and overlay the segmentation result to the projected court labels to obtain a label mask $\mathbf{B}^*$. An example generated through the above procedure is shown in **Fig. 5**. While segmentation results of Ref. [30] are almost correct in our test case, if the person segmentation clearly fails then we remove the frame from the dataset. Yielding the parameter set $\mathbf{p}^*$ from H^* is straightforward and the label adjacency indicator $\mathbf{a}^*$ can easily be computed from $\mathbf{B}^*$.

3.2.2 Loss Function

Given a training dataset $\mathcal{D} = \{(I, \mathbf{p}^*, \mathbf{B}^*, \mathbf{a}^*)\}$, model weights of SFLNet are learned by minimizing the following loss function:

$$L_{\mathcal{D}}(\Theta) = \sum_{i}^{|\mathcal{D}|} \tau(\mathbf{p}_i, \mathbf{p}_i^*) + w_\phi \sum_{i}^{|\mathcal{D}|} \phi(\mathbf{B}_i, \mathbf{B}_i^*) + w_\psi \sum_{i}^{|\mathcal{D}|} \psi(\mathbf{a}_i, \mathbf{a}_i^*), \tag{2}$$

where on the right side the first term is a parameter loss, the second term is a segmentation loss and the third term is a label adjacency prediction loss. Following Ref. [12], we use a per-pixel cross-entropy loss as ϕ and a binary cross-entropy as ψ. Since we have several choices for defining the parameter loss τ, we experimentally decide the best which is shown in Section 4.4.

A straightforward way of optimization for Eq. (2) is to minimize all the loss components all at once. Alternatively, in this work we use the following two-step approach that we found works better than the above: We first train the semantic segmentation module and the label adjacent prediction module by considering the corresponding losses (i.e., first and second terms on the right side of Eq. (2)), then optimize the whole model by minimizing the loss $L_{\mathcal{D}}$. Multi-step optimization is a common strategy in the deep multi-task learning literature [30], [31]. Intuitively, in our approach we first warm up modules related to semantic segmentation in the first step, then optimize all the modules including parameter regressor, which would ease the whole model training.

4. Experimental Evaluation

4.1 Evaluation Protocols

As discussed in Section 3.1, SFLNet produces three outputs: a parameter regression result $\mathbf{p}$ which will be transformed into a homography, a segmentation prediction result $\mathbf{B}$ and a label adjacency prediction result $\mathbf{a}$. While $\mathbf{p}$ is the main output for sports field localization, in this section we will evaluate all of the above with the following protocols tailored for each of them:

Parameter regression ($\mathbf{p}$): For $\mathbf{p}$, we evaluate how correctly it can predict the court shown in an input frame. To do so, here we compute an overlap between a predicted court and its metric model in one coordinate system. Specifically, we first transform $\mathbf{p}$ into a corresponding homography so as to generate a binary mask which represents the predicted court region in a coordinate system of the metric model. Corner points of the court are projected to generate the mask, where their positions in the image coordi-

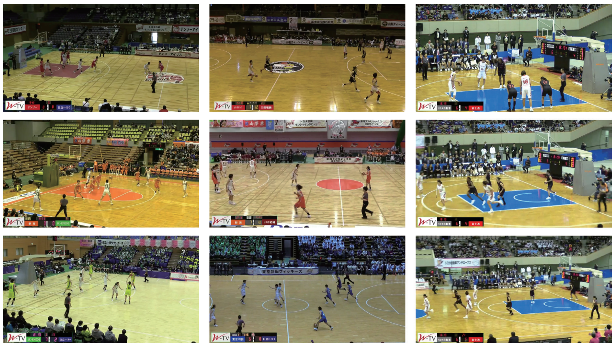


Fig. 6 Example frames included in our dataset. Some frames have similar parameters but their appearances are different (first and second column). Also some frames from the same game are captured with different intrinsic parameters (third column).

nate system is computed using a ground truth homography. We then compute the intersection-over-union (IoU) score between the predicted court and the metric model, each of which is defined as a binary mask. We use this IoU score as our metric, denoting it as $J_{\mathbf{p}}$.

Segmentation prediction (**B**): We use the IoU score between a predicted segmentation **B** and a ground truth, which is a standard metric for semantic segmentation. We denote the score as $J_{\mathbf{B}}$.

Label adjacency prediction (**a**): Since label adjacency (as shown in Fig. 4 (c)) can be seen as binary label set, we can evaluate this output via computing the IoU between **a** and the ground truth label adjacency. We denote the score as $J_{\mathbf{a}}$.

4.2 Dataset

In this work we create a new dataset for evaluating sport field localization methods using videos of basketball games. Basketball is challenging for this task because the appearances of basketball courts are varied between stadiums, and different court regions are occluded by players or referees moving over time (cf., **Fig. 6**). We collected the videos of 22 games from a Japanese basketball league, each of which is held in a unique stadium. For each video we sequentially sampled 50–60 frames [*4], and manually annotated point-to-point correspondences to each frame. Every frame size is 1,024 × 720. Points to be annotated are defined as in Fig. 5 (a), and we specified the position only if it can be seen within the frame. After discarding frames in which less than 4 points are annotated (i.e., DLT cannot be performed), we obtained the whole dataset consisting of 1,232 frames. This dataset can be used to automatically build the training data of SFLNet, following the procedure detailed in Section 4.2.

Note that we believe this dataset cannot be used to learn sequential models because our frame sampling is not so dense (i.e., about one frame per second). We are planning to extend this dataset for sequence learning, and leave it as a future work.

4.3 Implementation Details

We implemented our algorithms with PyTorch [*5], using the SGD optimizer with momentum of 0.9. The input frames are scaled to 448 × 448 pixels, and normalized by pixel mean subtraction and standard deviation division. In training we randomly crop training frames keeping its aspect ratio, and recompute the ground truth parameter $\mathbf{p}^*$ accordingly. Flipping and rotation are not performed since they degrade the performance. We use the mini-batch size of 16 during the training, and apply the approach of Ref. [28] to control the learning rate. We run 50 epochs on both steps in training, and use the final model for evaluating test data. Using a grid search, we set the hyperparameters w_ϕ and w_ψ in Eq. (2) as 1.0 and 0.2, respectively.

Table 1 Ablation for different architectures. Notice that in all the settings N_{label} is set to 27.

Segmentation?	Context?	$J_{\mathbf{B}}$	$J_{\mathbf{p}}$	$J_{\mathbf{B}\to\mathbf{a}}$	$J_{\mathbf{a}}$
	None	-	0.855	-	-
✓	None	0.489	0.892	0.643	-
✓	SE module [12]	0.504	0.909	0.712	-
✓	LAP module	0.521	0.924	0.779	0.818

4.4 Ablation/Parameter Study

In this section we perform several ablation/parameter studies with respect to (i) architecture designs, (ii) label definitions and (iii) loss functions for parameter regression. In the following we used all the frames in one game (denoted as #1) as test data and all the remaining as training data.

4.4.1 Architecture Design

We first validate our architecture design of SFLNet, focusing on the semantic segmentation module and the label adjacency prediction module. To evaluate the semantic segmentation module, we built an alternative model that replaced the CNN backbone of SFLNet to vanilla ResNet-50 and introduced a fc layer with 2,048 neurons after its global average pooling layer followed by ReLU and dropout with $p = 0.5$. This is followed by a final fc layer that outputs a parameter set $\mathbf{p}$. For label adjacency prediction module, we considered the following two alternatives: (1) simply removing the module from SFLNet, (2) replacing LAP module to SE module [12]. Loss functions are modified accordingly. **Table 1** shows the results of all the settings. Comparing the first row and others, we can see parameter regression ($J_{\mathbf{p}}$) is significantly improved by introducing the semantic segmentation module. Also, rows 2–4 indicate semantic segmentation ($J_{\mathbf{B}}$) performance is well correlated to parameter regression ($J_{\mathbf{p}}$) performance, and the best result is achieved when our LAP module is employed. To further analyze the results, we transform semantic segmentation results into label adjacency prediction results and compute IoU scores to ground truth, which is denoted as $J_{\mathbf{B}\to\mathbf{a}}$. From Table 1 the best $J_{\mathbf{B}\to\mathbf{a}}$ is achieved by the model with our LAP module and its performance is reaching the one produced by LAP module itself ($J_{\mathbf{a}}$). This indicates one reason for higher segmentation performance using LAP module is that LAP module more correctly regularizes label adjacencies than alternatives, which can also be seen in qualitative results shown in **Fig. 7**. From these facts we can say our SFLNet design is effective.

4.4.2 Label Definition

Here, we compare the performance of SFLNet on 5 different label definitions shown in Fig. 3. **Table 2** shows the results. Notice that in Table 2 semantic segmentation performance ($J_{\mathbf{B}}$) cannot be directly compared between different label definitions: Its

[*4] We avoid sampling when the game is stopping in order not to sample duplicate frames.

[*5] https://pytorch.org/

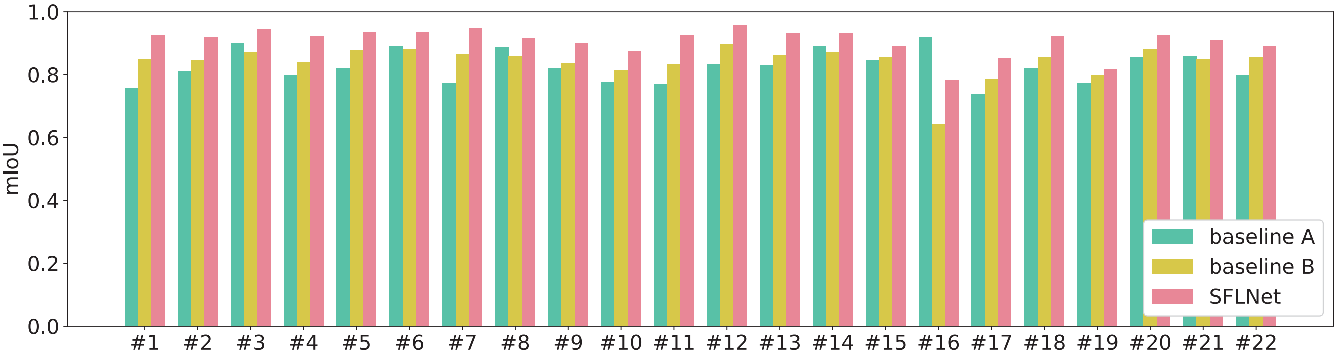
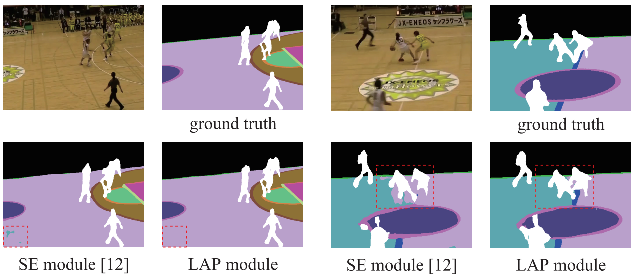


Fig. 8 Quantitative comparison between methods. #k represents the game id. Best viewed in color.



ground truth

ground truth

SE module [12]

LAP module

SE module [12]

LAP module

Fig. 7 Qualitative comparisons of semantic segmentation between SE module [12] and our LAP module. LAP module can regularize the correct adjacencies between labels, which produces better segmentation than SE module (cf., areas surrounded by red dotted lines). Best viewed in color.

Table 2 Comparison on different label definitions.

	SE module [12]		LAP module	
N_{label}	J_{B}	J_{p}	J_{B}	J_{p}
(a)3	0.860	0.855	0.864	0.875
(b)4	0.737	0.861	0.743	0.868
(c)12	0.685	0.913	0.712	0.919
(d)19	0.347	0.854	0.398	0.898
(e)27	0.498	0.909	0.521	0.924

Table 3 Comparison on different loss functions.

loss	J_{p}
L2	0.912
L1	0.924
SmoothL1 [31]	0.887

difficulty heavily depends on the number of labels and the shape of them (e.g., thin lines). Basically, finer label definitions would be more helpful for parameter regression, but their prediction is more difficult for semantic segmentation. For J_{p}, the best performance is achieved when $N_{label} = 27$ with LAP module, which is difficult for semantic segmentation (i.e., J_{B} is low). Interestingly, when SE module [12] is used, J_{p} achieves the peak at $N_{label} = 12$, which is relatively easier for semantic segmentation. One possible reason for this difference is that LAP module works better than SE module on the challenging setting of $N_{label} = 27$, making some positive effects to parameter regression. This result indicates our LAP module can achieve better tradeoffs between parameter regression and semantic segmentation.

4.4.3 Loss Function for Parameter Regression

As discussed in Section 3.1, we have several choices for evaluating the parameter loss (i.e., τ in Eq. (2)). Here, we applied L1, L2 and smoothed L1 [31] losses to SFLNet and evaluate the performances. From the results shown in **Table 3**, we chose L1 loss for our parameter loss and used it in the following experiments.

4.5 Comparison to Baselines

Based on the above ablation results, we compared our approach to existing methods after tuning SFLNet to the best setting: We used both semantic segmentation and label adjacency prediction modules, and set $N_{label} = 27$ and τ as L1 norm. In the following evaluations we used our dataset in 1-vs-all manner. Specifically, we used all the frames from one game as a test set, and all the remaining as a training (or dictionary) set. Since to our knowledge existing works do not make their codes public, we implemented the following baselines for comparison:

- **Baseline A** This baseline extracts line parameters from semantic segmentation results and estimates a homography from line-to-line correspondences. We used segmentation results of SFLNet (setting $N_{label} = 27$) to estimate line parameters via the approach shown in Ref. [3] and used RANSAC for robust parameter estimation.
- **Baseline B** This baseline retrieves a dictionary (i.e., training data) based on a visual feature extracted from frames, and returns a homography corresponding to the nearest neighbor data. We used the intermediate feature map of SLFNet and used L2 norm for computing a similarity. We experimentally found that SFLNet feature works better than typical CNN feature extractors like ResNet.

Figure 8 shows the results with respect to J_{p}. We can see that in most games SFLNet achieves the best results. Compared to baseline B, SFLNet achieves better results in all the cases. Qualitative results shown in **Fig. 9** also indicate SFLNet can correctly predict transformations between frames and the court model. However, in some cases (i.e., #16, #19) SFLNet does not perform well, and especially in the case of #16 the result of SFLNet is worse than baseline A. Some typical failure modes are shown in **Fig. 10**. One possible reason is a limited generalization power of our approach: Since in our dataset courts seen in frames like Fig. 10 are rare, SFLNet might fail to predict correct parameters. We may need to incorporate human supervision to address such unseen data.

Lastly, average running times per frame of methods are listed in **Table 4**. SFLNet is much faster than baselines and can be run over 30 FPS. Based on these results, we can say that our CNN-based single shot regressor is a reasonable choice with respect to both accuracy and efficiency for sports field localization.

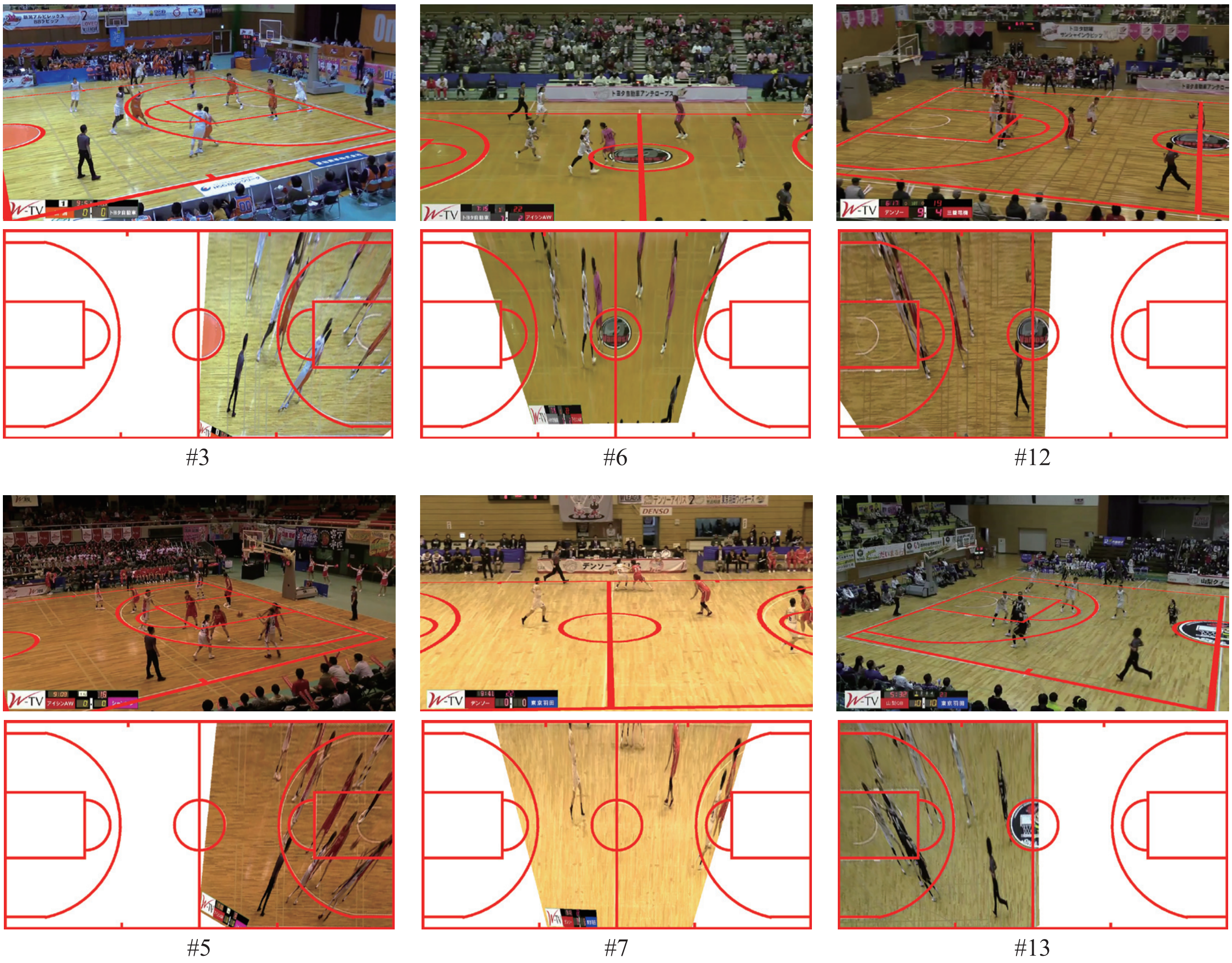


Fig. 9 Qualitative results of SFLNet. Odd rows show the projection of the model to the frame, and even rows show vice versa. #k represents the game id. Best viewed in color.

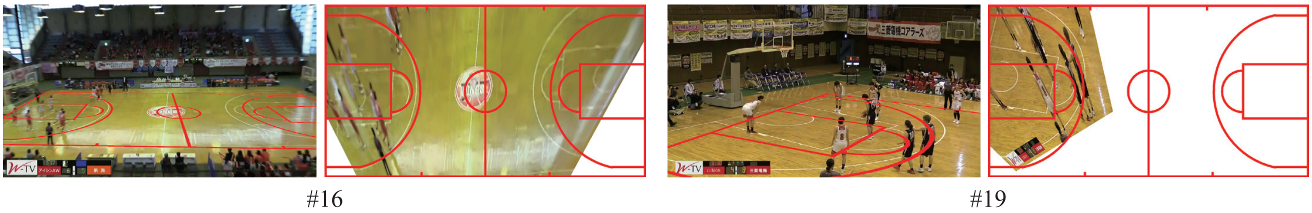


Fig. 10 Failure modes of SFLNet. #k represents the game id. Best viewed in color.

Table 4 Average running times per frame. We ran the algorithms on a standard desktop PC with a single GPU.

[ms]	Baseline A	Baseline B	SFLNet
	91.7	73.5	31.0

5. Conclusion

In this paper we proposed SFLNet, a CNN-based single shot regressor that predicts a parameter set relating a sports field in an input frame to its metric model. Experimental evaluations on our new dataset based on basketball games demonstrated that SFLNet can predict the parameter more precisely than baseline methods.

As a future work, we will evaluate our approach on different sports such as soccer and hockey [3], [32]. We also plan to extend SFLNet to sequential models, which can accept a video directly and produce temporally smooth results.

References

[1] Theagarajan, R., Pala, F., Zhang, X. and Bhanu, B.: Soccer: Who Has The Ball? Generating Visual Analytics and Player Statistics, *CVPR Workshop* (2018).

[2] Giancola, S., Amine, M., Dghaily, T. and Ghanem, B.: SoccerNet: A Scalable Dataset for Action Spotting in Soccer Videos, *CVPR Workshop* (2018).

[3] Homayounfar, N., Fidler, S. and Urtasun, R.: Sports Field Localization via Deep Structured Models, *CVPR* (2017).

[4] Sharma, R.A., Bhat, B., Gandhi, V. and Jawahar, C.V.: Automated Top View Registration of Broadcast Football Videos, *WACV* (2018).

[5] Chen, J. and Little, J.J.: Sports Camera Calibration via Synthetic Data, *CVPR Workshop* (2019).

[6] Kendall, A. and Cipolla, R.: Modelling Uncertainty in Deep Learning for Camera Relocalization, *ICRA* (2016).

[7] Kendall, A. and Cipolla, R.: Geometric Loss Functions for Camera Pose Regression with Deep Learning, *CVPR* (2017).

[8] Kendall, A., Grimes, M. and Cipolla, R.: PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization, *ICCV* (2015).

[9] Sattler, T., Zhou, Q., Pollefeys, M. and Leal-Taixé, L.: Understanding the Limitations of CNN-based Absolute Camera Pose Regression,

CVPR (2019).

[10] Carr, P., Sheikh, Y. and Matthews, I.: Point-less Calibration: Camera Parameters from Gradient-Based Alignment to Edge Images, *WACV* (2012).

[11] Brahmbhatt, S., Gu, J., Kim, K., Hays, J. and Kautz, J.: Geometry-Aware Learning of Maps for Camera Localization, *CVPR* (2018).

[12] Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A. and Agrawal, A.: Context Encoding for Semantic Segmentation, *CVPR* (2018).

[13] Hartley, R. and Zisserman, A.: *Multiple View Geometry in Computer Vision*, Cambridge University Press (2004).

[14] Kim, H. and Hong, K.S.: Soccer Video Mosaicing using Self-calibration and Line Tracking, *ICPR* (2000).

[15] Yamada, A., Shirai, Y. and Miura, J.: Tracking Players and A Ball in Video Image Sequence and Estimating Camera Parameters for 3D Interpretation of Soccer Games, *ICPR* (2002).

[16] Farin, D., Krabbe, S., de With, P.H.N. and Effelsberg, W.: Robust Camera Calibration for Sport Videos using Court Models, *Electronic Imaging* (2004).

[17] Watanabe, T., Haseyama, M. and Kitajima, H.: A Soccer Field Tracking Method with Wire Frame Model from TV Images, *ICIP* (2004).

[18] Wang, F., Sun, L., Yang, B. and Yang, S.: Fast Arc Detection Algorithm for Play Field Registration in Soccer Video Mining, *ICSMC* (2006).

[19] Okuma, K., Little, J. and Lowe, D.: Automatic Rectification of Long Image Sequences, *ACCV* (2004).

[20] Dubrofsky, E. and Woodham, R.J.: Combining Line and Point Correspondences for Homography Estimation, *ISVC* (2008).

[21] Hess, R. and Fern, A.: Improved Video Registration using Nondistinctive Local Image Features, *CVPR* (2007).

[22] Gupta, A., Little, J.J. and Woodham, R.J.: Using Line and Ellipse Features for Rectification of Broadcast Hockey Video, *CRV* (2011).

[23] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A.: Going Deeper with Convolutions, *CVPR* (2015).

[24] He, K., Zhang, X., Ren, S. and Sun, J.: Deep Residual Learning for Image Recognition, *CVPR* (2016).

[25] Wu, H., Zhang, J., Huang, K., Liang, K. and Yu, Y.: FastFCN: Rethinking Dilated Convolution in the Backbone for Semantic Segmentation, arXiv preprint arxiv:1903.11816 (2019).

[26] Chen, L.-C., Papandreou, G., Schroff, F. and Adam, H.: Rethinking atrous Convolution for Semantic Image Segmentation, arXiv preprint arxiv:1706.05587 (2017).

[27] Yu, F., Koltun, V. and Funkhouser, T.: Dilated Residual Networks, *CVPR* (2017).

[28] Zhao, H., Shi, J., Qi, X., Wang, X. and Jia, J.: Pyramid Scene Parsing Network, *CVPR* (2017).

[29] Lin, C.-H. and Lucey, S.: Inverse Compositional Spatial Transformer Networks, *CVPR* (2017).

[30] He, K., Gkioxari, G., Dollár, P. and Girshick, R.: Mask R-CNN, *ICCV* (2017).

[31] Girshick, R.: Fast R-CNN, *ICCV* (2015).

[32] Chen, J., Zhu, F. and Little, J.J.: A Two-point Method for PTZ Camera Calibration in Sports, *WACV* (2018).

Editor's Recommendation

This paper proposes a method of homography transformation for a sports field appearing in video images taken by a camera at an unknown position. Although the use of the end-to-end architecture itself is a common approach, the overall quality of the system has been proven to be high, and accurate localization for the field can be achieved with few manual operations, utilizing the general characteristics of the field. The original contributions, such as the label estimation of adjacent regions, are also fully verified.

(Chairman of Program Committee of FIT2019, Kunio Kashino)

Shuhei Tarashima received his B.S. and M.S. degrees from The University of Tokyo in 2009 and 2011, respectively. He joined NTT in 2011 and he is currently a researcher at NTT Communications Corporation. His research interest is computer vision and pattern recognition. He is a member of the IPSJ.